



مدخل إلى الحاسوب والبرمجة



منشورات جامعة دمشق
كلية الهندسة الميكانيكية والكهربائية

مدخل إلى الحاسوب والبرمجة

بنية الحاسوب وتنظيمه & البرمجة بلغة C++

الدكتور المهندس

جمال الياسين

قسم هندسة الحواسيب والأتمتة

الدكتور

أكرم مذكور

قسم هندسة الحواسيب والأتمتة

الدكتور المهندس

مأمون يونس

قسم هندسة الحواسيب والأتمتة

1435 – 1436

2014 – 2015

جامعة دمشق :



الفهرس

الفصل الأول

بنية الحاسوب وتنظيمه

٢٥	١-١- تطور الحاسوب وأجياله
٢٥	١-١-١- مقدمة
٢٥	١-١-٢- أجيال الحاسوب
٣٠	٢-١- تعريف الحاسوب
٣٣	٣-١- المكونات الأساسية للحاسوب
٣٥	٤-١- لوحة النظام
٣٧	١-٤-١- وحدة المعالجة المركزية
٤١	١-٤-٢- وحدة الذاكرة الرئيسية
٤٩	١-٤-٣- وحدة الترابط مع الحاسوب
٥٤	٥-١- الوحدات المحيطية لنظام الحاسوب
٥٤	١-٥-١- وحدات الإدخال
٦٧	١-٥-٢- وحدات الإخراج
٧٦	٦-١- وحدات التخزين الثانوية
٨٦	٧-١- أداء الحاسوب
٨٨	٨-١- نظام الإدخال والإخراج الأساسي
٨٩	٩-١- شريحة الـ CMOS
٩١	١٠-١- النفاذ المباشر على الذاكرة DMA

٩١	١١-١ - الذاكرة الافتراضية
٩٢	١٢-١ - الحاسوب في حياتنا
٩٦	١٣-١ - تصنيف الحواسيب

الفصل الثاني

أنظمة العد وتمثيل المعطيات

١٠٧	١-٢ - مقدمة
١٠٧	٢-٢ - أنواع أنظمة العد
١٠٨	٣-٢ - نظام العد العشري
١١٠	٤-٢ - نظام العد الثنائي
١١٣	٢-٤-١ - التحويل من النظام الثنائي إلى النظام العشري
١١٤	٢-٤-٢ - التمثيل الثنائي للأعداد العشرية
١١٥	٢-٤-٣ - إجراء العمليات الحسابية على الأعداد الموجبة
١١٩	٢-٥-١ - التحويل من النظام العشري إلى النظام الثنائي
١٢٦	٢-٦-١ - نظام العد الثنائي
١٢٧	٢-٦-٢ - التمثيل الثنائي لنظام العد الثنائي
١٢٧	٢-٦-٣ - التحويل من النظام الثماني إلى العشري
١٢٩	٢-٦-٣ - التحويل من النظام العشري إلى الثماني
١٣١	٢-٦-٤ - التحويل من النظام الثماني إلى الثنائي
١٣٣	٢-٦-٥ - التحويل من النظام الثنائي إلى الثماني
١٣٥	٢-٦-٦ - جمع وطرح الأعداد الثمانية

١٣٧	٧-٢- نظام العد السداسي عشر
١٣٨	١-٧-٢- التمثيل الثنائي لنظام العد السداسي عشر
١٤٦	٢-٧-٢- التحويل من النظام العشري إلى السداسي عشر
١٣٩	٣-٧-٢- التحويل من النظام العشري إلى السداسي عشر
١٤٠	٤-٧-٢- التحويل من النظام السداسي عشر إلى الثنائي
١٤١	٥-٧-٢- التحويل من النظام الثنائي إلى السداسي عشر
١٤٣	٦-٧-٢- التحويل من النظام الثماني إلى السداسي عشر
١٤٦	٧-٧-٢- التحويل من النظام السداسي عشر إلى الثماني
١٥١	٨-٢- النظام العشري المرمز ثنائيا BCD
١٥٢	١-٨-٢- عملية الجمع في نظام الـ BCD
١٥٣	٢-٨-٢- ملاحظات حول القيمة 0110
١٥٦	٩-٢- الترميز وتمثيل المعطيات
١٥٦	١-٩-٢- ترميز ASCII
١٥٧	٢-٩-٢- ترميز EBCDIC

الفصل الثالث

البرمجيات ونظم التشغيل

١٦١	١-٣- البرمجيات
١٦١	١-١-٣- أنواع البرمجيات
١٦٥	٢-١-٣- لغات البرمجة
١٦٥	١-٢-١-٣- تطور لغات البرمجة
١٦٥	٢-٢-١-٣- مستويات لغات البرمجة

١٦٧	٣-٢-١-٣- بعض أنواع لغات البرمجة
١٧٠	٣-٢-٤- مترجمات لغات البرمجة عالية المستوى
١٧٢	٣-٢- نظم التشغيل
١٧٢	٣-٢-١- مقدمة
١٧٣	٣-٢-٢- التطور التاريخي لنظم التشغيل
١٧٧	٣-٢-٣- تعريف نظام التشغيل
١٨٠	٣-٢-٤- أنواع أنظمة التشغيل
١٨١	٣-٢-٥- واجهات المستخدم في نظم التشغيل
١٨٣	٣-٢-٦- بنية أنظمة التشغيل العامة
١٨٤	٣-٢-٧- مهام نظم التشغيل الرئيسية
١٨٧	٣-٢-٨- أنواع نظم التشغيل الشائعة
١٨٧	٣-٢-٨-١- نظام التشغيل DOS
١٩٤	٣-٢-٨-٢- نظام تشغيل النوافذ
٢٠١	٣-٢-٨-٣- نظام تشغيل ماكنتوش
٢٠٢	٣-٢-٨-٤- نظام تشغيل Chrome
٢٠٣	٣-٢-٨-٥- نظام تشغيل UNIX
٢٠٧	٣-٢-٨-٦- نظام التشغيل Linux
٢١٣	٣-٢-٨-٧- أنظمة تشغيل الشبكات

الفصل الرابع

الشبكات الحاسوبية والانترنت

٢١٧	٤-١- مقدمة
٢١٨	٤-٢- تعريف الشبكات الحاسوبية

٢١٨	٣-٤- أنواع الشبكات الحاسوبية
٢١٨	١-٣-٤- الشبكات المحلية LAN
٢١٩	٢-٣-٤- شبكات المدينة MAN
٢٢٠	٣-٣-٤- الشبكات الواسعة WAN
٢٢١	٤-٤- فوائد الشبكات الحاسوبية
٢٢٣	٥-٤- نظم تشغيل وإدارة الشبكات
٢٢٣	١-٥-٤- شبكات الند للند
٢٢٥	٢-٥-٤- شبكات المخدم/عميل
٢٢٦	٣-٥-٤- شبكات الند للند
٢٢٧	٦-٤- مكونات شبكات الحاسوب
٢٣١	٧-٤- طبولوجيا الشبكات الحاسوبية
٢٣٢	١-٧-٤- طريقة التوصيل الخطي
٢٣٤	٢-٧-٤- طريقة التوصيل انجمي
٢٣٥	٣-٧-٤- طريقة التوصيل الحلقي
٢٣٧	٤-٧-٤- طريقة التوصيل المتشابك
٣٢٨	٨-٤- وسائط نقل المعطيات في الشبكات
٢٣٩	١-٨-٤- وسائط الاتصال السلكية
٢٣٩	١-١-٨-٤- الكابلات المزدوجة المجدولة
٢٤١	٢-١-٨-٤- الكابلات المحورية
٢٤٥	٣-١-٨-٤- كابلات الألياف الضوئية
٢٥٠	٢-٨-٤- وسائط الاتصال اللاسلكية
٢٥١	١-٢-٨-٤- الأمواج الراديوية
٢٥١	٢-٢-٨-٤- الأمواج الميكروية

٢٥٢	 ٤-٨-٢-٣- الأقمار الصناعية
٢٥٣	 ٤-٩- بروتوكولات شبكات الحواسيب
٢٥٣	 ٤-٩-١- مقدمة
٢٥٣	 ٤-٩-٢- المنظمات الدولية للمقاييس والمعايير
٢٥٥	 ٤-٩-٣- المهام الأساسية لبروتوكولات الشبكات الحاسوبية
٢٥٧	 ٤-١٠- شبكة الانترنت
٢٥٧	 ٤-١٠-١- تعريف شبكة الانترنت
٢٥٧	 ٤-١٠-٢- التطور التاريخي لشبكة الانترنت
٢٦١	 ٤-١٠-٣- إدارة شبكة الانترنت
٢٦٢	 ٤-١٠-٤- متطلبات الدخول إلى الانترنت
٢٦٥	 ٤-١٠-٥- متطلبات الوصول إلى مزود خدمة الانترنت
٢٦٥	 ٤-١١- خدمات الانترنت
٢٦٥	 ٤-١١-١- شبكة الأخبار
٢٦٦	 ٤-١١-٢- خدمة الاتصال بوساطة الصوت والصورة
٢٦٦	 ٤-١١-٣- خدمة البريد الالكتروني
٢٦٩	 ٤-١١-٤- خدمة الغوفر
٢٧٠	 ٤-١١-٥- خدمة نقل الملفات
٢٧٠	 ٤-١١-٦- خدمة الدخول إلى حاسوب آخر عبر الشبكة
٢٧١	 ٤-١١-٧- الشبكة العنكبوتية العالمية WWW
٢٧٣	 ٤-١٢- شبكة الاكسترانت
٢٧٤	 ٤-١٣- أمن المعلومات
٢٧٦	 ٤-١٤- الفيروسات

الفصل الخامس

مبادئ الخوارزميات

٢٨١	١-٥- مقدمة
٢٨٢	٢-٥- تعريف الخوارزميات
٢٨٢	٣-٥- خطوات حل المسألة باستخدام الحاسوب
٢٨٥	٤-٥- أنواع الخوارزميات
٢٨٥	١-٤-٥- الخوارزميات غير الحسابية
٢٨٦	٢-٤-٥- الخوارزميات الحسابية
٢٨٧	٥-٥- أولوية تنفيذ العمليات الحسابية والمنطقية
٢٩١	٦-٥- طرق صياغة الخوارزميات
٢٩١	١-٦-٥- الطريقة اللغوية
٢٩٢	٢-٦-٥- الطريقة الترميزية
٢٩٣	٣-٦-٥- الطريقة البيانية
٢٩٤	٧-٥- فوائد استخدام المخططات التدفقية
٢٩٥	٨-٥- أنواع المخططات التدفقية
٢٩٦	١-٨-٥- المخطط التدفقي التتابعي البسيط
٢٩٩	٢-٨-٥- المخطط التدفقي التفرعي
٣٠٣	١-٨-٥- المخطط التدفقي الحلقي
٣٠٥	٩-٥- العداد
٣٠٧	١٠-٥- المجاميع الإجمالية
٣٠٩	١١-٥- المخططات التدفقية الحلقية المتداخلة
٣١٠	١٢-٥- صيغة التكرار باستخدام الشكل الاصطلاحي

٣١٤	١٣-٥- تطبيقات على الخوارزميات
٣٢١	مسائل عامة

الفصل السادس

أساسيات لغة C++

٣٢٥	١-٦- مقدمة
٣٢٦	٢-٦- تاريخ تطور لغة C++
٣٢٨	٣-٦- مزايا لغة C/C++
٣٢٩	٤-٦- بنية لغة C++
٣٣٠	٥-٦- الشكل العام لبنية البرنامج بلغة C++
٣٣١	٦-٦- قواعد بناء البرنامج بلغة C++
٣٣٨	١-٦-٦ متواليات الهروب
٣٣٩	٢-٦-٦ التعليمة endl
٣٣٩	٣-٦-٦ رموز لغة C++
٣٣٩	٤-٦-٦ الكلمات المحجوزة في لغة C++
٣٤٠	٧-٦- المتحولات والثوابت
٣٤١	١-٧-٦ انواع المعطيات الأساسية في لغة C++
٣٤٢	٢-٧-٦ المتحولات
٣٤٢	١-٢-٧-٦ تسمية المتحولات
٣٤٣	٢-٢-٧-٦ مجال المتحولات
٣٤٤	٣-٢-٧-٦ التصريح عن المتحولات
٣٥٠	٣-٧-٦ الثوابت
٣٥١	١-٣-٧-٦ تعريف الثوابت برمجياً

٣٥٢	٦-٧-٣-٢- التصريح عن الثوابت
٣٥٣	٦-٨- تعليمات الإدخال والإخراج الأساسية
٣٦١	٦-٩- العوامل الحسابية والمنطقية والعلائقية
٣٦١	٦-٩-١- العوامل الحسابية
٣٦٣	٦-٩-٢- العوامل العلائقية
٣٦٥	٦-٩-٣- العوامل المنطقية
٣٦٦	٦-٩-٤- المعامل الشرطي ثلاثي الحدود
٣٦٨	٦-٩-٥- عوامل الإسناد المركبة
٣٦٨	٦-٩-٦- عمليات الزيادة والنقصان
٣٧١	٦-١٠- أولوية تنفيذ العمليات الحسابية والمنطقية والعلائقية
٣٧٣	مسائل عامة محلولة
٣٧٦	مسائل عامة غير محلولة

الفصل السابع

بنى التحكم والتكرار

٣٨١	٧-١- مقدمة
٣٨١	٧-٢- بنى التحكم
٣٨٢	٧-٢-١- التعليم if
٣٨٥	٧-٢-٢- التعليم if/else
٣٨٩	٧-٢-٣- التعليم if-else-if
٣٩٣	٧-٢-٤- التعليم if المركبة
٣٩٥	٧-٢-٥- بنية التعليم switch

٤٠٠	٧-٢-٦- بنية تعليمة التحكم (?:)
٤٠٢	٧-٣- بنى حلقات التكرار
٤٠٢	٧-٣-١- حلقة for
٤٠٨	٧-٣-٢- حلقة التكرار while
٤١٣	٧-٣-٣- بنية الحلقة do/while
٤١٩	٧-٤- تعليمات التفرع
٤١٩	٧-٤-١- تعليمة التوقف break
٤٢١	٧-٤-٢- تعليمة الاستمرار continue
٤٢٢	٧-٤-٣- تعليمة return
٤٢٢	٧-٥- تعليمات الانتقال goto
٤٢٤	٧-٦- تعليمات الخروج (exit)
٤٢٧	مسائل عامة محلولة
٤٣٧	مسائل عامة غير محلولة

الفصل الثامن

المصفوفات

٤٤١	٨-١- مقدمة
٤٤١	٨-٢- تعريف المصفوفة
٤٤٢	٨-٣- المصفوفات أحادية البعد
٤٤٢	٨-٣-١- التصريح عن المصفوفات أحادية البعد
٤٤٤	٨-٣-٢- إعطاء قيم ابتدائية لعناصر أحادية البعد
٤٤٦	٨-٣-٣- إدخال وإخراج عناصر المصفوفة أحادية البعد

٤٥٤	٨-٤- المصفوفات ثنائية البعد
٤٥٥	٨-٤-١- التصريح عن المصفوفات ثنائية البعد
٤٥٦	٨-٤-٢- إعطاء قيم ابتدائية لعناصر ثنائية البعد
٤٥٩	٨-٤-٣- إدخال وإخراج عناصر المصفوفة ثنائية البعد
٤٦٤	مسائل عامة محلولة
٤٧٣	مسائل عامة غير محلولة

الفصل التاسع

التوابع

٤٧٧	٩-١- مقدمة
٤٧٨	٩-٢- فوائد استخدام التوابع
٤٧٨	٩-٣- إنشاء التوابع في البرنامج
٤٨٠	٩-٣-١- التصريح عن التابع
٤٨٢	٩-٣-٢- استدعاء التابع
٤٨٤	٩-٣-٣- تعريف التابع
٤٩١	٩-٤- مكتبة التوابع الرياضية
٤٩٤	٩-٥- التوابع بدون وسطاء
٤٩٦	٩-٦- التوابع الفورية
٤٩٩	٩-٧- التحميل الزائد للتوابع
٥٠٢	٩-٨- الوسطاء الافتراضية
٥٠٤	٩-٩- التمرير بالقيمة والتمرير بالعنوان
٥٠٨	٩-١٠- التوابع العودية

٥١٢	١١-٩- توليد الأعداد العشوائية
٥١٨	١٢-٩- تمرير المصفوفات كوسطاء للتتابع
٥٢٦	مسائل عامة محلولة
٥٣٤	مسائل عامة غير محلولة

الفصل العاشر

المؤشرات

٥٤١	١-١- مقدمة
٥٤١	٢-١- تعريف المؤشرات
٥٤٣	٣-١- التصريح عن المؤشرات
٥٤٥	٤-١- الإدخال والإخراج
٥٥٠	٥-١- استدعاء التتابع
٥٥٩	٦-١- تابع التبديل (swap)
٥٦١	٧-١- الفرز الفقاعي
٥٦٣	٨-١- المؤشرات والمصفوفات
٥٧٤	٩-١- مصفوفة المؤشرات
٥٧٥	١٠-١- مؤشرات المؤشرات
٥٧٧	١١-١- مؤشرات التتابع

الفصل الحادي عشر

السلاسل

٥٨٧	١-١١- مقدمة
٥٨٧	١-١١-٢- أساسيات السلاسل المحرفية
٥٩٢	١-١١-٣- توابع مكتبة السلاسل المحرفية
٥٩٢	١-١١-٣-١- نسخ سلاسل المحارف
٥٩٤	١-١١-٣-٢- لصق سلاسل المحارف
٥٩٧	١-١١-٣-٣- المقارنة بين سلاسل المحارف
٦٠٠	١-١١-٣-٤- تقسيم سلاسل المحارف إلى وحدات token
٦٠٢	١-١١-٣-٥- حساب أطوال سلاسل المحارف
٦٠٥	المراجع الأجنبية
٦٠٧	المراجع العربية
٦٠٩	المصطلحات العلمية





مقدمة

إن توسع استخدام النظم الحاسوبية وتطورها في السنوات الأخيرة ، فرض وجود شروط محددة على بنية الحواسيب ، وذلك من خلال المعدات والبرمجيات والتطبيقات الحاسوبية ولغات البرمجة التي تم تصميمها للاستخدام على نطاق واسع ، لأن لها تطبيقات متنوعة في مختلف مجالات الحياة. حيث إن اتساع مجالات استخدام الحواسيب وانتشارها ، وزيادة تطور علوم صناعة البرمجيات قد أدى إلى ظهور العديد من اللغات البرمجية العالية المستوى والتي يستطيع الدارس أن يتعلمها وفقاً لحاجته واختصاصه .

الحاسوب لا يمكنه أن يعمل بكفاءة إلا بتوجيهه من طرف المبرمج ومعنى ذلك برمجته لتنفيذ متطلبات المبرمج . إذاً فبرامج الحاسوب هي مجموعة من الأوامر يقوم الحاسب بتنفيذها، وتتراوح هذه الأوامر أو التعليمات ابتداء من بعض الأوامر القليلة التي تؤدي مهمة بسيطة إلى قائمة أوامر أكثر تعقيداً ؛ التي من الممكن أن تحتوي جداول من المعطيات . فبرامج الحاسوب لا يتم كتابتها مباشرة بلغة الآلة لأن البرمجة بهذه اللغة عملية مملة جداً و خطأ بسيط يكلف وقتاً كثيراً لإيجاده و من ثم تصحيحه ما يؤخر عملية الإنتاج لدى المبرمج وعوضاً عن هذه اللغة ظهرت لغات البرمجة عالية المستوى ؛ التي يتم ترجمتها آلياً إلى لغة الآلة عن طريق برامج خاصة تسمى المفسرات و المترجمات.

فظهرت مجموعة من اللغات البرمجية عالية المستوى و وقريبة من لغة الإنسان مثل :

BASIC, FORTRAN ,Pascal, PROLOG,C\C++,Java ,C#,...

وفي هذا الكتاب نلقي الضوء على متركزات أساسية في البرمجة بلغة C++ وذلك من خلال البدء بفهم الخوارزميات وكيفية صياغتها وفق الطرق

المعروفة لصياغة الخوارزميات ومن ثم الانطلاق بكتابة البرامج اللازمة بلغة C++.

تُعد لغة C++ من لغات البرمجة عالية المستوى ، وفي الوقت نفسه قريبة من لغة التجميع ذات المستوى المحدود ، كما أنها تعد لغة برمجة إجرائية (يمكن كتابة برنامج يحتوي على إجراءات و توابع فقط) و لغة غرضية التوجه (البرنامج المكتوب عبارة عن صفوف و تستخدم الخواص المتاحة من التغليف و تعددية الأشكال و الوراثة والتركيب...). وهي لغة ناشئة من لغة C الذي قام بتطويرها Bjarne Stroustrup إلى لغة C++، وتشمل لغة C++ جميع مزايا لغة C بالإضافة إلى مزايا البرمجة غرضية التوجه .

ويصف هذا الكتاب بنية الحاسوب وتنظيمه والبرمجة بلغة C++ ، ويهدف إلى تعلم أساسيات بنية الحاسوب وأساسيات لغة C++ ، حيث يتدرج من المفاهيم البسيطة وينتهي بشرح معمق لأكثر مواضيع اللغة تعقيداً ، ويدعم الشرح بالأمثلة الواضحة والقوية في آن واحد . كما يلي كل فصل مجموعة من المسائل العامة التي تدرب الطالب على حل المسائل بنفسه وتعمق فهمه للمواضيع وتطبيقاتها العملية .

والهدف الأساسي لهذا الكتاب هو تغطية منهاج مقرر " مدخل إلى الحاسوب والبرمجة " في قسم هندسة الحواسيب والأتمتة لطلاب السنة الأولى. وقد تم وضع مادة هذا الكتاب بحيث تمكن مختلف القراء من الاستفادة منه في مختلف التطبيقات . ويتكون من أحد عشر فصلاً وسرد للمراجع الأجنبية والعربية والمصطلحات العلمية .

ويتضمن الفصل الأول البنية الأساسية للحاسوب وتنظيمه ، والهدف من هذا الفصل التعرف على المكونات الأساسية للحاسوب، وكما هو معروف فإن مكونات أي حاسوب لا تعمل بشكل مستقل منفصلة عن بعضها بعضاً ، بل تعمل معاً كنظام متكامل .

أما الفصل الثاني فيتناول أنظمة العد ، وبما أن نظام العد العشري لا يناسب الخاصية المادية أو الفيزيائية للعناصر المكونة للحاسوب الرقمي فقد تم تطوير أنظمة متنوعة أخرى للعد .

ويغطي الفصل الثالث البرمجيات ونظم التشغيل ، والهدف من هذا الفصل التعرف على نظم التشغيل وكذلك البرامج المعيارية التي يقوم مصنعو الحاسوب بإعدادها والتي تمكن المستخدمين من استغلال عمل الحاسوب على أفضل وجه .

ويغطي الفصل الرابع الشبكات الحاسوبية والانترنت ، والهدف من هذا الفصل التعرف على الشبكات الحاسوبية وأنواعها وطرق تصميمها ، حيث أصبح من النادر وجود حاسوب في شركة أو مؤسسة غير متصل بشبكة حواسيب ويعود السبب في ذلك إلى ما حققته هذه الشركات من فوائد تعود عليها من وجود هذه الشبكات والمشاركة في الأجهزة كالطابعات والراسمات وغيرها، وكذلك المشاركة في المعلومات التي تُعد العنصر الأهم لأي شركة.

والفصل الخامس خُصص للتعرف على مبادئ الخوارزميات وأنواعها وكيفية صياغتها من خلال بعض الأمثلة التطبيقية.

والفصل السادس يغطي أساسيات البرمجة في لغة C++، وذلك من خلال التعرف على الخطوات اللازمة لكتابة وتنفيذ برنامج بلغة C++ والقواعد الواجب إتباعها وأيضاً التعرف على أنواع المعطيات والعمليات الحسابية والمنطقية .

وُخُصص الفصل السابع للتعرف على بنى التحكم والتكرار وأنواعها و الفرق بينهما ومتى تستخدم واحدة دون الأخرى.

وُحْصِصَ الفصل الثامن للتعرف على المصفوفات أحادية البعد والمصفوفات ثنائية البعد وكيفية التصريح عنها والتعامل معها من حيث إدخال قيماً لعناصرها وطباعة هذه القيم وأيضاً كيفية استخدام المصفوفات لإجراء بعض العمليات الحسابية ، وأيضاً تمرير المصفوفات إلى التوابع كوسطاء لها .

ويعطي الفصل التاسع التوابع من حيث التصريح عنها وتعريفها وكيفية استدعائها لتنفيذ بعض المهام ، بالإضافة إلى عملية ربط التوابع مع المصفوفات .
وكما حُصِصَ الفصل العاشر للتعرف على المؤشرات وضرورة استخدامها وكيفية تمرير المصفوفات إلى التوابع باستخدام المؤشرات والعلاقة بين المؤشرات والعبارات المحرّفة وأخيراً مصفوفة المحارف والمؤشرات .

والفصل الحادي عشر والأخير حُصِصَ للتعرف على السلاسل المحرّفة والتوابع اللازمة للتعامل معها من خلال نسخ السلاسل ودمجها والمقارنة بين السلاسل والتوابع الأخرى .

ونأمل أن نكون قد وضعنا للقارئ المبادئ الأساسية للانطلاق في مجال بنية الحاسوب وتنظيمه ولغات البرمجة، وأسهمنا بتقديم المادة العلمية المتواضعة لكوادرنا الهندسية لتساعدكم في تطوير معلوماتهم ، بحيث تتماشى مع التطور الكمي والنوعي في مجال هندسة الحواسيب والأتمتة .
ولا يفوتنا أن نتقدم بالشكر والعرفان إلى جميع الزملاء الذين أسهموا في إبراز هذا العمل إلى حيز الوجود .

والله ولي التوفيق

المؤلفون

دمشق / أيلول / 2014

الجزء الأول
مدخل إلى الحاسوب

الفصل الأول
بنية الحاسوب وتنظيمه



البنية الأساسية للحاسوب وتنظيمه

١-١-١ تطور الحاسوب وأجياله

١-١-١-١ مقدمة

إن تطور الحواسيب الالكترونية منذ تطوير أول آلة حاسوبية ميكانيكية في عام 1632 وحتى بداية الحرب العالمية الثانية كان تطوراً عادياً وكان بعضها ميكانيكياً والآخر تطوراً كهربائياً ولكن الحواسيب التي تم إنتاجها خلال فترة الجيل الأول تعد تحولاً كبيراً في استخدام المعدات الميكانيكية والكهربائية و تطوير أجهزة الكترونية مبرمجة .

ويمكن القول إن الحواسيب تطورت بشكل فعلي منذ بداية الخمسينات وحتى اليوم تطوراً هائلاً تجاوز كل التوقعات نتيجة لتطور الإلكترونيات الدقيقة Microelectronics بشكل كبير ، فظهرت أجيال متعاقبة من الحواسيب ، وكان آخرها حواسيب الجيل الرابع والتي تميزت باستخدام كبير للدارات المتكاملة الفائقة . إن كلمة computer مشتقة من الفعل compute ويعني " أحسب " ، وإن كلمة computer تعني " حاسب " ، ولقد قام مجمع اللغة العربية بتعريب هذه الكلمة اصطلاحاً فأطلق عليها لفظ "حاسوب"، وفيما يتعلق بالتعريف العلمي للحاسوب فيمكن أن يعرف بعدة أشكال.

١-١-٢ أجيال الحاسوب Computer Generation

لقد مرت الحواسيب الالكترونية بمراحل تطوير مهمة ، ويختلف تصميم الحاسوب وبناءه في كل جيل تبعاً للتطور التكنولوجي في صناعة الحواسيب ، ويمكن سرد هذه الأجيال مع ذكر أهم الأحداث التي تمت فيها على النحو التالي :

١ - حواسيب الجيل الأول :

تمتد فترة هذا الجيل بين منتصف الثلاثينيات ومنتصف الخمسينيات، واستخدمت فيها تقنية الصمامات المفرغة (Vacuum Tubes) ، وكانت كبيرة الحجم وثقيلة الوزن وسرعتها أبطأ بكثير من الحواسيب المستخدمة في الوقت الحالي و تستهلك طاقة كهربائية كبيرة ؛ بسبب استعمال الصمامات ، وبالتالي يتولد عنها كميات كبيرة من الحرارة وتستلزم استخدام وحدات تكييف لتخفيض درجة الحرارة العالية . ومن أهم الأحداث التي تمت خلال هذا الجيل :

- وضع العالم الإنكليزي تورينغ Turing عام 1936 نموذجاً رياضياً للآلات الرقمية سميت فيما بعد بآلة تورينغ .

- وضع العالمان الفرنسيان Couffignal و Valtat عام 1936 الأساس النظري لنظام العد الثنائي .

- تم إنتاج أول حاسوب الكتروني عام 1946 باسم ENIAC (Electronic Numerical Integrator And Calculator) يحتوي على أكثر من 18000 صماماً مفرغاً وكان كبير الحجم يزن حوالي 30 طناً .

وكانت عمليات الإدخال وعمليات البرمجة بالنسبة للحاسوب ENIAC عمليات معقدة ؛ لذلك بدأ التفكير بإنتاج حاسوب تسهل فيه عمليات البرمجة والإدخال ، ومن هنا جاء اقتراح العالم فون نيومان Von Neumann باستخدام مبدأ مجموعة الأوامر المخزنة (البرنامج المخزن) فتم تصميم حاسوب جديد عام 1951 وسمي بـ UNIVAC 1 (Universal Automatic Calculator 1) وكانت البرمجة فيه تعتمد على النظام الثنائي بلغة الآلة (1,0) .

ويُعد العالم فون نيومان أول من وضع المبادئ الأساسية لعمل الحواسيب والتي التي تستدعي أن يتم تخزين البرامج في ذاكرة الحاسوب بصورة مشابهة لتخزين المعطيات ، واستخدام نظام العد الثنائي في تمثيل المعلومات داخل الحاسوب ،

ومن حواسيب هذا الجيل IBM 700 وأجهزة CRC 102A من شركة NCR وغيرها . وتمتاز حواسيب الجيل الأول بما يلي :

- ✓ استهلاك كبير للطاقة الكهربائية .
- ✓ ذات حجم كبير .
- ✓ ذات سرعة متدنية لمعالجة البيانات مقارنة مع الحواسيب الحالية وتصل إلى (10 - 20 ألف عملية / ثانية) .
- ✓ بحاجة إلى أجهزة تبريد بسبب ارتفاع درجة الحرارة .
- ✓ تستخدم ذاكرة القلب المغناطيسي .
- ✓ البرمجيات المستخدمة : لغة الآلة ولغة الاختصارات .

٢ - حواسيب الجيل الثاني:

تمتد فترة هذا الجيل بين منتصف الخمسينيات و حتى منتصف الستينيات ، فقد أدى اكتشاف الترانزيستور إلى تطور كبير في صناعة الدارات الإلكترونية فتم استبدال الصمامات المفرغة بالترانزيستورات ما أدى إلى إنتاج حواسيب ذات أحجام أصغر ، واستهلاك للطاقة أقل ، وسرعة أكبر بكثير من حواسيب الجيل الأول .

وحدث تطور كبير في مجال البرمجيات سواء بالنسبة لنظم التشغيل أو في مجال البرمجيات و استخدام لغات البرمجة عالية المستوى مثل لغة فورتران (FORTRAN) . واستخدمت حواسيب هذا الجيل نظم تشغيل متعددة البرمجة (Multiprogramming Systems) ونظم المعالجة المتعددة (Multiprocessing Systems) حيث بدأت فكرة نظم التشغيل بنظم المعالجة الدفعية (Batch Processing) وكان هناك وقت ضائع بين نهاية عمل البرنامج وبداية البرنامج التالي ولا يتم تنفيذ البرنامج التالي إلا عندما ينتهي الحاسوب من تنفيذ البرنامج الأول (ولا يوجد في ذاكرة الحاسوب إلا عمل واحد

فقط) . وتم في هذا الجيل إنتاج وحدات الحساب والمنطق ووحدات تحكم أكثر تعقيداً ، ومن حواسيب هذا الجيل IBM 1400 و CDC 1004 .
وتمتاز حواسيب الجيل الثاني بما يلي :

- ✓ استهلاك قليل للطاقة الكهربائية مقارنة مع حواسيب الجيل الأول .
- ✓ ذات حجم صغير مقارنة مع حواسيب الجيل الأول .
- ✓ ذات سرعة عالية لمعالجة المعطيات مقارنة مع حواسيب الجيل الأول وتبلغ مئات الآلاف من العمليات / ثانية .
- ✓ تستخدم ذاكرة القلب المغناطيسي .
- ✓ البرمجيات المستخدمة : لغات عالية المستوى مثل : BASIC , FORTRAN الخ .

٣- حواسيب الجيل الثالث :

تمتد فترة هذا الجيل بين منتصف الستينيات و حتى منتصف السبعينيات . واستخدمت هذه الحواسيب في تصميمها الدارات المتكاملة IC (Integrated Circuits) وهي دارات متكاملة على شكل شرائح صغيرة من السليكون Silicon) .

الدارة المتكاملة IC : عبارة عن تجمع عدد كبير من العناصر الإلكترونية (عشرات الترانزستورات) في قطعة واحدة تسمى IC . وكانت هذه التقنية أكثر متانة وقل تكلفة وذات حاجة اقل للطاقة من تقنية الترانزستور وبالتالي فإن حواسيب هذا الجيل تميزت بأنها ذات حجم اقل وأكثر سرعة وأرخص قيمة. وخلال فترة هذا الجيل تم تطوير لغة البرمجة (Basic) في عام 1964 ، وظهرت فكرة البرمجة البنيوية (Structured Programming) وتم إنتاج أول حاسوب عملاق Supercomputer تحت اسم

(CDC6000) في عام 1964 . وتم تجميع عدد صغير من البوابات على شريحة واحدة وسمي هذا الإنتاج بالدارات المتكاملة صغيرة التكامل SSI (Small Scale Integration) ، ومع زيادة التطور العلمي وتقدم تقنية التصنيع تم التوصل إلى تجميع عدد أكبر من العناصر الإلكترونية على شريحة واحدة وسمي هذا الإنتاج بالدارات المتكاملة المتوسطة التكامل MSI (Medium Scale Integration) ، ومن حواسيب هذا الجيل 70 SPECTRA من شركة RCA و 360 / IBM من شركة IBM ، وتمتاز حواسيب الجيل الثالث بما يلي :

- ✓ ذات وثوقية عالية مقارنة مع حواسيب الجيل الثاني .
- ✓ ذات حجم صغير مقارنة مع حواسيب الجيل الثاني .
- ✓ ذات سرعة عالية لمعالجة المعطيات مقارنة مع حواسيب الجيل الثاني وتصل إلى مليون عملية / ثانية .
- ✓ ذات سعة كبيرة للذاكرة حوالي 8 مليون بايت (8 MB) .
- ✓ البرمجيات المستخدمة : لغات عالية المستوى ونظم التشغيل واستعمل فكرة الذاكرة الافتراضية ونظام تعدد البرامج ونظام تعدد المعالجات .

٤ - حواسيب الجيل الرابع :

ومع التطور العلمي وتقدم تقنية التصنيع وزيادة كثافة التكامل تم الوصول إلى درجة كثافة تكامل عالية جداً VLSI (Very Large Scale Integration) واستخدمت حواسيب الجيل الرابع في تصميمها الدارات المتكاملة الكبيرة LSI والدارات المتكاملة الكبيرة جداً VLSI . والتي تحتوي على آلاف العناصر الإلكترونية في الدارة المتكاملة الواحدة IC . وقد احتوت هذه الدارات على وحدات متكاملة كوحدة المعالجة والتي سميت بالمعالجات الميكروية Microprocessors وبعد ذلك تطورت الدارات المتكاملة الكبيرة والكبيرة جداً

ليصل عدد العناصر الإلكترونية التي تحتويها إلى ملايين وسميت بالدارات المتكاملة الفائقة ULSI .

وتم إنتاج وحدة التحكم ووحدة الحساب والمنطق في شريحة واحدة وتم تسمية هذه الشريحة بالمعالج الصغير Microprocessor ، وأنتج أول معالج من قبل شركة Intel تحت اسم 4004 في أوائل السبعينيات ، وفي عام 1974 أنتجت شركة Intel المعالج 8080 الذي يتعامل مع معطيات بطول 8bit ، وخلال فترة هذا الجيل وفي مجال نظم التشغيل ظهر نظام التشغيل DOS للحواسيب الصغيرة ، وظهر خلال فترة هذا الجيل العديد من اللغات العالية المستوى مثل C ، Pascal... الخ ، و تم ظهرت شبكات الحاسوب وما تبع ذلك من تطوير نظم تشغيل الشبكات ، وتمتاز حواسيب الجيل الرابع بما يلي :

✓ ذات حجم صغير جداً ، وإنتاج الحاسوب الشخصي PC .
✓ ذات سرعة عالية جداً تصل إلى مئات الملايين من العمليات الحسابية في ثانية .

✓ الذاكرة : استخدمت ذاكرات مصنوعة من أنصاف النواقل منها :

١. الذاكرة ROM (Memory Read Only)

٢. لذاكرة RAM (Random Access Memory) .

✓ البرمجيات :

١. لغات عالية المستوى مثل : Pascal, Java, C++

٢. نظم التشغيل مثل :

Windows(98, 2000,xp,7, NT), MS DOS, Unix, Linux .

١-٢- تعريف الحاسوب

يمكن تعريف الحاسوب (computer) بعدة أشكال كما يلي :

- ❑ الحاسوب آلة إلكترونية قابلة للبرمجة تقوم باستقبال المعطيات التي يتم إدخالها بواسطة وحدات الإدخال حيث يتم تخزينها وإجراء العمليات الحسابية والمنطقية عليها بواسطة التعليمات المكونة للبرامج وذلك للحصول على النتائج وإخراجها من وحدات الإخراج المختلفة.
- ❑ الحاسوب آلة إلكترونية يمكن برمجتها لكي تقوم بمعالجة المعطيات وتخزينها واسترجاعها وإجراء العمليات الحسابية والمنطقية عليها .
- ❑ الحاسوب آلة حساب إلكترونية سريعة تستقبل معلومات دخل رقمية وفقاً لمنهجية تحددها تعليمات برمجية مخزنة ثم تصدر نتائج المعالجة إلى الوسط الخارجي .
- ❑ الحاسوب مجموعة من الأجهزة أو الوحدات المستقلة Hardware تؤدي كل منها وظيفة معينة وتعمل هذه الوحدات فيما بينها بأسلوب متناسق من خلال البرمجيات Software ، ويوضح الشكل (١-١) نظام الحاسوب .



الشكل (١-١) - يوضح نظام الحاسوب

فالحاسوب يقوم بأربع عمليات أساسية هي:

- استقبال المعطيات المدخلة.
 - معالجة المعطيات إلى معلومات.
 - تخزين المعطيات المدخلة .
 - إخراج المعلومات .
- ويبين الشكل (٢-١) العمليات الأساسية الأربع التي يقوم بها الحاسوب .



الشكل (٢-١) - يوضح العمليات الأساسية الأربع التي يقوم بها الحاسوب

ويتميز الحاسوب بالخواص التالية :

- ١- القدرة على تخزين واسترجاع المعطيات : كالأرقام ، والحروف الهجائية ، والصور .
- ٢- إمكانية معالجة المعطيات وإجراء العمليات الحسابية عليها : كالجمع ، والطرح ، والقسمة ، والضرب ، وإجراء العمليات المنطقية كالمقارنة بين قيمها .
- ٣- إمكانية برمجة الحاسوب أي إعطاء تعليمات وأوامر للحاسوب لكي يقوم بتنفيذ أعمال محددة .

٣-١ - المكونات الأساسية للحاسوب

يتكون النظام الحاسوبي من قسمين رئيسيين هما :

- العتاد (Hardware) .
- البرمجيات (Software) : سيتم شرحه في الفصل الثالث .

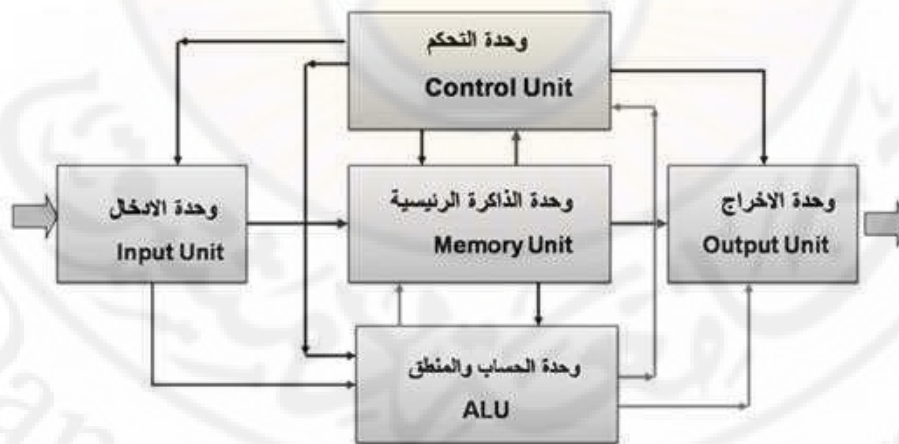
أما المكونات الأساسية لنظام الحاسوب حسب فان نيومان فتتكون من خمس وحدات أساسية هي :

١. وحدة الإدخال
٢. وحدة الإخراج
٣. وحدة الحساب والمنطق
٤. وحدة التحكم
٥. وحدة الذاكرة الرئيسية

ويوضح الشكل (٣-١) المكونات الأساسية للحاسوب ، أما الشكل (٤-١) يوضح المخطط الصندوقي لمكونات الحاسوب ومبدأ عمله .



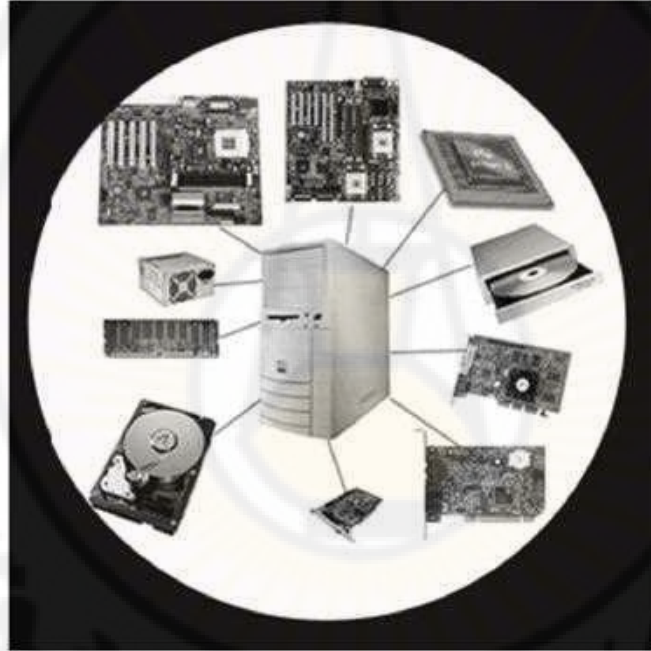
الشكل (٣-١) - يوضح المكونات الأساسية للحاسوب



الشكل (٤-١) - يوضح المخطط الصندوقي لمكونات الحاسوب ومبدأ عمله

العتاد Hardware :

هي الأجزاء التي يمكن لمسها ورؤيتها من مجموعة الأجهزة داخل أو خارج صندوق النظام ، ويمكن تصنيف أجهزة العتاد إلى قسمين رئيسيين هما لوحة النظام ، والوحدات المحيطية لنظام الحاسوب ، ويبين الشكل (١-٥) مكونات نظام الحاسوب داخل الصندوق system case .



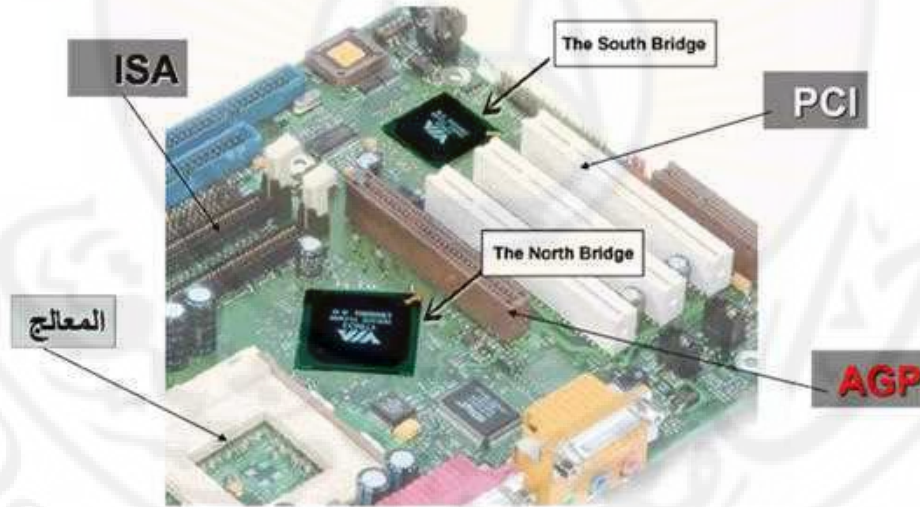
الشكل (١-٥) - نظام الحاسوب

١-٤ - لوحة النظام (System Board)

وتسمى أيضاً باللوحة الأم (Mother Board) ، وهي اللوحة الأساسية ، وتقع داخل صندوق النظام وتتكون من مجموعة كبيرة من القطع الإلكترونية

المتبنة عليها ويتصل بها جميع أجزاء الحاسوب ، ويتم تغذيتها بالطاقة الكهربائية (Power Supply) ، ويطلق على كل قطعة من هذه القطع الإلكترونية بالدارة المتكاملة ، ويتم تصنيع هذه الدارات المتكاملة عن طريق وضع عدد كبير من الدارات الإلكترونية داخل حجم صغير من مادة السيلكون ، والتي تحوي في داخلها على الدارات الإلكترونية أو الكهربائية كالترانزستورات والمقاومات والمكثفات وغيرها ، ويوضح الشكل (٦-١) اللوحة الأم ، وتتكون لوحة النظام من الوحدات التالية :

- وحدة المعالجة المركزية (CPU) Central Processing Unit.
- وحدة الذاكرة الرئيسية Main Memory Unit.
- وحدة الترابط مع الحاسوب (Interface Unit) أو الواجهة .



الشكل (٦-١) - يوضح مكونات اللوحة الأم

١-٤-١ - وحدة المعالجة المركزية (CPU)

إن وحدة المعالجة المركزية بمثابة العمود الفقري للحاسوب الشخصي ، و بدون هذه الوحدة يكون الجهاز مجرد آلة صماء غير قادر على القيام بأي عمل مفيد ، ويطلق على هذه الوحدة في الحاسوب الشخصي بالمعالج الصغير (Micro-processor) ، وهو عبارة عن دائرة إلكترونية عالية التكامل بمعالجة المعطيات وتنفيذ أوامر البرمجة التي ترسل إليها من الدارات الإلكترونية الأخرى في لوحة النظام . لذلك فإن هذه الوحدة هي وحدة التشغيل المركزية ، وهي الوحدة التي تتحكم في نظام الحاسوب في كل عملياته، و الجزء الأكثر أهمية لاحتوائها على جميع الإمكانيات الضرورية لإنجاز مهام المعالجة وتداول المعطيات، بالإضافة إلى رقابة وتوجيه جميع الوحدات والأجهزة الأخرى وتنسيق العمل بينها ، ويبين الشكل (٧-١) وحدة المعالجة CPU .



الشكل (٧-١) - يوضح وحدة المعالجة CPU

ومن المهام والوظائف التي تقوم بها وحدة CPU كما يلي:

- تقوم وحدة المعالجة المركزية بتنفيذ جميع العمليات الخاصة بالتشغيل .
- تقوم بعمليات المقارنة المنطقية وكذلك تقوم بالعمليات الحسابية التي تكون موجودة في البرنامج المراد تنفيذه .
- تقوم بتنظيم نقل المعطيات من/و إلى الوحدات المساعدة حيث تتم العملية باستقبال المعطيات وإرسال تلك المعطيات إلى وحدات محددة كما هو مطلوب وذلك في الوقت المناسب.
- تقوم بتمرير المعطيات من/و إلى الذاكرة الرئيسية Main Memory.

وتتكون وحدة المعالجة المركزية من :

- وحدة التحكم Control Unit.
- وحدة الحساب والمنطق Arithmetic and Logic Unit.
- المسجلات الداخلية Registers .

١ - وحدة التحكم Control Unit

إن عمل وحدة التحكم مهم بالنسبة لوحدة المعالجة المركزية و لما تقوم به من تنسيق جميع أعمال الحاسوب ، فتقوم بمهام الإشراف والتحكم والتوجيه والضبط لجميع العمليات التي تقوم بها وحدات الحاسوب الداخلية، وتعمل هذه الوحدة كالقلب النابض لوحدة المعالجة المركزية والحاسوب بشكل عام . وهي التي تقوم بمراقبة وتوجيه جميع الوحدات الأخرى وتعمل على توليد النبضات الزمنية اللازمة لتحقيق تسلسل تنفيذ العمليات ، وتقوم هذه الوحدة بتنظيم وتنسيق عمل الوحدات في الحاسوب دون أي خلل وبأسلوب متناسق ومنظم . وتحصل على التعليمات من أماكن تخزين البرامج في الذاكرة الرئيسية ، وتتكون هذه الوحدة من الوحدات الجزئية التالية :

• وحدة العنوان Address Unit

• وحدة التعليمات Instruction Unit

• وحدة التوقيت والتتابع الزمني Timing Unit

وتقوم هذه الوحدات الجزئية بالنسبة لجلب وتنفيذ التعليمات بالمهام التالية:

١. تقوم بتحديد عنوان التعليمة المراد تنفيذها ويتم وضع هذا العنوان (عنوان بداية البرنامج) على مسرى العنوان Address Bus .
٢. تقوم بجلب محتوى الذاكرة حسب العنوان المذكور ووضعه على مسرى المعطيات Data Bus
٣. تقوم بتفسير التعليمة من أجل إصدار الأوامر المناسبة لتنفيذها في وحدة ALU

٤. تقوم بالإشراف على تخزين النتائج في أماكن تخزينها.
٥. يتم زيادة عداد البرنامج PC بمقدار مساوٍ لطول التعليمة وذلك للانتقال إلى عنوان التعليمة التالية في البرنامج الذي يجري تنفيذه ، وفيما بعد يتم تكرار العمليات نفسها المذكورة أعلاه على جميع تعليمات البرنامج.

ومن المهام والوظائف التي تقوم بها وحدة التحكم :

- ✓ تأهيل وحدات الإدخال والإخراج المناسبة حين الحاجة إليها.
- ✓ قراءة وتفسير تعليمات البرنامج ومن ثم إصدار الإشارات اللازمة للوحدات المختلفة لتنفيذ هذه التعليمات.
- ✓ التحكم بتدفق المعطيات والتعليمات من وإلى الذاكرة الرئيسية و متحكمات وحدات الإدخال والإخراج .
- ✓ توجيه العمليات داخل وحدة المعالجة المركزية CPU .
- ✓ تنظيم وتنسيق عمل الوحدات في الحاسوب .

٢ - وحدة الحساب والمنطق Arithmetic and Logic Unit

هي الوحدة التي تقوم بتنفيذ العمليات الحسابية والمنطقية ، حيث تقوم باستقبال المعطيات إما من وحدة الإدخال مباشرة لمعالجتها أو من وحدة الذاكرة الرئيسية وفيما بعد يتم إرسال نتائج المعالجة إلى وحدة الذاكرة الرئيسية لتخزينها أو إرسالها إلى وحدات الإخراج لإظهارها . وتتكون هذه الوحدة من مجموعة من السجلات المساعدة وذلك لحفظ العناوين والنتائج الجزئية ، وتتأثر وحدة الحساب والمنطق بسجل المؤشرات Flags الذي توضع فيه مؤشرات تصف خرج وحدة الحساب والمنطق ، وتقوم هذه الوحدة بالعمليات التالية :

✓ بجميع العمليات الحسابية (الجمع - الطرح - الضرب - القسمة)

✓ بجميع العمليات المنطقية (AND - OR - EX OR الخ)

✓ بعمليات المقارنة والنقل بين المسجلات

وتحتوي وحدة الحساب والمنطق على عدد كبير من المسجلات Registers و عدد من دارات الجامع Adder وعدد من العدادات Counters ، ومن مهام الجامع Adder القيام بتنفيذ جميع العمليات الحسابية المختلفة (+ ، - ، * ، /) وكذلك عمليات المقارنة (> , < , = , <= , >=) وذلك عندما تكون المعطيات التي يتعامل معها بالشكل الثنائي Binary ، وتقوم هذه الوحدة بالمعالجة الفعلية للمعطيات تحت إشراف وحدة التحكم ، حيث تنتقل المعطيات أثناء دورة التنفيذ إلى أحد العدادات أو إلى أكثر من عداد وهناك يتم تداول المعطيات بواسطة دارات الجامع لكي يتم استخراج النتائج التي يمكن تخزينها بإحدى العدادات لجمع أو تحويل هذه النتائج إلى أمكنة أخرى بوحدة التخزين .

٣- السجلات الداخلية Registers

عبارة عن مواقع تخزين خاصة عالية السرعة ، تخزن المعطيات و المعلومات بشكل مؤقت لاستخدامها من قبل وحدة الحساب والمنطق ALU. وتحتوي وحدة المعالجة على أنواع مختلفة من السجلات ، كل منها مختص بتخزين نوع معين من المعطيات. وتتكون من سجلات للأغراض العامة GBR يستخدمها المعالج من أجل المتحولات والقيم التي يعالجها، وسجلات للأغراض الخاصة SBR كمسجل عداد البرنامج PC ، وسجل البرنامج الذي يحتفظ بقيمة عنوان الذاكرة ومنها سجل أساسي يسمى المراكم Accumulator، ويتم دمج الأقسام الثلاثة المكونة لوحدة المعالجة المركزية جميعها في دائرة متكاملة واحدة.

١-٤-٢ وحدة الذاكرة الرئيسية Main Memory Unit

تعد وحدة الذاكرة الرئيسية من الأجزاء الرئيسية بجهاز الحاسوب ، وتتكون من مجموعة من الدارات الالكترونية المثبتة على لوحة نظام الحاسوب ، وتقوم بمهام الاحتفاظ بالمعطيات والأوامر التي يحتاجها المعالج عند إجراء العمليات المختلفة ، وإرسالها للمعالج عند طلبها ، وتحتفظ بالمعطيات الأساسية المطلوبة لتشغيل جهاز الحاسوب أو المتطلبات اللازمة لعمل نظام التشغيل ، ويتم في هذه الوحدة تخزين النتائج الوسيطة والنهائية لكل من المعطيات والتعليمات و تخزين أيضاً البرامج ليتم تنفيذها فيما بعد ، وترتبط هذه الوحدة مباشرة بوحدة المعالجة المركزية (CPU) وتتألف من مجموعة من المواقع وكل موقع مكون من عدد من الخلايا الثنائية (bits) وهي مواقع لتخزين البرامج والمعطيات التي يتم معالجتها .

تعمل الذاكرة الرئيسية وكأنها تحتوي على مجموعة كبيرة من صناديق التخزين بشكل مشابه لصناديق البريد ، حيث تستطيع وحدة التحكم الوصول إلى أي صندوق من هذه الصناديق ووضع عنواناً عليه تطلب فتحه لتخزين معطيات أو تعليمات فيه ، وجميع صناديق التخزين متساوية ، وتنقسم إلى عدد من الخانات وتسمى كل خانة بالبت (bit) ، وهي أصغر وحدة تخزين في الحاسوب ، وتأخذ إحدى القيمتين : صفر أو واحد . وعند وضع ثماني خانات نحصل على ما يسمى بالبايت (Byte) الذي يتسع لتخزين محرف واحد من المعطيات.

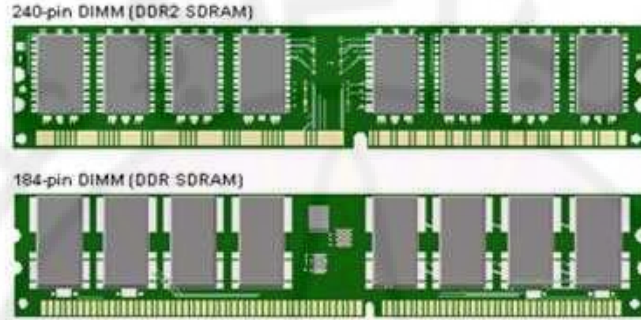
إن الذاكرة الرئيسية للحاسوب تُعد الجزء المهم الذي يتم فيه تخزين المعطيات ومعالجتها وتخزين نتائجها بصفة مؤقتة حسب ما تتطلبه الحاجة، وكذلك يمكن تخزين البرامج المراد تنفيذها في أي وقت. وتعد سعة الذاكرة الرئيسية من أهم المواصفات التي يتمتع بها الحاسوب، وكما أن هذه السعة تُعد عنصراً مهماً في تحديد قدرات الحاسوب، وتحدد سعة ذاكرته الرئيسية بكمية المحارف (أحرف + أرقام + رموز) التي يمكن تخزينها في تلك الذاكرة.

وتتميز الذاكرات بالإضافة إلى سعتها وطول كلمتها بمقدارين هما زمن النفاذ Access Time الذي يُعرف بأنه الفاصل الزمني ما بين لحظة تطبيق عنوان الموقع المراد قراءته ولحظة ظهور هذا المحتوى على خطوط المعطيات ، وهذا الزمن ثابت بالنسبة لكل مواقع الذاكرة . وزمن الكتابة Write Time الذي يُعرف على أنه الزمن اللازم لكتابة المعطيات في أي موقع في الذاكرة بعد وضع العنوان على خطوط العنونة . وفيما يلي سنستعرض بعض أنواع الذاكرات :

١ - ذاكرة الوصول العشوائي Random Access Memory (RAM)

هي دائرة إلكترونية متكاملة تستخدم للاحتفاظ بالمعلومات مؤقتاً أثناء تشغيل الحاسوب ، ويتم حذف هذه المعلومات عند قطع التيار الكهربائي عن الحاسوب . وهذا الجزء يكون تحت تصرف مستخدم الحاسوب حيث تحتفظ بالبرامج

والمعطيات التي يدخلها مستخدم الحاسوب أثناء التشغيل ، أو يتم إخراجها لوحدة الإخراج والتخزين ، و كلما زادت سعة الذاكرة RAM . أمكن للحاسوب التعامل مع برامج كبيرة وحجم أكبر من المعطيات، ويبين الشكل (٨-١) الذاكرة RAM .



الشكل (٨-١) - يوضح الذاكرة RAM

تخزن في الذاكرة RAM البرامج والمعطيات التي يدخلها المستخدم ، وكذلك نتائج معالجة المعطيات حسب البرامج المدخلة . بمعنى أن للمستخدم كامل الصلاحية للتحكم في محتويات هذا الجزء من الذاكرة ، وتتميز الذاكرة RAM بما يلي :

- ✓ تُعد ذاكرة قراءة وكتابة .
- ✓ تُعد ذاكرة قراءة وكتابة بها لأكثر من مرة.
- ✓ تُعد ذاكرة مؤقتة لأنها لا تحتفظ بالمعلومات بعد انقطاع التغذية عنها .
- ✓ تستخدم لتخزين البرامج والمعلومات التي يحتاجها المستخدم أثناء العمل على الحاسوب .
- ✓ ذات حجم كبير تصل إلى 4GB أو أكثر .

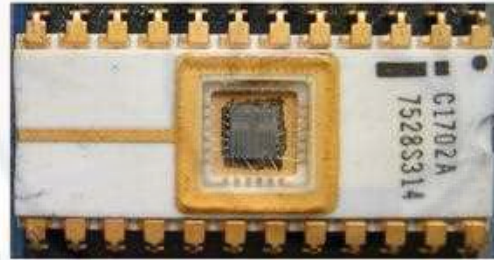
ثمة نوعان من الذاكرة RAM:

- ١- ذاكرة ساكنة Static RAM ويرمز إليها بـ SRAM .
- ٢- ذكرات ديناميكية Dynamic RAM ويرمز إليها بـ DRAM .

الذاكرات الديناميكية وهي أرخص من الذاكرات الساكنة ، وذات زمن نفاذ أعلى نسبياً ، والذاكرة الساكنة تحافظ على محتواها مادامت هناك تغذية كهربائية مطبقة عليها ، بينما تفقد الذاكرة الديناميكية محتواها بعد فترة قصيرة ، وتحتاج لإعادة إنعاش للاحتفاظ بالمعطيات (Refresh) .

٢ - الذاكرة ROM (Read Only Memory)

تقوم الذاكرة ROM بالاحتفاظ الدائم بالمعلومات ، ومن خلالها يتم الاحتفاظ بالمعلومات الأساسية التي يحتاجها الحاسوب لبدء تشغيله ، مثل موقع وجود نظام التشغيل وعناوين بوابات الإدخال والإخراج وغيرها . ويتم تخزين البرامج عليها من قبل الشركة الصانعة وبذلك تكون ضمن وحدات جهاز الحاسوب ، ويبين الشكل (٩-١) الذاكرة ROM .



الشكل (٩-١) - يبين الذاكرة ROM

وتتميز الذاكرة ROM بما يلي :

- ✓ تُعد ذاكرة قراءة فقط .
- ✓ يتم القراءة منها لأكثر من مرة ولا يمكن التعديل عليها أو الإضافة أو الحذف منها كلياً أو جزئياً أو الكتابة عليها .
- ✓ تُعد ذاكرة دائمة ، لأنها تحتفظ بالمعلومات بعد انقطاع التغذية عنها .

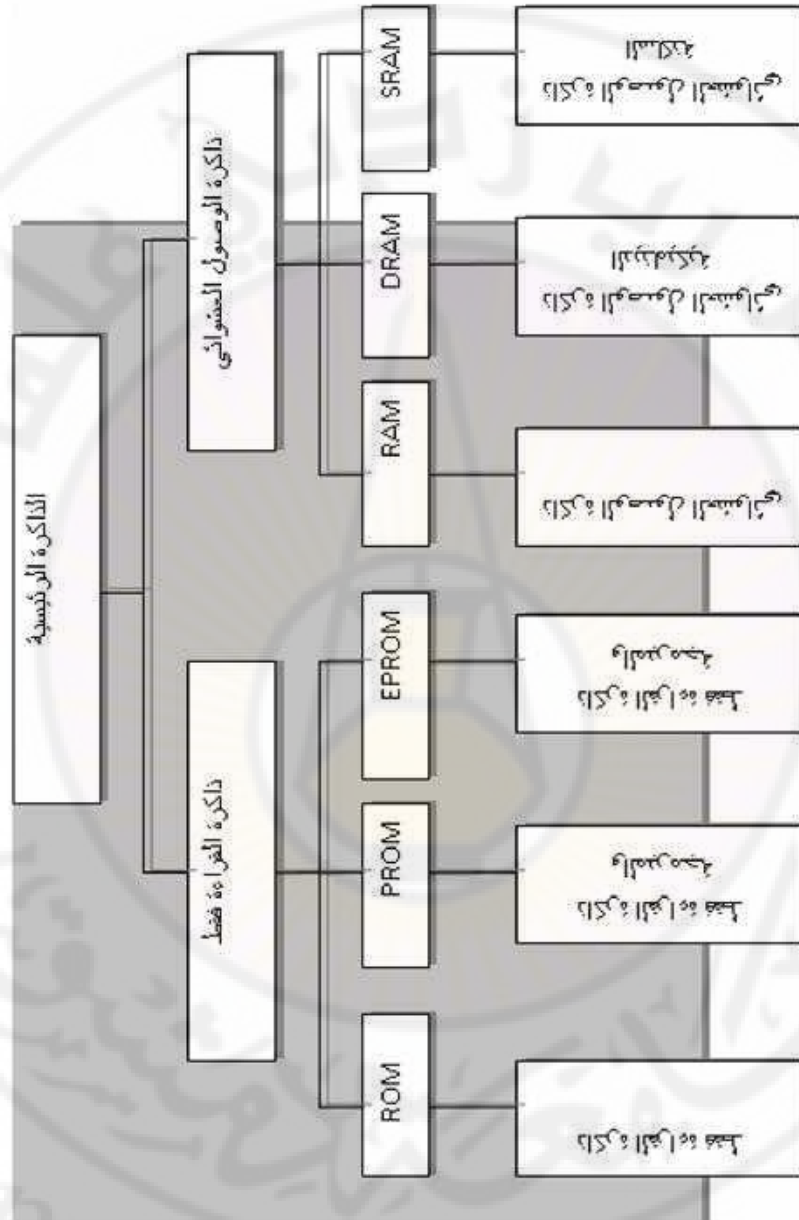
- ✓ سعتها صغيرة مقارنة مع الذاكرة RAM .
- ✓ تستخدم هذه الذاكرة لتخزين البرامج والمعلومات الثابتة مثل برنامج إقلاع الحاسوب.

٣- الذاكرة PROM (Programmable ROM)

- وهي ذاكرة مشابهة للذاكرة ROM ولكن يتم برمجتها من قبل المستخدم لمرة واحدة فقط، ويتم الكتابة عليها ، بعد تصنيعها وذلك باستخدام جهاز مخصص لبرمجة ذاكرات PROM ، وتتميز الذاكرة PROM بما يلي :
- ✓ تُعد ذاكرة دائمة .
 - ✓ يمكن برمجتها من قبل المستثمر ولمرة واحدة فقط .
 - ✓ تُعد ذاكرة قراءة لأكثر من مرة .
 - ✓ لا يمكن إعادة برمجتها مرة أخرى أو تعديل البرامج عليها .

٤- الذاكرة EPROM (Erasable PROM)

- وهي ذاكرة مبرمجة وقابلة للمحي ، ويستطيع المستخدم محي (إزالة) محتواها وإعادة برمجتها ، وهي ذاكرات يمكن إعادة الكتابة فيها بعد محيها ، ويتطلب هذا النوع من الذاكرات جهاز خاص لبرمجتها يسمى مبرمج EPROM ، ويتم محي محتواها بالأشعة ما فوق البنفسجية (Ultraviolet Light) ، ولا يمكن تغيير محتواها أو برمجتها خلال استخدامها في النظام ، ولا يمكن محي كلمة واحدة فقط، وبعد محي الذاكرة تكون جميع خلايا التخزين بحالة (1) منطقي ، وعند البرمجة يتم فقط تخزين (0) منطقي في كل خلية من خلايا الكلمة المراد برمجتها ، ويبين الشكل (١-١٠) أنواع الذاكرات.



الشكل (١٠-١) - يبين أنواع الذاكرات

ومن أهم التطبيقات ومجالات استخدام الذاكرات EPROM يكون في تطوير نظم الحواسيب الميكروية ، حيث يتم وضع برامج النظام في ذاكرات EPROM للتأكد من صحة عمل النظام أو إجراء التعديلات المناسبة ، وبعد ذلك يمكن نقل هذا البرنامج إلى ذاكرات (ROM) وذلك لتوفر مثل هذه الذاكرات ذات نفس أرجل الذاكرات EPROM . إن وجود ذاكرات من النوع ROM مكافئة للذاكرات من النوع EPROM يعطي منتجي نظم الحواسيب الميكروية إمكانية استخدام ذاكرات من النوع ROM والتي تشكل تكاليفها عند الإنتاج بكميات كبيرة ضئيلة بالنسبة لتكاليف الذاكرات من نوع EPROM .

٥ - ذاكرة الكاش Cache Memory

هي ذاكرة تتصل بالمعالج (وحدة المعالجة المركزية CPU) وتتميز بسرعة عالية جداً وتخزن عليها المعطيات و البرامج المستخدمة بكثرة من قبل المستخدم ما يوفر الزمن عند استدعائها من الذاكرة الرئيسية و بالتالي زيادة الإنتاجية وعادة ما تكون هذه الذاكرة بسعات مختلفة تصل إلى 1GB أو أكثر .

ومبدأ عمل ذاكرة الكاش يقوم على أنه يتم استقبال طلب قراءة Read من وحدة المعالجة المركزية CPU ، ويتم نقل كتلة (Blok) من كلمات الذاكرة التي تحتوي على الموقع المحدد إلى ذاكرة الكاش وبالتالي يتم قراءة محتويات هذه المواقع مباشرة منها ، ويتم تحديد العلاقة بين كتل الذاكرة الرئيسية وذاكرة الكاش بتابع الإسقاط Mapping Function.

- عندما تكون ذاكرة الكاش ممثلة وتم طلب كلمة ذاكرة غير موجودة فيها تقرر دارات تحكم الكاش أي كتلة يجب إزالتها من أجل وضع الكتلة الجديدة التي تحتوي على الكلمة المطلوبة ، وتسمى مجموعة القواعد التي تقوم باتخاذ هذا القرار بخوارزمية الاستبدال Replacement Algorithm .

- وعندما لا تكون الكلمة المعنونة في عملية القراءة Read موجودة في الكاش يتم نسخ كتلة من الكلمات التي تحتويها من الذاكرة الرئيسية إلى الكاش ، ثم ترسل الكلمة المطلوبة إلى وحدة المعالجة المركزية CPU ، وكلما زاد حجم ذاكرة الكاش أمكن تزويدها بالمزيد من المعطيات التي يحتاجها المعالج وبالتالي تقليل حدوث حالات انتظار أثناء تشغيل البرامج .

من مميزات

- ✓ متصلة بالمعالج CPU
- ✓ سرعتها عالية مقارنة بذاكرة RAM
- ✓ تقسم الذاكرة إلى مجموعة كتل Blocks مما يجعل تخزين وحذف المعطيات أسرع من الـ RAM إلا أنها أغلى ثمناً.
- ✓ تخزن البرمجيات و المعطيات المستخدمة بكثرة وبالتالي يسرع عمل الحاسوب
- ✓ سعتها عادة: 1 MB, 512KB, 256 KB, 128 KB ، أو أكثر .

٦- ذاكرة الـ Flash (Flash Memory)

الذاكرة Flash نوع من أنواع الذاكرة الدائمة (غير المتطايرة) مثل ROM إلا أنها تخزن المعطيات في مجموعة كتل Blocks ، ويتم التخزين و المسح في الكتلة بحركة واحدة تدعى Flash ما يجعلها أسرع من الذاكرة RAM إلا أنها أغلى ثمناً. وتستخدم الآن في تخزين نظام الإدخال /الإخراج الأساسي الخاص بالحاسوب BIOS وهو عبارة عن برنامج يتم تحميله عند تشغيل الحاسوب للتعرف على وحدات الإدخال والإخراج المرتبطة معه. كما تستخدم أيضاً في الحواسيب المحمولة و الطابعات و الكاميرات الرقمية، والأجهزة الخليوية.

من مميزات

- ✓ ذاكرة دائمة
- ✓ تستخدم في تخزين نظام الـ BIOS
- ✓ تقسم الذاكرة إلى مجموعة كتل Blocks مما يجعل تخزين وحذف المعطيات أسرع من الـ RAM إلا أنها أغلى ثمناً.
- ✓ تستخدم في الحواسيب المحمولة و الكاميرات الرقمية و الأجهزة الخلوية.

١-٤-٣ - وحدات الترابط مع الحاسوب

Interface Units

تنظم هذه الوحدات عملية تبادل المعطيات بين وحدة المعالجة المركزية و وحدة الذاكرة الرئيسية الموجودة على لوحة نظام الحاسوب أو المتصلة بها ، وتتكون وحدة الترابط من المكونات التالية :

١ - مسرى المعطيات (Data Bus)

يتكون هذا المسرى من مجموعة من الأسلاك والدارات الالكترونية التي تربط بين الدارات الالكترونية المختلفة كدارة المعالج ودارة الذاكرة الرئيسية ، ويتم عن طريق هذا المسرى مرور الإشارات والنبضات الكهربائية التي تمثل المعطيات ما بين الوحدات المختلفة داخل لوحة النظام (اللوحة الأم) أو إلى وحدات التخزين والإدخال والإخراج خارج لوحة النظام .

٢ - شقوق وبطاقات التوسيع (Expansion Cards and Slots)

ينتهي مسرى المعطيات على النظام بتقوُب (شق) مستطيلة ، ويمكن إدخال وتثبيت بطاقات داخلها تسمى بالشقوق (Slots) و التي هي عبارة عن

فتحات مستطيلة الشكل موجودة على لوحة النظام ، ويتم توصيل بطاقات التوسيع عليها ، وبالتالي يتم الاتصال بين دارات لوحة النظام والدارات الالكترونية على بطاقة التوسيع من خلال الشق (النقب) ، ويستفاد من ثقوب المسرى للقيم بزيادة إمكانيات جهاز الحاسوب ، و يمكن على سبيل المثال توسيع الذاكرة عن طريق إضافة بطاقة تحمل دارات الذاكرة الإضافية وتثبيتها في إحدى ثقوب التوسيع ، أما بطاقات التوسيع فهي بطاقات يتم تثبيتها في ثقوب التوسيع ، ويتم توصيل بعض ملحقات الحاسوب بها ، ومن أنواع هذه البطاقات :

- بطاقة الصوت (Sound Card) : والتي تستخدم لتوصيل الميكروفون والسماعات .
- بطاقة الرسوم (Graphic Card) : تستخدم لتوصيل الشاشة .
- بطاقة التلفزيون (TV Tuner) : عن طريقها يمكن استقبال الإرسال التلفزيوني .
- بطاقة الشبكة (LAN Card) : تستخدم لربط جهاز الحاسوب بالحواسيب الأخرى في الشبكة .
- بطاقة فاكس / مودم (Modem / Fax) : تستخدم لربط جهاز الحاسوب بالحواسيب الأخرى البعيدة عبر خط الهاتف .

٣ - بوابات الحاسوب (المنافذ Ports)

تستخدم لتوصيل لوحة النظام أو بطاقات التوسيع بالأجهزة الواقعة خارج صندوق نظام الحاسوب ، مثل لوحة المفاتيح أو الفأرة أو شاشة العرض أو سماعات الصوت أو الطابعة ، حيث يتم توصيل الأسلاك المثبتة في مسرى المعطبات على لوحة النظام أو أسلاك بطاقات التوسيع إلى فتحات ظاهرة يطلق عليها بوابات (منافذ) الجهاز (Port) ، وتثبت هذه الفتحات عادة في الجانب

الخلفي من إطار صندوق الجهاز ، وتختلف أشكال هذه الفتحات بحسب نوع التوصيل عليها ، ومن أنواع بوابات (المنافذ) الحاسوب :

• البوابة التسلسلية Serial Port

وهي البوابة التي ترسل المعطيات (الإشارات الكهربائية) بالتسلسل خانة ثنائية تلو الأخرى ، وتستخدم عادة لربط الحاسوب بأجهزة الاتصال الهاتفي عن طريق كبل للاتصال يتم تثبيت طرف منه في البوابة بالجهاز والطرف الآخر على جهاز الاتصال الهاتفي . سابقاً كانت تستخدم البوابة التسلسلية لتوصيل الفارة ولوحة المفاتيح ، أما الآن فأصبحت تستخدم لتوصيل بعض الملحقات مثل عصا التحكم بالألعاب ، والمودم الخارجي .

• البوابة (PS/2)

تم تطوير هذه البوابة من قبل شركة IBM لكي تحل مكان البوابة التسلسلية ، وهي مخصصة لتوصيل جهاز الفارة ولوحة المفاتيح بلوحة النظام .

• البوابة التفرعية Parallel Port

تسمح هذه البوابة بمرور مجموعة من المعطيات (الإشارات الكهربائية) معاً على الأسلاك المرتبطة بها ، وبالتالي يمكن مرور ثماني خانات ثنائية أو ست عشرة خانة أو أكثر معاً في الوقت نفسه عبرها ، وذلك باستخدام كبل يربط طرف منه بجهاز الحاسوب والطرف الآخر بالملحقات المراد وصلها ، وتعد البوابة التفرعية أسرع من البوابة التسلسلية ، وتستخدم لتوصيل الملحقات مثل الطابعات وبعض أنواع الماسحات الضوئية .

- **المسرى التسلسلي العام USB1.1 (Universal Serial Bus)**

إحدى البوابات التي ظهرت منذ فترة وأصبحت من ضمن البوابات الرئيسية لأجهزة الحاسوب ، فمن حسناتها أنه يمكن توصيل عدد كبير من ملحقات الحاسوب من خلال منفذ واحد دون الحاجة إلى تعريف سابق لهذه الملحقات ، ولقد أصبحت أغلب منتجات الشركات المصنعة لملحقات الحاسوب في الوقت الحالي تدعم المنفذ (USB) ، ومن تلك الملحقات (لوحات المفاتيح ، الفارات ، الماسحات الضوئية ، وحدات التخزين الخارجية) .

- **منفذ المسرى التسلسلي العام USB 2.0**

تتشابه وظيفة المنفذ (USB2.0) مع المنفذ (USB1.1) ، ويسمح هذا المنفذ بتوصيل ملحقات الحاسوب الشخصي والمحمول باستخدام نفس واجهة التركيب والتشغيل . والفارق الأساسي بين المنفذين هو أن المنفذ (USB2.0) يستطيع نقل المعطيات أسرع بحوالي 50 مرة تقريباً من المنفذ (USB1.1) وهناك مسرى أسرع هو المسرى USB3.0.

- **منفذ الـ SCSI (واجهة ترابط نظام الحاسوب الصغير)**

Small Computer System Interface

يتواجد هذا المنفذ بعدد قليل ، ويستخدم لتوصيل الأقراص الصلبة الخارجية والماسحات الضوئية ، وهو منفذ شائع الاستخدام في نظام ماكنتوش و Intel و Windows ويستخدم في العديد من محطات العمل ، ويعد الواجهة القياسية لمشغلات الأقراص CD-ROM والمعدات الصوتية وتجهيزات التخزين الواسعة ، ويعتمد واجهة الترابط المتوازية وعدد خطوط المعطيات 8 أو 16 أو 32 بت ، ويملك المسرى SCSI معدل نقل للمعطيات يصل إلى 160MB/sec .

• منفذ معطيات الأشعة الحمراء IrDA

Infrared Data Association

يستخدم هذا المنفذ من أجل نقل المعطيات من خلال الموجات الضوئية تحت الحمراء ، ويستخدم بكثرة في الهواتف المحمولة ، وأجهزة الحواسيب المحمولة ، وأجهزة الحواسيب الكفية .

• المنفذ الناري Fire wire

وهو أحد المنافذ الحديثة السريعة جداً ، وهو منفذ تسلسلي عالي الأداء ويصل معدل نقل المعطيات الأعظمي إلى 400Bbit/sec ، ويستخدم لتوصيل بعض الأجهزة ، مثل : الكاميرا الرقمية ، والماسح الضوئي . ويتطلب هذا المنفذ تركيب موصلات النقل Fire wire على الحاسوب ، أو يتم وضعها عبر تركيب موثاق في إحدى فتحات التوسيع على لوحة النظام ، وهو مسرى عالي السرعة وسهل الانجاز ومنخفض الكلفة .

• منفذ توصيل الشاشة

يستخدم المنفذ AGP لتوصيل الشاشة وهو منفذ سريع جداً ويرتبط مباشرة مع المعالج والذاكرة الرئيسية RAM، مجهز لدعم الإرسال السريع للصور والرسومات والفيديو . ويستخدم لتوصيل شاشة العرض إلى جهاز الحاسوب ، ومع ظهور المسرى PCI Express x16 تم الحصول على نظام جديد لبطاقة الفيديو ، وتبديل المسرى AGP بالمسرى PCI Express x16 الذي يوفر إمكانية نقل للمعطيات بسرعة 8GB/sec .

١-٥- الوحدات المحيطية لنظام الحاسوب

Computer Peripheral

وهي عبارة عن أجهزة يتم ربطها بجهاز الحاسوب عبر بطاقات التوسعة ويوابات الحاسوب ، وتُعد صلة وصل بين مستخدمي الجهاز ولوحة النظام ، حيث يقوم المستخدم بالتعامل مع لوحة النظام من خلال الوحدات المحيطية ، وتشمل الوحدات المحيطية لنظام الحاسوب ثلاث وحدات هي :

- ✓ وحدات الإدخال .
- ✓ وحدات الإخراج .
- ✓ وحدات التخزين الثانوية .

١-٥-١- وحدات الإدخال Input Units

وهي الوحدات التي يتم عن طريقها إدخال المعطيات إلى جهاز الحاسوب ، وتستقبل وحدة الإدخال المعلومات المرمزة (معطيات Data أو التعليمات Instruction) من الأجهزة الخارجية للإدخال وترسلها إلى وحدة الذاكرة الرئيسية ليتم حفظها ولتعالج فيما بعد ، أو ترسل إلى وحدة الحساب والمنطق ALU لتعالج مباشرة ، ونتيجة المعالجة ترسل إلى وحدة الذاكرة لحفظها ، أو ترسل إلى وحدة الإخراج لإظهارها .

وهي وسيلة اتصال بين المستخدم والحاسوب ، حيث تقوم هذه الوحدات باستقبال المعطيات والتعليمات وإدخالها إلى الحاسوب، حيث أنها تقوم بعملية توصيل المعلومات من لغة الإنسان المكونة من الأرقام والحروف إلى لغة الآلة المكونة من نبضات كهربائية (وجود نبضة كهربائية تمثل بـ 1 منطقي وعدم وجود نبضة كهربائية تمثل بـ 0 منطقي)، ومن أهم وحدات الإدخال يمكن أن نذكر :

- لوحة المفاتيح (Keyboard) .
- الفأرة (Mouse) .
- الماسح الضوئي (Scanner) .
- الكاميرا الرقمية (Digital Camera) .
- جهاز القلم الضوئي (Light Pen) .
- شاشة اللمس (Touch Screen) .
- السبورة الإلكترونية (Electronic Board) .
- عصا التحكم بالألعاب (Joy stick) .
- جهاز اللاقط (الميكروفون) (Microphone) .
- جهاز قارئ الأعمدة (Bar Code Reader) .
- بطاقات الاتصال (Telecommunication Cards) .
- الفاكس Fax .

وفي هذا السياق سنتعرف على بعض الوحدات الأكثر حاجة وأكثر استخداماً:

١- لوحة المفاتيح Keyboard

تُعد لوحة المفاتيح من أهم مكونات الحاسوب وأكثر أجهزة الدخل استخداماً، ويتم بواسطتها إدخال المعطيات بشكل أرقام وحروف ورموز (المعطيات النصية) ، ولا يمكن الاستغناء عنها في أي حاسوب ، وتحتوي جميع لوحات المفاتيح على اختلاف أنواعها وأشكالها على دائرة إلكترونية ، على شكل مصفوفة، لها أسلاك بشكل أسطر وأخرى بشكل أعمدة غير متصلة بعضها ببعض عند نقاط التقاطع ، وفوق كل نقطة تقاطع يوجد مفتاح من مفاتيح اللوحة، وعند الضغط على أحد المفاتيح يؤدي ذلك إلى توليد إشارة كهربائية تُنقل إلى الحاسوب وتوافق رمز الحرف الذي ضُغَطَ مفتاحه ، ويبين الشكل (١-١) لوحة المفاتيح ، وتحتوي

لوحة المفاتيح على مجموعات من المفاتيح الاختيارية الخاصة تسمى مفاتيح الوظائف هي:

- ✓ مفاتيح الأسهم Arrow Keys : التي تمكن المستخدم من تحريك مؤشر الشاشة نحو الأعلى والأسفل أو نحو اليمين واليسار ، ونستفيد من هذه المفاتيح عند استخدام وكتابة البرامج.
- ✓ مفاتيح الأحرف والرموز Typing Keys : وتشمل مفاتيح الحروف الإنكليزية (A - Z) والحروف العربية (أ - ي)، وكما تشمل الرموز الخاصة المختلفة مثل (+ ، - ، # ، \$ ، % ، & ، ، ...).
- ✓ مفاتيح الأرقام Number Keys : وتشمل الأرقام من 0 إلى 9.
- ✓ مفاتيح الوظائف Function Keys : وتشمل المفاتيح (F1—F12) وهي مفاتيح قابلة للبرمجة من قبل المبرمج للقيام بوظيفة محددة، لذلك تختلف وظيفتها باختلاف البرنامج المستخدم وقد يستخدم بعضها فقط.
- ✓ مفاتيح الحاسوب الخاصة Special Keys : وهي مفاتيح خاصة تستخدم استخداماً منفرداً أو مشتركاً مع مفاتيح أخرى لأداء وظائف مختلفة. مثل مفاتيح Esc , Home , Delete , Tab , Shift..الخ.



الشكل (١١-١) - يوضح لوحة المفاتيح

٢ - الفأرة Mouse

تُعد لوحة المفاتيح وسيلة فعالة لإدخال الحروف والأرقام والرموز ، ولكنها غير سهلة الاستخدام عند الرغبة في إدخال الرسوم أو إعطاء الأوامر للحاسوب ، أو التحرك من موقع لآخر على شاشة الجهاز ، لذا يتم استخدام جهاز الفأرة والتي هي إحدى وحدات الإدخال الأكثر استخداماً ، ويمكن من خلالها التحكم في المؤشر الذي يظهر في نظام التشغيل لإعطاء الأوامر والتعليمات للحاسوب ، وببين الشكل (١٢-١) الفأرة .



الشكل (١٢-١) - يوضح الفأرة

- الفأرة (الماوس) هي وحدة إدخال توفر للمستخدم سهولة كبيرة في التعامل مع الحاسوب ، ويوجد العادية منها والليزرية.
- تستخدم الفأرة كوسيلة للتحكم في موضع مؤشر صغير مضيء على الشاشة.
- بتحريك الفأرة على الطاولة يتحرك المؤشر بحركة تتوافق مع حركة الفأرة.
- تتضمن الفأرة على زرّين أو ثلاثة بحسب الطراز ، ويستخدم الزر الأيسر في الغالب لسحب القوائم واختيار العناصر وتشغيل البرامج وفتح الملفات ، بينما الزر الأيمن يستخدم لاستعراض القوائم المختصرة (قوائم التعليمات والأوامر) ، ومن أجل اختيار أي أمر يتم الضغط على الزر الأيسر .

• تستخدم الفأرة في عملية السحب والإفلات .

وتختلف أجهزة الفأرة بحسب الأمور التالية :

✓ منفذ التوصيل : حيث أن بعضها يتم توصيله بالمنفذ التسلسلي ، وبعضها

الأخر يتم توصيله على المنفذ USB .

✓ تقنية التوصيل : حيث أن بعضها يتم توصيله بواسطة كبل (سلكي) ،

وبعضها الآخر يتم توصيله عن بعد (لا سلكي) .

✓ نوع المحرك : حيث أن بعضها يكون بواسطة كرة متحركة (Wheel)

، وبعضها الآخر يكون بشكل ضوئي (Optical) .

٣- الماسح الضوئي Scanner

الماسح الضوئي وحدة إدخال يُستخدم لإدخال الصور والوثائق لمعالجتها أو أرشفتها عن طريق عدسة ضوئية مثبتة على رأس الماسح، مهمتها تحويل المساحات المطبوعة إلى معلومات رقمية يمكن تخزينها ضمن الحاسوب . ويوجد من الماسحات ما يستطيع تمييز اللون الأسود والأبيض فقط ومنها ما يستطيع تمييز الألوان وتدرجاتها، بحيث تعطي الصورة دقة عالية.

والماسح الضوئي جهاز يشبه آلة التصوير ، يستخدم حزمًا ضوئية يتم مرورها على المستند أو الصورة ، ثم ينقل نسخة من هذا المستند أو الصورة إلى داخل الحاسوب ، ويمكن من خلال الماسح الضوئي إدخال الصور والرسومات التي يطلق عليها بالمعطيات الرسومية ، بينما يطلق على الحروف والأرقام والرموز بالمعطيات النصية ، ويبين الشكل (١-١٣) الماسح الضوئي Flat

bed .



الشكل (١٣-١) - يوضح الماسح الضوئي Flat bed

وتختلف الماسحات الضوئية بحسب الأمور التالية :

• النوع : فمنها :

- الماسحات الضوئية اليدوية (Hand hold) ، والتي تستخدم في تصوير جزء من الصفحة أو قراءة شيفرة البضاعة في المحلات التجارية، كما هو مبين في الشكل (١٤-١) .



الشكل (١٤-١) - الماسح الضوئي Hand hold

- الماسحات المسطحة (Flat bed) كآلة التصوير وتستخدم لتصوير الصفحات كما هو مبين في الشكل (١٣-١) .
- ومنها ما توضع الورقة فيه مثل الفاكس و تدعى Sheet Fed ، كما هو مبين في الشكل (١٥-١) .



الشكل (١٥-١) - ماسح ضوئي Sheet Fed

- كما يستخدم لهذا الغرض ماسحات مثبتة في مكانها تدعى Stationary Scanners وهي بحجم الطاولة تمرر عليها البضاعة. وأخيراً هناك الماسحات الاسطوانية Drum Scanner والتي تلتف فيها الورقة المطلوب تصويرها على اسطوانة كما هو مبين في الشكل (١٦-١) .



الشكل (١٦-١) - الماسح الاسطواني Drum Scanner

عندما يتم تصوير وثيقة من الماسحات الضوئية عادة لا يمكن التعديل عليها داخل الحاسوب لأنها تخزن كصورة ، أما إذا تحولت الوثيقة إلى نص قابل للتعديل فإن النظام المستخدم هنا يدعى OCR (Optical Character Recognition) وهو يحتاج إلى برمجة خاصة . أيضاً النظام المستخدم في قراءة شيفرة البضاعة هو نظام OCR حيث يتم إدخال الشيفرة و التي تكون على شكل خطوط سوداء إلى الحاسوب الذي يقوم بتحليلها و بالتالي تمييز البضاعة المباعة عن غيرها، ومن مميزات:

- **الدقة :** وهي درجة وضوح الصورة أو النص ، وتقاس بعدد النقاط في الأنش (Dots Per Inch) .
- **السرعة :** وتقاس بعدد الصفحات المقرؤة في الدقيقة الواحدة .
- **التوصيل :** يمكن توصيل الماسح الضوئي إلى المنفذ التفرعي ، أو المنفذ الاسكازي SCSI ، أو المنفذ USB .

٤ - الكاميرا الرقمية Digital Camera

تشبه الكاميرا العادية ، وتختلف عنها بطريقة العمل ، ويمكن تخزين الصورة بداخلها أو إرسالها إلى الحاسوب أو الطابعة مباشرة ، وتعد وحدة إدخال وإخراج في نفس الوقت ويبين الشكل (١-١٧) الكاميرا الرقمية . وتختلف أجهزة الكاميرا الرقمية بحسب الأمور التالية :

- **نوع الصورة الملتقطة :** هل هي ثابتة أم متحركة (فيديو) ؟
- **نوع التخزين :** أين يتم تخزين الصور الملتقطة في ذاكرة خارجية أم قرص مرن ؟
- **حجم الذاكرة الداخلية :** كم عدد الصور التي يمكن تخزينها بدون وجود ذاكرة خارجية أو قرص مرن ؟

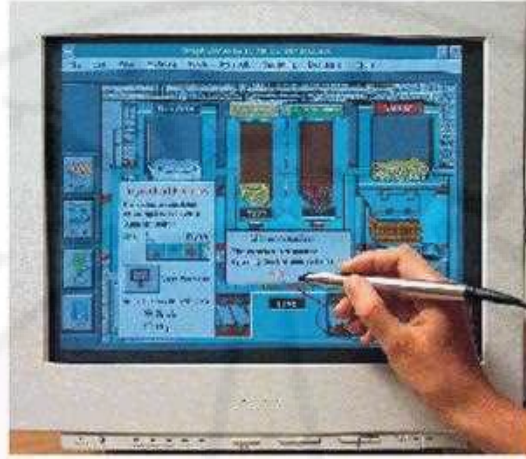
- **دقة التصوير :** أي كثافة (عدد) النقاط الضوئية pixels لكل صورة ، فكلما ازدادت عدد هذه النقاط كلما كانت الصورة أكثر دقة .



الشكل (١٧-١) - يوضح الكاميرا الرقمية

٥-جهاز القلم الضوئي Light Pen

يستخدم هذا الجهاز لإدخال الرسوم والخطوط والأشكال إلى جهاز الحاسوب، ويختلف عن الفأرة ولوحة المفاتيح من حيث أن المعطيات المدخلة لا تعتمد على حركة اليد بل تعتمد على موقع الحزمة الضوئية الصادرة عن القلم ، حيث يتم توجيه القلم الضوئي إلى شاشة الجهاز ويقوم الحاسوب بالتعرف على موقع الضوء على الشاشة وإدخال المعطيات المطلوبة في ذلك الموقع ، وبالتالي يمكن رسم الخطوط والأشكال المختلفة على شاشة الجهاز بتحريك القلم الضوئي على الشاشة حسب الشكل أو الخطوط المطلوب رسمها وإدخالها ، ويبين الشكل (١٨-١) توجيه القلم الضوئي إلى الشاشة .



الشكل (١٨-١) - يبين توجيه القلم الضوئي إلى الشاشة

الشكل الآخر للقلم الضوئي عبارة عن قضيب يكتب على لوح خاص يسمى Digitizer يوضع على سطح المكتب ثم تخزين ما تم كتابته أو رسمه داخل الحاسوب، كما هو مبين في الشكل (١٩-١) .



الشكل (١٩-١) - شكل آخر للقلم الضوئي

٦- شاشة اللمس Touch Screen

تسمح بعض أجهزة الحاسوب بأن تكون شاشة العرض للجهاز حساسة للمس من قبل المستخدم ، وبالتالي يقوم المستخدم للجهاز بإدخال المعطيات لإعطاء الأوامر أو اختيار موقع داخل الشاشة عن طريق اللمس للمواقع الحساسة على الشاشة ، وأحياناً يتم تزويد الجهاز بشاشة لمس منفصلة عن شاشة العرض وتقوم بنفس المهمة عن طريق لمس المستخدم لها ، ومن الأمثلة الشائعة لشاشات اللمس الشاشات المستخدمة في معظم أجهزة الصرف الآلي ، كما هو مبين في الشكل (٢٠-١) .



الشكل (٢٠-١) - يبين شاشة اللمس

وكما يستخدم لوح مسطح حساس للمس صغير الحجم كبديل للفأرة ، عندما يتحرك الأصبع على سطحه يتغير موقع مؤشر الشاشة تبعاً لذلك كما هو مستخدم في الحواسيب المحمولة ، ويبين الشكل (٢١-١) هذا النوع .



الشكل (٢١-١) - الإدخال بلمس لوح خاص Touch Pad

٧- السبورة الإلكترونية Electronic Board

من أحدث المنتجات العلمية والتعليمية التي تعمل مع الحاسوب ، حيث يتم نقل كل ما يكتبه المعلم أو المدرب على السبورة إلى جهاز الحاسوب باستخدام قلم خاص .

٨- عصا التحكم بالألعاب Joy stick

وتستخدم للتعامل مع برمجيات الألعاب ، حيث يتم نقل حركة العصا إلى داخل الجهاز لإعطاء الأمر بالتحرك على شاشة العرض من موقع إلى آخر ، وبالضغط على المفاتيح الموجودة على عصا التحكم يتم إدخال الأوامر المطلوبة . وهي عبارة عن عصا تمسك باليد مثبتة بقاعدة متصلة بالحاسوب و تحرك في جميع الاتجاهات للتحكم بالحركة على الشاشة ، قد تزود بمجموعة أزرار لتأدية بعض المهام ، ويبين الشكل (٢٢-١) عصا الألعاب .



الشكل (٢٢-١) - يبين عصا الألعاب

٩-جهاز الميكروفون Microphone

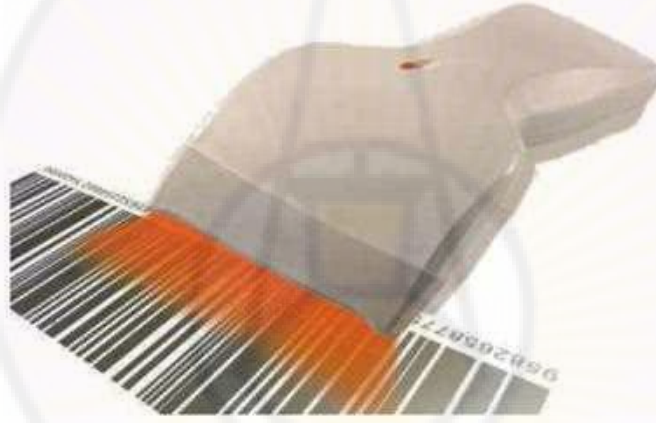
وهو جهاز يستقبل الموجة الصوتية الصادرة عن المتكلم ويحولها إلى
أشارات رقمية يتم إدخالها وتخزينها داخل جهاز الحاسوب (أي إدخال المعطيات
الصوتية إلى الحاسوب)، ويُستخدم في تسجيل الأصوات في الحاسوب كما
يستخدم في نقل التعليمات من المستخدم إلى الحاسوب وبالتالي القيام بمهمة أو
نقل حديث المستخدم إلى معالج النصوص فيتحول الحديث إلى نص مكتوب بدلا
من إدخاله عبر لوحة المفاتيح ، ويبين الشكل (٢٣-١) جهاز الميكروفون .



الشكل (٢٣-١) - يبين جهاز الميكروفون

١٠-جهاز قارئ الأعمدة Bar Code Reader

يستخدم هذا الجهاز لإدخال المعطيات الموجودة على المنتجات التجارية ، حيث يتم وضع شكلاً على المنتج يتكون من مجموعة من الأعمدة السوداء المتفاوتة في العرض والمسافة بينها ، ويمثل تعريفاً بالمنتج والشركة الصانعة له ، ويقوم جهاز قارئ الأعمدة بإمرار حزمة ضوئية على هذه المجموعة ، وبالتالي يتم نقل العبارة التي يمثلها شكل مجموعة الأعمدة إلى داخل الجهاز ، ويبين الشكل (١-٢٤) جهاز قارئ الأعمدة .



الشكل (١-٢٤) - يبين جهاز قارئ الأعمدة

١-٥-٢- وحدات الإخراج Output Units

وظيفتها إخراج المعطيات على شكل معلومات من الحاسوب بعد معالجتها ، ويتم في هذه الوحدة إظهار نتائج المعالجة من وحدة الحساب والمنطق أو إخراج وإظهار المعلومات من وحدة الذاكرة عبر أجهزة الإخراج ، وأهمها الشاشة ، التي توضح للمستخدم ماذا تفعل وحدة المعالجة وتظهر نتائج المعالجة ولذلك تسمى بالمراقب Monitor ، ومن أهم وحدات الإخراج يمكن أن نذكر :

- الشاشة (Screen) .
 - الطابعة (Printer) .
 - الراسمة (Plotter) .
 - السماعات الصوتية (Speakers) .
- وفي هذا السياق سنتعرف على بعض الوحدات الأكثر حاجة واستخداماً:

١-الشاشة Screen

- تشبه الشاشة المستخدمة لعرض معطيات الحاسوب وشاشة العرض التلفزيوني ، حيث تقوم بعرض المعطيات النصية (الحروف والأرقام والرموز) ، وكذلك المعطيات الرسومية (الصور والأشكال والرسومات) ، وكذلك المعطيات الفيديوية (صور الفيديو المتحركة) ، وتعد من أهم أجهزة الإظهار، وهي الوسيط المستخدم لإظهار نتائج معالجة المعطيات ، وتسمى وحدة العرض المرئي Video Display unit أي (VDU) ، ولكي نستطيع إظهار العرض المرئي نحتاج إلى كرت شاشة ، وتختلف الشاشات فيما بينها تبعاً للأمور التالية :
- نوع التقنية المستخدمة للعرض : يوجد تقنية أنبوب الكاثود CRT ، أو تقنية الشاشة المسطحة flat ، أو تقنية الشاشة البلورية LCD .
 - مساحة شاشة العرض : ويقاس قطرها بالبوصة (انش) ، حيث توجد شاشات قطرها 14 بوصة ، 15 بوصة ، 17 بوصة ، 21 بوصة أو أكثر .
 - دقة العرض : وتقاس بعدد النقاط الضوئية pixel ، فكلما زاد عدد النقاط الضوئية كلما كانت الصورة أكثر دقة ووضوحاً .

وهناك عدة أنواع أهمها :

١- شاشات الـ CRT (Cathode Ray Tube)

وهي شاشة أنبوبة الأشعة المهبطية تشبه شاشة التلفاز إلا أنها أكثر وضوحاً ، وتأخذ حيزاً كبيراً كما أنها ثقيلة الوزن، ويبين الشكل (٢٥-١) شاشة CRT .



الشكل (٢٥-١)- يبين شاشة CRT

٢- شاشات العرض المسطحة Flat Panel Display

هي شاشات مستوية تُستخدم في الحواسيب المحمولة laptop ، تبلغ سماكتها حوالي 0.5 إنش ، وأكثرها شيوعاً شاشة السائل البلوري Liquid Crystal Display (LCD) وهذا النوع من الشاشات يأخذ حيزاً صغيراً ويتميز بخفة الوزن ، ويبين الشكل (٢٦-١) شاشة LCD .



الشكل (٢٦-١)- يبين شاشة LCD

٢-الطابعات Printers

الطابعة هي وحدة إخراج متميزة لإمكانية الحصول منها على سجل دائم ومطبوع (نسخة ورقية) للنتائج والمعطيات ، التي تم الحصول عليها بعد عملية المعالجة بالحاسوب ، حيث يمكن الحصول على معطيات نصية (حروف وأرقام و رموز) ومعطيات رسومية (صور وأشكال و رسومات) مطبوعة على أوراق ، وتزود الطابعات بمعالج ميكروبي Microprocessor يتحكم في جميع الوظائف ، كما تزود بذاكرة مؤقتة Buffer لتخزين أوامر وإشارات الطباعة التي تتلقاها من الحاسوب ، وتتميز أجهزة الطابعات فيما بينها بالأمور التالية :

- لون الطباعة : توجد طابعات ملونة (التي تطبع بالألوان) ، أو غير ملونة (التي تطبع باللون الأسود فقط) .
- سرعة الطباعة : وتقاس سرعة الطابعات النقطية بعدد الحروف والكلمات المطبوعة في الدقيقة ، بينما تقاس سرعة الطابعات الحبرية والليزرية بعدد الصفحات بالدقيقة ، وتعتبر الطابعات الليزرية الأسرع في الطباعة .
- دقة الطباعة : وتقاس الجودة في الطابعات بعدد النقط التي يمكن طباعتها في الإنش Inch (Dot Per Inch) DPI ، فكلما ازداد عدد النقاط ازدادت جودة الطباعة ، وتعتبر الطباعة الليزرية أكثر الأنواع دقة ، تليها الطباعة الحبرية (النافثة للحبر) ، ثم الطباعة النقطية .
- متعددة المهام : تقوم الطباعة بعدد من المهام مثل : طباعة ، ماسح ضوئي ، نسخ الورق ، فاكس .
- نوع التقنية المستخدم للطباعة : وفيما يلي سوف نذكر بعض هذه التقنيات وبعض أنواع الطابعات وبعض مواصفات كل منها :

أ - طابعة العجلة Daisy Wheel

تستخدم هذه الطابعات عجلة محفوراً عليها الأحرف و الرموز ، وعندما تبدأ عملية الطباعة تدور العجلة حتى الوصول إلى الرمز المطلوب وتضرب مطرقة العجلة من الخلف فيصبح شكل الحرف على الورقة بواسطة شريط التحبير . وهذه الطابعة مطرقية وتطبع نسخاً كربونية وهي بطيئة ومزعجة، ويبين الشكل (٢٧-١) طابعة العجلة .



الشكل (٢٧-١) - يبين طابعة العجلة

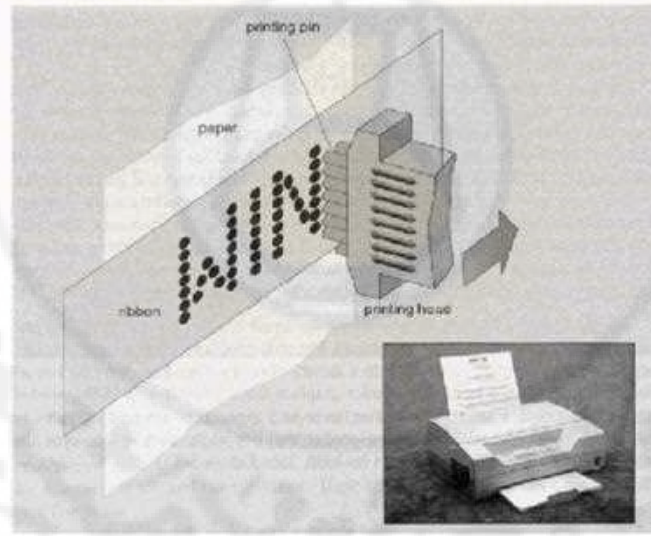
ب - الطابعة النقطية Dot Matrix Printer

تشبه الآلة الكاتبة في طريقة عملها والفرق هو أن الحرف يطبع بكامله في الآلة الكاتبة ، بينما في الطابعة النقطية يتم تكوين أي حرف من مجموعة من النقاط يتم صفها بحيث تأخذ شكل الحرف المطلوب ، وتتم عملية الطباعة كما يلي:

الرمز عبارة عن مجموعة من النقاط مرتبة بشكل معين في مصفوفة ، ويتكون رأس الطابعة Print Head من مجموعة دبابيس يمتد منها دبابيس معينة أثناء حركة الطباعة على عرض الورقة من اليمين إلى اليسار حيث تضرب على شريط التحبير مكونة الرمز على شكل نقاط، ويتحكم حجم و عدد الدبابيس المستخدمة في رأس الطباعة بنوعية المخرجات وتستخدم في طباعة المسودات.

وبين الشكل (٢٨-١) الطابعة النقطية ، ومن أهم مميزاتة :

- تستخدم في أعمال الطباعة على الوثائق المتعددة الطبقات.
- رخيصة الثمن.
- جودة ودقة الطباعة فيها ليست عالية.
- تشبه الآلة الكاتبة في طريقة عملها.
- تقاس سرعتها بعدد المحارف التي يمكن طباعتها في الثانية وكلما كان العدد كبيراً كانت الطابعة سريعة ، وأسرع الطابعات يصل إلى 400 حرف/ الثانية .



الشكل (٢٨-١) - بين الطابعة النقطية

ج- الطابعة النافثة للحبر Ink Jet Printer

وتحتل موقعاً وسطاً بين الطابعات النقطية والليزرية ، وهي نوعان :
الأول يعمل بنظام النبضات الكهربائية والثاني بنظام تسخين الحبر ، ومع النوعين
يستخدم حبر سائل مذاب في الماء ويوجد فيها خرطوشة لها فتحات صغيرة جداً ،
وسواء كانت كهربائية أو حرارية ، يتم نفث الحبر من هذه الفتحات على الورقة
بينما يتحرك رأس الطابعة عبر الصفحة حتى تكتمل الطابعة ، قد تستخدم هذه
الطابعات عبوة ملونة واحدة أو عدة عبوات كل منها بلون مختلف بالرغم من سرعة
هذه الطابعات وهدوئها وجودة طباعتها العالية إلا أن تكلفة تشغيلها عالية نسبة إلى
الطابعات النقطية وعجلة ديزي، وهي بطيئة نسبياً وأحبارها غالية ، إلا أنها
رخيصة الثمن ، وذات جودة طباعة عالية 1400 DPI ، ويبين الشكل (٢٩-١)
الطابعة النافثة للحبر .



لشكل (٢٩-١) يبين الطابعة النافثة للحبر

د- الطابعة الليزرية Laser Printer

تعد الطابعات الليزرية من أفضل أنواع الطابعات في جودة طباعة النصوص والرسوم ، وأغلاها سعراً ، ويستخدم فيها نوع من الحبر الجاف في علبه خاصة Toner يشبه المستخدم في آلة تصوير المستندات ، وهو حبر غالي الثمن ، وأقل طابعات الليزر تعطي جودة طباعة 300DPI ، وتصل اليوم إلى أكثر من 600DPI ، تقاس سرعة الطباعة بعدد الصفحات التي تُطبع في الدقيقة، ويبين الشكل (٣٠-١) الطابعة الليزرية .



الشكل (٣٠-١) - يبين الطابعة الليزرية

٣-الراسمة Plotter

الراسمة وحدة إخراج وتستخدم لإخراج المخططات الهندسية التي تم رسمها على الحاسوب، و تملك مجموعة أقلام يقوم الحاسوب بالتحكم فيها وتحريكها لرسم المخطط المطلوب، كما يمكن تزويدها بعدة أقلام ملونة حسب الحاجة، وتمتاز بدقة إخراجها للرسومات بالمقارنة مع الطابعة ، وتستخدم في الهندسة المعمارية ودراسة الزلازل الأرضية ، و في أنظمة التصميم باستخدام برنامج

الحاسوب CAD ، ويوجد منها أنواع تعمل على مبدأ نفث الحبر بدلاً من استخدام الأقلام ، وتستخدم بكثرة من قبل المهندسين والمعماريين لإخراج الرسوم والمخططات المعمارية والميكانيكية والصناعية، ويبين الشكل (٣١-١) الراسمة .



الشكل (٣١-١) - يبين الراسمة

٤- السماعات الصوتية Speakers

بواسطة هذه الأجهزة يتم تحويل الإشارات الرقمية إلى موجات صوتية ، حيث تتيح هذه الأجهزة لمستخدم الحاسوب سماع الأصوات الصادرة عن التطبيقات داخل الحاسوب مثل برمجيات الألعاب ، أو البرمجيات الصوتية ، أو البرمجيات الخاصة بالفيديو ، ويبين الشكل (٣٢-١) السماعات .

وهي وحدة الإخراج الصوتية وهي جزء من النظام الصوتي Sound System الذي يتكون من كرت الصوت Sound Card والميكروفون والسماعات .

ويمكن أن نضيفه للحاسوب كجزء من الوسائط المتعددة Multimedia والتي تشمل أيضاً على الصور والفيديو والحركة Animation .



الشكل (١-٣٢) يبين السماعات

١-٦- وحدات التخزين الثانوية

Secondary Storage Units

يقوم جهاز الحاسوب بتخزين المعطيات في مكانين داخل الحاسوب ، وذلك بناءً على مدى الحاجة لها من قبل وحدة المعالجة المركزية ، فإذا استخدمت المعطيات بشكل فوري من قبل وحدة المعالجة المركزية ، يتم الاحتفاظ بها في وحدة الذاكرة الرئيسية الموجودة على لوحة النظام ، وإذا كانت المعطيات لا تتطلبها وحدة المعالجة المركزية بشكل فوري فإنه يتم تخزينها في وحدات التخزين الثانوية بالجهاز والتي تكون داخل صندوق وحدة النظام أو خارجه وتتصل به عن طريق بوابة مناسبة لذلك .

وهي ذاكرات ذات سعات عالية إضافية للحاسوب ويمكن حفظ المعطيات والبرامج بصورة دائمة عليها لتستخدم عند الحاجة ، وهذا النوع يقع خارج وحدة المعالجة المركزية ويطلق على هذه الوحدات اسم وحدات التخزين

الخارجية External Storage Units ، وتتميز الذاكرة عن وسائل وحدات التخزين الثانوية الملحقة بالجهاز بالأمور التالية :

◀ إن الذاكرة هي دارة الكترونية بعكس وحدات التخزين الثانوية الملحقة بالجهاز ، والتي تتكون من أجهزة ميكانيكية المعتمدة على تقنية الكهرومغناطيسية أو على التقنية الضوئية .

◀ إن الذاكرة تتصل مباشرة بالمعالج على لوحة النظام ما يسمح بتمرير المعطيات ما بين المعالج والذاكرة بسرعة عالية (سرعة مرور النبضات الكهربائية) ، بينما تتصل وحدات التخزين الثانوية الملحقة بالجهاز بالمعالج عن طريق مسار خاص يسمى المسرى وعبر بوابات الجهاز ، ما يحتاج إلى زمن أطول بكثير لإمرار المعطيات ، ويبين الشكل (١ - ٣٣) بعض وحدات التخزين الثانوية



الشكل (١ - ٢٦) - يبين بعض وحدات التخزين الثانوية

وهذه الفروق بين الذاكرة ووحدات التخزين الثانوية ، لذلك يطلق أحياناً على الذاكرة الرئيسية تسمية وحدة التخزين الرئيسية ، بينما يطلق على أجهزة التخزين تسمية وحدات التخزين الثانوية . وتتميز وحدات التخزين الثانوية بما يلي :

١. ذات تكلفة صغيرة (رخيصة الثمن) .
٢. ذات سعة أكبر بكثير من الذاكرة الرئيسية .
٣. ذات سرعة وصول للمعلومات أصغر بكثير من الذاكرة الرئيسية .
٤. يوجد منها عدة أنواع، لكل منها مشغل خاص .

ويمكن قياس القدرة التخزينية لوحدات التخزين وكذلك لحجم المعطيات بداخلها بواسطة وحدات القياس التالية :

1. **Byte** = 8 bit
2. **Kbyte** = 1024 Byte
3. **Mbyte** = 1024 K byte = $1024 * 1024$ Byte
4. **Gbyte** = 1024 M byte = $1024 * 1024 * 1024$ Byte
5. **Tbyte** = 1024 Gbyte = $1024 * 1024 * 1024 * 1024$ Byte
6. **Bbyte** = 1024 Tbyte = $1024 * 1024 * 1024 * 1024 * 1024$ Byte
7. **Ebyte** = 1024 Bbyte = $1024 * 1024 * 1024 * 1024 * 1024 * 1024$ Byte

ومن أهم وحدات التخزين الثانوية :

١ - الأقراص المرنة Floppy Disks

الأقراص المرنة هي أقراص بلاستيكية مطيية بمادة قابلة للتمغنط ، والأقراص الشائعة للاستخدام هي بقياس 3.5 إنش (بوصة) وتبلغ سعتها التخزينية 1.44 ميغابايت ، ويبين الشكل (١-٣٤) القرص المرن ، وتمتاز هذه الأقراص بما يلي:

- ١- ذات حجم 1.44 MB .
- ٢- تعد ذاكرة قراءة وكتابة .
- ٣- تعد ذاكرة دائمة .
- ٤- قابلة للتلف بشكل سريع.



الشكل (١-٣٤) - يبين القرص المرن

٢- الأقراص الصلبة Hard Disks

هي أقراص ممغنطة من النوع الصلب وتكون الأقراص متوضعة بعضها فوق بعض ، ويبين الشكل (١-٣٥) القرص الصلب ، وتمتاز هذه الأقراص بما يلي:

- ١- ذاكرة ذات حجم كبير جداً بسعات تصل حالياً إلى مئات أو آلاف الجيجابايت وهي في تزايد مستمر .
- ٢- تعد ذاكرة قراءة وكتابة .
- ٣- تعد ذاكرة دائمة .



الشكل (١-٣٥) - يبين القرص الصلب

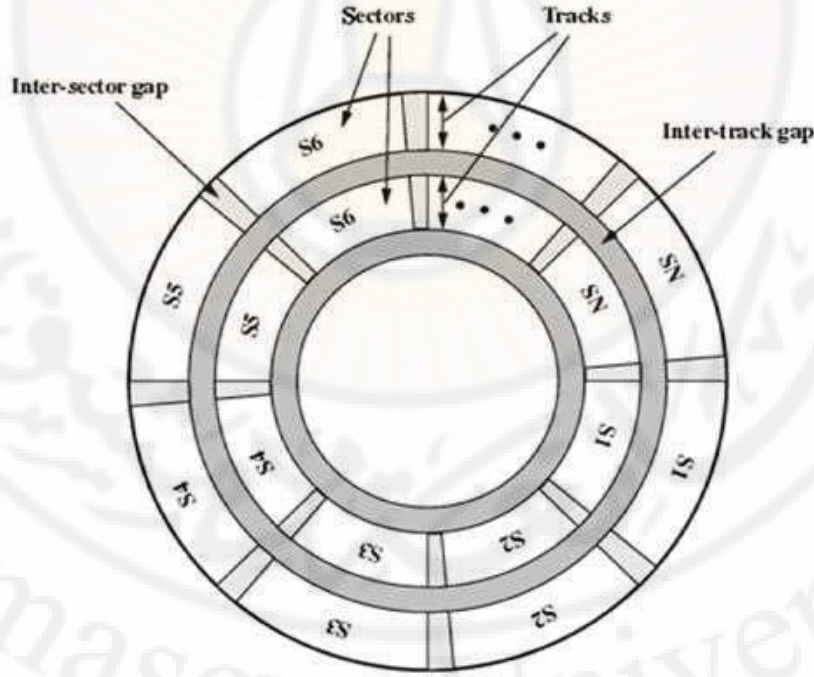
وتختلف الأقراص الصلبة بحسب الأمور التالية :

موضعها : داخل صندوق الحاسوب (ثابتة) ، أو خارج صندوق الحاسوب (متحركة) عن طريق المنفذ (USB) .

سعة التخزين : مئات أو آلاف الجيجابايت وفي تزايد مستمر .

تنظيم المعطيات وصياغتها في القرص الصلب:

إن الرأس هو جهاز صغير نسبياً قادر على قراءة جزء من القرص الدائر أو الكتابة فيه ، وهذا ما يؤدي إلى تنظيم المعطيات على القرص كمجموعة حلقات تسمى المسالك Tracks. يساوي ثخن كل مسلك ثخن الرأس ، ويتراوح عدد المسالك بين 500 و 2000 مسلك على السطح الواحد . وتفصل المسالك المتجاورة بفجوات gaps وذلك لمنع حدوث الأخطاء الناتجة عن عدم محاذاة الرأس أو عن تداخل الحقول المغناطيسية . ويحتوي المسلك الواحد على عدد يتراوح بين (10→100) قطاع وتفصل القطاعات المتجاورة بفجوات gaps ويُخزن في كل قطاع 512 بايت من المعطيات ، ويوضح الشكل (١-٣٦) مخطط المعطيات النسقي للقرص الصلب .



الشكل (١-٣٦) - يبين مخطط المعطيات النسقي للقرص الصلب

٣- الأقراص الليزرية CD- ROM (Compact Disks)

هي أقراص ذات سعات عالية حيث يتم التسجيل عليها بواسطة أشعة ليزرية ، ويقوم محرك الأقراص الليزرية بقراءة أقراص مستديرة الشكل أو شبه مستديرة مكونة من مادة عاكسة للضوء ، وتمتاز هذه الأقراص بما يلي :

- ذات سعة كبيرة أكثر من 700MB.
 - تعد ذاكرة قراءة وكتابة.
 - تعد ذاكرة دائمة .
 - من أجل الكتابة على هذه الأقراص نحتاج إلى مشغل الأقراص CD- ROM (Read \Write).
- وهناك نوعان من مشغلات الأقراص الليزرية :

النوع الأول : يقرأ الأقراص ولا يسمح بالكتابة عليها وهذا النوع يسمى قارئ الأقراص الضوئية (CD-ROM).

النوع الثاني : يقرأ الأقراص و يسمح بالكتابة عليها وهذا النوع يسمى بكاكتب الأقراص الضوئية (CD-Writer).

أما بالنسبة للأقراص المستخدمة للكتابة عليها فهي على نوعين :

أقراص تسمح بالكتابة مرة واحدة وتسمى CD-R ، وأقراص تسمح بالكتابة عليها عدة مرات وتسمى CD-RW ، ويختلف القرص المرن والصلب عن القرص الليزري من حيث إن القرص الليزري CD-R لا يمكن إعادة التخزين عليه أو نسخ المعطيات المخزنة عليه مرة أخرى ، وذلك لأن ثقوب أشعة الليزر تكون دائمة بخلاف القرص الممغنط الذي يمكن إعادة التخزين عليه مرة أخرى ، وتمتاز الأقراص الليزرية بما يلي :

✓ ذات سعة تخزينية تصل 650 MB للقرص الواحد

- ✓ تعد الأقراص CD-ROM ذاكرة قراءة فقط بينما أقراص CD-RW ذاكرة قراءة وكتابة .
- ✓ تعد ذاكرة دائمة .
- ✓ للكتابة بحاجة إلى CD - ROM Read / Write .
- ✓ يستخدم أشعة الليزر لقراءة المعلومات .
- ✓ يستخدم لتخزين البرامج متعددة الوسائط (الصوت و الصورة و الفيديو) .
- ✓ خفيف الوزن ذو وثوقية عالية.

٤ - أقراص الفيديو الرقمية DVD

Digital Video (Versatile) Disc

وهذه الأقراص تشبه الأقراص الليزرية ، وتستخدم تقنية ضوئية (الليزر) للتخزين ، إلا أن السعة التخزينية لهذا النوع من الأقراص كبيرة، ويبين الشكل (١-٣٧) القرص DVD .



الشكل (١-٣٧) - يبين القرص DVD

ويستطيع القرص DVD تخزين حوالي 4.7 GB في كل طبقة ويمكن لقرص وحيد الجانب ومزدوج الطبقات تخزين 8.5 GB ، و يستخدم القرص DVD شكلاً من أشكال ضغط الفيديو ، المسمى MPEG ، بغية تخزين الصور العالية الجودة ، وتستخدم هذه الأقراص لتسجيل شرائط الفيديو والألعاب الحاسوبية ،

٥- الشريط الممغنط Tape Driver

يستخدم هذا النوع من الأشرطة لعمل النسخ الاحتياطية للمعطيات ، وأغلب مشغلات الأشرطة الممغنطة تكون خارجية وتوصل بالحاسوب ، وهناك أيضاً أنواع داخلية ، وتختلف الأشرطة الممغنطة من حيث سعة التخزين والشكل الخارجي ، ويبين الشكل (١-٣٨) الشريط المغناطيسي .



الشكل (١-٣٨) - يبين الشريط المغناطيسي

٦- ذاكرة الفلاش ديسك Flash Disk

يعد الفلاش ديسك من أحدث وسائل التخزين الثابتة ، وذات حجم صغير وسعات تخزين كبيرة تصل إلى عشرات الجيجابايت وهي في تزايد مستمر ، وهو

يعمل كقرص تخزين قابل للإزالة ، ويتم توصيله بالمنفذ USB، ويبين الشكل (٣٩-١) الفلاش ديسك .



الشكل (٣٩-١) - يبين الفلاش ديسك

٧- كرت الذاكرة الآمن SD memory

- تُستخدم هذه الذاكرة عادة في الكاميرات والهواتف النقالة .
- يصل حجمها إلى 16GB أو أكثر .

ويبين الشكل (٤٠-١) كرت الذاكرة الآمن .



الشكل (٤٠-١) - يبين كرت الذاكرة الآمن

١-٧- أداء الحاسوب Computer Performance

يقصد هنا سرعة تنفيذ التعليمات لوحدة المعالجة المركزية CPU، وتتحدد هذه السرعة بعدة عوامل :

١. تردد ساعة الحاسوب Clock Speed
٢. حجم الذاكرة الرئيسية Main Memory
٣. القرص الصلب Hard Disk
٤. المسرى Bus Speed
٥. بطاقة الشاشة Graphics Acceleration Card

أما الميزات التي تميز معالج عن معالج آخر هي :

١- سرعة وحدة المعالجة CPU Speed

يزود كل حاسوب بمولد للنبضات يغذي المعالج بنبضات كهربائية منتظمة تتحكم في تزامن تنفيذ التعليمات تسمى نبضات الساعة، وهي التي تحدد سرعة المعالجة أي عدد العمليات المنفذة في الثانية، وكلما زادت سرعة المعالج كلما أمكن تنفيذ عدداً أكبر من العمليات داخل جهاز الحاسوب بالثانية الواحدة (سرعة معالجة الحاسوب) وتقاس سرعة المعالجة بالهرتز Hz (أي عدد النبضات في الثانية الواحدة) .

٢- مسرى المعطيات Data Bus

وتحدد أكبر عدد من الأرقام الثنائية التي يقوم المعالج بالتعامل معها معاً في كل عملية ، وتقاس بالبت مثلاً: 8bit, 16bit, 32bit, 64bit وكلما كان عدد

الخطوط أكبر ، كان المعالج أسرع في جلب المعلومات ، ويزداد طول هذا المسرى كلما تطور الحاسوب .

٣- مسرى العنوان Address Bus

وتحدد أكبر سعة ذاكرة يمكن التعامل معها وتقاس بالبت 16bit,32bit,64bit . ويزداد طول هذا المسرى كلما تطور الحاسوب.

٤- عرض الكلمة Word size

وتحدد أكبر عدد من الأرقام الثنائية يمكن التعامل معه بعملية واحدة ، أو طول السجل وتقاس أيضا بالبت .

٥- معالج مساعد داخلي Internal CO-processor

يستخدم هذا المعالج المساعد للقيام بالعمليات الحسابية المعقدة مثلاً (الفاصلة العائمة) .

٦- ذاكرة كاش داخلية Internal cache memory

وهي ذاكرة ساكنة SRAM ذات زمن نفاذ قليل ، أسرع بكثير من الذاكرة الرئيسية وتوضع لتسريع المعالجات .

توجد معالجات من صنع شركات متخصصة متعددة ، ويختلف تصميمها

حسب هذه الشركات ، ومن الأسماء المشهورة لهذه المعالجات 80X86

وPentium من إنتاج شركة إنتل Intel ، ومعالجات شركة AMD ، ومعالجات

ريسك (RISC) المستخدمة في أجهزة أبل وماكنتوش .

١-٨- نظام الإدخال والإخراج الأساسي الـ BIOS :

هو نظام الإدخال والإخراج الأساسي اختصاراً لـ Basic Input Output System ، وهو عبارة عن مجموعة من البرامج مهمتها التحكم في عمليات الإدخال والإخراج عبر وحدات الدخل والخرج والمنافذ الأساسية التي كانت موجودة مع نظام التشغيل MS DOS ، حيث يستخدم لإدخال المعطيات إلى الحاسوب وإخراج نتائج معالجة المعطيات . عندما يتم الضغط على زر تشغيل الحاسوب يتم سماع صوت نغمة بدء تشغيل الحاسوب ومن ثم فإن الحاسوب يقوم بما يسمى بـ POST وهو اختصار لـ Power On Self Test أي الاختبار الذاتي عند التشغيل وهي أول وظيفة من وظائف الـ BIOS .

يتكون عمل نظام الإدخال والإخراج الأساسي الـ BIOS من مهمتين أساسيتين وكل منهما تتكون من عدد من الوظائف :

١- المهمة الأولى : وتتكون من عدد من الوظائف هي :

١- عملية فحص العتاد Hardware (المعالج وسجلاته الداخلية والذاكرة الرئيسية RAM و بطاقة العرضالخ) ، حيث يتم عرض حجم الذاكرة الرئيسية بالإضافة إلى الكثير من المعلومات عن نظام الـ BIOS ، وإذا وجد النظام أية أخطاء فإنه يتصرف حسب درجة خطورتها ونوعها ، ففي الأخطاء البسيطة يكتفي بالتنبيه .

٢- بعد ذلك يقوم نظام الـ BIOS بفحص أجهزة الإدخال والإخراج (الأقراص الصلبة - الأقراص المدمجة - المنافذ التسلسلية والتفرعية - المنفذ التسلسلي العام - لوحة المفاتيحالخ) .

٣- بعد ذلك يقوم نظام الـ BIOS بالبحث عن نظام التشغيل ليسلمه مهمة التحكم بالحاسوب ، حيث يبدأ برنامج الإقلاع Bootstrap بالبحث عن

نظام التشغيل على السواقة المرنة أو الصلبة أو الليزرية وينقله إلى الذاكرة الرئيسية RAM حيث يبدأ نظام التشغيل بالتحكم بالحاسوب .
المهمة الثانية : بعد انتهاء المهمة الأولى بكامل وظائفها تبدأ المهمة الثانية ويتم فيها ما يلي:

١. خلال هذه المهمة تحدث عمليات الإدخال والإخراج في الحاسوب جنباً إلى جنب مع نظام التشغيل حيث بدونها لا نستطيع تخزين المعطيات ولا أن نسترجعها.
٢. يستطيع نظام الـ BIOS أن يتحكم في جميع المكونات المادية بلا استثناء ، وإذا أراد أي برنامج التحكم في المكونات المادية فيجب عليه أن يقوم بذلك عن طريق الـ BIOS ، وهذا ليس شرطاً مع بعض البرامج الحديثة .

ملاحظة :

- يمكن استبدال الـ BIOS بأخر جديد ولكن مع الحذر ، حيث لا بد أن يكون الـ BIOS الجديد مناسباً لنوع اللوحة الأم Motherboard المستخدمة وعندها لا يعمل الحاسوب بعد تركيب الـ BIOS الجديد

١-٩- شريحة الـ CMOS

كيف يمكن لنظام الإدخال والإخراج الأساسي الـ BIOS أن يتعامل مع الأنواع المتعددة والمنتشرة من المكونات المادية والتي هي في تطور دائم بشكل يومي ، إذ لا بد له من معلومات عن الحاسوب لأن الـ BIOS لا يستطيع أن يحدد المكونات المادية للحاسوب ، لذلك لا بد أن نعرفه على نوعية العتاد المتوفر في الحاسوب بشكل يدوي ، لذلك تم اللجوء إلى ما يسمى بشريحة RAM حيث يخزن الـ BIOS هذه المعلومات عليها والتي تسمى بشريحة الـ CMOS وهي اختصار

لـ (Complementary Metal-Oxide Semi-conductor) وهي عبارة عن نوع من الذاكرة العشوائية RAM يتم تخزين المعلومات التي يحتاجها الـ BIOS مثل حجم الأقراص الصلبة وما إلى ذلك وتحتاج لبطارية حتى تحتفظ بمحتوياتها ، على عكس من شريحة الـ BIOS حيث تخزن نظام الـ BIOS حتى تسترجعه عند بداية عمل الحاسوب في المرة القادمة ولا تحتاج لبطارية حتى تحتفظ بمحتوياتها . و يتم التخزين على شريحة الـ CMOS معلومات عن حجم ونوع الأقراص المرنة والصلبة وكذلك التاريخ والوقت ولا يزيد حجمها عن مئات الـ Bytes . يمكن تعديل محتويات ذاكرة الـ CMOS بالدخول إلى إعدادات الـ BIOS عن طريق الضغط على زر del عند إقلاع الحاسوب ، مع ضرورة الانتباه إلى أن تغيير الإعدادات بدون معرفة وظائفها قد يعطل الحاسوب عن العمل . إن شريحة الـ CMOS هي ذاكرة RAM يترافق عملها مع ذاكرة الـ BIOS وفي هذه الذاكرة يتم تخزين المحددات الأساسية للحاسوب مثل : الأقراص المرنة والصلبة الموجودة في الحاسوب وسعة كل منها ، ونوع بطاقة المتحكم في الشاشة ، والتاريخ ، والتوقيت ، والقرص الذي يحتوي على نظام التشغيل ، وكلمة المرور إلى النظام Password ، وعناوين البوابات التسلسلية والتفرعية ، وتنظيم الذاكرةالخ) .

ويتم صناعة هذه الذاكرة من مواد أنصاف النواقل CMOS ، وهذ المواد تستهلك طاقة قليلة جداً ، وتتضمن هذه الذاكرة أيضاً ساعة توقيت تعمل دائماً بسبب وجود البطارية الداخلية لإعطاء التوقيت والتاريخ اللازمين لأنظمة التشغيل ، وذلك لتحديد تاريخ إنشاء الملفات أو تاريخ آخر تعديل أجري عليها .

١٠-١ - النفاذ المباشر إلى الذاكرة DMA

Direct Memory Access

يتم نقل المعطيات من/ و إلى وحدات الإدخال والإخراج (I/O) باستخدام طريقة المقاطعة الخارجية بأعلى معدل لنقل المعطيات ، ولكن كثيراً من وحدات (I/O) تتطلب نقل معطيات بمعدل عالٍ لا يمكن تحقيقه بطريق المقاطعة لأنها تتطلب برامج جزئية لخدمة هذه المقاطعة ، وفي هذه الحالة ينقل المعالج المعلومات من وحدات (I/O) إلى سجلات المعالج ، ثم ينقلها إلى الذاكرة الرئيسية ، وبالعكس يتم نقل المعلومات من الذاكرة الرئيسية إلى وحدات (I/O) عن طريق المعالج .

لذلك تم استخدام طريقة النفاذ المباشر إلى الذاكرة DMA وهذه الطريقة تختلف عن طريقة برمجة (I/O) أو مقاطعة (I/O) اللتين يتم فيهما نقل المعطيات من/ و إلى الذاكرة عبر المعالج الصغري وذلك بنقل المعطيات مباشرة ما بين الذاكرة الرئيسية و وحدة (I/O) ، ولتحقيق هذا النقل بطريقة DMA يجب أن تتوفر في المعالج الصغري مداخل ومخارج أيضاً يجب توفر عناصر منطقية إضافية خارجية ، هذه العناصر يطلق عليها وحدة تحكم الـ DMA (DMA Controller) ، وتتضمن سجل عنوان خاص بها وسجل لعدد الكلمات المراد نقلها ، وعناصر منطقية لتحقيق قراءة أو كتابة المعطيات من أو في الذاكرة .

١١-١ - الذاكرة الافتراضية Virtual Memory

يعتمد مفهوم الذاكرة الافتراضية على مبدأ حجز جزء من القرص الصلب واستخدامه كامتداد للذاكرة الرئيسية للحاسوب ، ويتوقف حجمها على المساحة الخالية من القرص الصلب ، ويتولى نظام التشغيل Windows بتغيير حجم المساحة المخصصة للذاكرة الافتراضية تلقائياً ، ويمكن للمستخدم تغيير حجم هذه

الذاكرة بنفسه بشرط أن يكون لديه الخبرة الكافية ، وذلك لمعرفة الحجم المناسب الذي يمكن تخصيصه لتلك الذاكرة . ويمكن تحديد حجم الذاكرة المطلوب عن طريق إتباع الخطوات التالية :

١- نحدد من قائمة / ابدأ / الأمر / إعدادات / ومن ثم نضغط على / لوحة التحكم / ومن ثم نختار / النظام / فيظهر مربع حوار تحت عنوان خصائص النظام .

٢- نقوم بالضغط على زر التثبيت / أداء / فسيظهر قسم / أداء / من مربع خصائص النظام ، الذي يحوي الخانتين الرئيسيتين / وضع الأداء / و / إعدادات متقدمة / .

٣- نضغط من خانة / إعدادات متقدمة / على الزر / الذاكرة الظاهرية / فسيظهر مربع حوار تحت عنوان الذاكرة الظاهرية ، والذي يحتوي على خيارين ، نختار منها / تحديد الإعدادات الخاصة للذاكرة الظاهرية .

٤- نحدد في خانة الحد الأدنى أقل مساحة نريد تخصيصها للذاكرة الظاهرية ، و نحدد في الحد الأقصى أكبر مساحة نريد تخصيصها للذاكرة الظاهرية ، ثم نضغط على الزر موافق .

لقد أسهم استخدام مفهوم الذاكرة الافتراضية في توسيع الذاكرة الرئيسية ، ولكن يعاب على ذلك ببطء في التشغيل حيث يفضل أن تكون الذاكرة الرئيسية كبيرة قدر الإمكان .

١-٢- الحاسوب في حياتنا

أصبح الحاسوب يقتحم كل مجال، وأصبح أحد الدعائم التي تركز عليها حياتنا اليومية فنجد جزءاً أساسياً في الأعمال التجارية والمصانع والمستشفيات والمدارس والأجهزة الحكومية والبنوك والشركات أيضاً في المنازل وغيرها من

المجالات، ولقد ساعدت تقنيات الاتصالات الحديثة على هذا الانتشار الهائل من خلال استخدام الشبكات وخاصة الانترنت، فالحواسيب حولنا في كل مكان وقد تداخلت مع حياتنا اليومية بشكل كبير، ونورد فيما يلي بعض مجالات استخدام الحواسيب :

١. يستخدم في المؤسسات المالية والبنوك

تستخدم الحواسيب في البنوك بصفة رئيسية لفتح حسابات العملاء ومتابعة أعمالهم المصرفية، كما يمكننا الآن أداء الأعمال المصرفية بنفسنا من خلال أجهزة الصراف الآلي (Automatic Teller Machine) ATM، وتوفر العديد من البنوك حالياً خدمة التعامل من خلال شبكة الانترنت.

٢. يستخدم في النقل والمواصلات

تستخدم الحواسيب في حجز تذاكر السفر والتحكم في حركة وسائل النقل المختلفة مثل القطارات والطائرات والسفن ومركبات الفضاء وغيرها.

٣. يستخدم في المجالات التجارية والاقتصادية :

يستخدم الحاسوب في تنظيم شؤون العاملين ونظام الرواتب وإدارة المخزون وفي أعمال الحفظ والأرشفة وأعمال السكرتاريا، وقد ظهر مفهوم التجارة الإلكترونية e-commerce التي تمكن المستخدم من البيع والشراء وإجراء المعاملات التجارية من خلال الانترنت

٤. يستخدم في المجالات التعليمية

انتشرت البرامج التعليمية المتنوعة والتي تستخدم الوسائط المتعددة لتبسيط العملية التعليمية، كما يمكن للمدرس أن يتابع ويوجه ويختبر طلابه من خلال شبكة الحاسوب، كما اتجهت الكثير من المعاهد والجامعات حديثاً لاستخدام تكنولوجيا الحاسوب والانترنت لتطبيق فكرة التعلم عن بعد (Distance Education).

٥. يستخدم في المجالات الطبية

تستخدم الحواسيب في عدة مجالات فتقوم بمراقبة المرضى في وحدات العناية المركزة، وفي عمل مناظير مقطعية للجسم ، ومساعدة الطبيب في تشخيص المرض واقتراح العلاج ووصف الدواء، وفي حفظ واسترجاع خبرات كبار الأطباء للاستفادة منها، وفي صناعة الدواء.

٦. يستخدم في المجالات العلمية والهندسية والفلك

تستخدم الحواسيب في تخطيط وتصميم المدن الجديدة والمباني والجسور وغيرها ، وعمل التجارب المختلفة على هذه التخطيطات والرسومات الهندسية للوصول إلى الشكل المناسب لتنفيذه.

٧. يستخدم في مجال الدعاية والإعلان :

يمكن استخدام برامج الحاسوب المتخصصة في تصميم الإعلانات عن طريق إدخال بيانات الإعلان الأصلية وتنسيقه ومعالجته بتغيير نوع ونمط وحجم ولون الخط والخلفيات، وإضافة التأثيرات المختلفة على النصوص وإدراج وتنسيق ومعالجة الصور والرسومات.

٨. يستخدم في المجال العسكري والأمني :

يستخدم الحاسوب في عمليات تصنيع المعدات الحربية وتوجيه الصواريخ والطائرات والغواصات وتحديد المواقع ، ويستخدم في التجسس وتشفير الرسائل السرية، ويستخدم أيضاً في عمل ومتابعة سجلات المطلوبين أمنياً، وحفظ البصمات وفحص حقائب السفر، والجوازات والمرور والمخالفات والمعامل الجنائية وكشف المتفجرات ، ويستخدم كذلك في حراسة المنشآت الهامة والحساسة.

٩. يستخدم في المنزل

يكاد لا يخلو منزل هذه الأيام من جهاز حاسوب فهو يستخدم في تعليم الأبناء وتنظيم ميزانية الأسرة وحفظ أرقام الهواتف وبيانات أفراد الأسرة والاتصال بالانترنت للحصول على المعلومات، وشراء حاجيات المنزل، وفي الألعاب والتسلية التي تتراوح بين الألعاب التقليدية مثل الشطرنج وسباق السيارات ثلاثي الأبعاد وألعاب الإثارة والحركة.

١٠- يستخدم في الاتصالات والانترنت

تعتمد الكثير من المؤسسات الحكومية والشركات التجارية على أنظمة الاتصالات وتكنولوجيا الحاسبات في تعاملاتها مع الأفراد والعملاء. ومن خلال تكنولوجيا الاتصالات يتمكن الأفراد من إرسال الرسائل الإلكترونية والتحدث عن بعد وتبادل الملفات وتصفح الانترنت.

١١- يستخدم في المكتبات

تعتمد المكتبات الحديثة اليوم على أنظمة آلية متكاملة تغطي أنشطة ووظائف المكتبات بداية من فهرس المكتبة الآلي الذي يمكن الإطلاع عليه والبحث فيه من خلال الدخول على مواقع المكتبة على شبكة الانترنت، ومروراً بباقي الوظائف كالتزويد بالمصادر وبناء وتنمية المجموعات وغيرها من الوظائف كالخدمة المرجعية وتوفير الدوريات الإلكترونية وقواعد البيانات التخصصية وغيرها من التطبيقات التي سوف يتناولها هذا الكتاب بالمزيد من التفاصيل.

١٢- يستخدم في الصناعة

تستخدم الحواسيب بفعالية في التصميم والتصنيع من قبل المهندسين والمصممين بغرض زيادة الإنتاج وتقليل التكلفة وذلك باستخدام الروبوت خاصة للقيام بالأعمال المتكررة والأعمال التي تتطلب دقة عالية والأعمال الخطرة كاللحام والدهان والصناعات المعدنية وصناعة الأجهزة والآلات.

١-١٣ - تصنيف الحواسيب Computer Classifications

يمكن تصنيف الحواسيب حسب :

- الغرض من استعماله .
- حسب حجم الحاسوب .
- حسب نوع المعطيات التي يتعامل معها .
- حسب درجة الوثوقية .
- حسب السرعة .
- حسب سعة الذاكرة .

ويمكن تصنيف الحاسوب إلى الأنواع الرئيسية التالية :

١ - حسب الغرض من استعماله By Purpose

يمكن تصنيفها إلى:

أ- حواسيب الأغراض العامة General Purpose Computers

✓ يستعمل في المجالات العلمية والهندسية

✓ ويستعمل في التصميم الهندسي

✓ ويستعمل في المجالات الطبية والعسكرية

ب- حواسيب الأغراض الخاصة Special Purpose Computers

وهي حواسيب خاصة لغرض واحد صمم من أجله مثل :

✓ التحكم في المركبات الفضائية

✓ التحكم في أجهزة الإنذار المبكر .

٢- حسب نوع المعطيات التي يعالجها

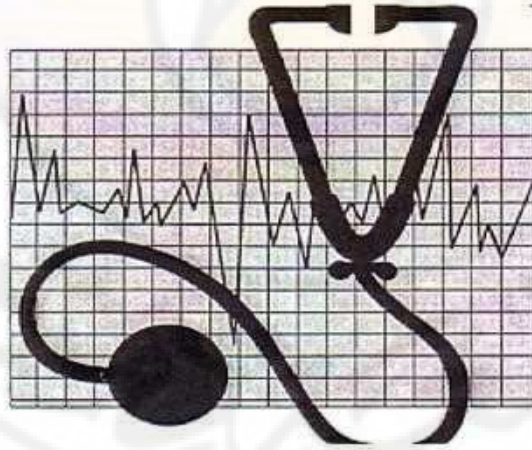
By Type of Data Processed

يمكن تصنيفها إلى:

أ- الحاسوب التماثلي Analogue Computer

يعالج المعطيات التي تتغير باستمرار مثال - درجة الحرارة والضغط الجوي وتمثل بالجهد الكهربائي المتغير داخل الحاسوب التماثلي ، ويبين الشكل (١-٤١) الحاسوب التماثلي . ومن ميزاته :

- ✓ يعمل في الزمن الحقيقي .
- ✓ يمتلك سرعة رد الفعل تجاه التغيرات التي تحدث على المداخل .
- ✓ أهم مكوناته هو مكبر العمليات .



الشكل (١-٤١) - يبين الحاسوب التماثلي

ب- الحاسوب الرقمي Digital Computer

يعالج الحاسوب الرقمي المعطيات والمعلومات الممثلة بنظام العد الثنائي أي على شكل أرقام بالنظام الثنائي ، ويبين الشكل (١-٤٢) الحاسوب الرقمي .

مميزاته :

- القدرة على تخزين البرامج والمعطيات
- الدقة العالية لعدة خانات
- مكوناته البوابات المنطقية

أهم استعمالاته :

- في حل المسائل العلمية والهندسة وفي المجالات التجارية والإدارية .
- في نظم المعلومات الإدارية وفي التحليل الإحصائي .



الشكل (١-٤٢) - يبين الحاسوب الرقمي

ج _ الحاسوب الهجين Hybrid Compute

هو خليط ما بين الحاسوب الرقمي والتماثلي ، ويحتوي على مداخل ومخارج تماثلية ، بينما المعالجة تتم بشكل رقمي ، و يجمع أفضل الصفات والإمكانيات من كلا الحاسوبين الرقمي والتماثلي .

من مميزاتة :

- ✓ سرعة رد الفعل تجاه تغير الدخل .
- ✓ إمكانية التدخل المباشر أثناء الحساب On – Line Control .
- ✓ سهولة توليد التوابع الرياضية بواسطة الحاسب الرقمي .

من مساوئها

- ✓ ذو حجم متعلق بحجم المسألة الواجب حلها .
- ✓ ذو برمجة معقدة .
- ✓ ذو تكلفة مزدوجة .
- ✓ الأخطاء التي قد تحدث دائماً .

٣- حسب الحجم و الأداء

يمكن تصنيفها إلى:

أ- الحاسوب الميكروي Microcomputer

يُعد أصغر حاسوب للأغراض العامة وقد سمي بالحاسوب الميكروي لأنه صمم وركب على أساس المعالج الميكروي Microprocessor ، ويسمى الحاسوب الشخصي PC ، مثل الحاسوب الشخصي من IBM و الحواسيب المتوافقة مع IBM ، وحواسيب Apple و Macintosh ، ويمثل الشكل (١-٤٣) الحاسوب الشخصي .



الشكل (١-٤٣) - يبين الحاسوب الشخصي

ويوجد نوع آخر من الحاسوب الشخصي وهو أصغر حجماً ويسمى الحاسوب المحمول **Laptop** (Portable Computer) ، وهو في حجم الشنطة وهو مناسب لمن يستدعي ظروفهم التنقل باستمرار كأصحاب الشركات والعاملين في البورصة ورجال الأعمال ، والجامعات ، الخ ، ويبين الشكل (١-٤٤) بعض أنواع الحواسيب المحمولة.



الشكل (١-٤٤) - يبين بعض أنواع الحواسيب المحمولة

وهناك نوع آخر هي المساعدات الرقمية الشخصية PDA (Personal Digital Assistants) ، ويطلق عليها أيضاً "حواسيب الكف " Palm PC- وهي أجهزة صغيرة لحفظ الوثائق الهامة والمواعيد وجدول الأعمال والاتصال عبر البريد الإلكتروني، وتتميز بإمكانية إدخال النصوص والأوامر عبر قلم خاص، كما يمكن وصل الأجهزة الكفية عن بعد لاسلكياً بالحاسوب الشخصي أو بالشبكة المحلية، ويبين الشكل (١-٤٥) الحواسيب الكفية .



الشكل (١-٤٥) - يبين الحواسيب الكفية

ب- الحواسيب الصغيرة Minicomputer

تشبه الحواسيب الصغيرة الحواسيب الكبيرة ولكن حجمها أقل حيث تشكل نوعاً وسطاً في حجم الحواسيب وطاقتها ، ويبين الشكل (١-٤٦) الحاسوب الصغير ، واستعملت الحواسيب الصغيرة في البداية كأجهزة متخصصة لأغراض معينة ، وبعد ذلك تم استخدامها للإغراض التجارية والإدارية والعلمية والتعليم وأيضاً للإغراض الخاصة مثل :

- لنحكم في العمليات الصناعية وتوجيه المركبات الفضائية وأجهزة الإنذار المبكر .



الشكل (٤٦-١) - يبين الحاسوب الصغير

ج - الحواسيب الكبيرة Main Frame

وهي حواسيب كبيرة وقوية وسريعة وغالية الثمن ، وتخدم مئات المستخدمين في وقت واحد ، كما تنفذ ملايين التعليمات في الثانية الواحدة، ويتراوح حجمها من حجم خزانتيين للملفات إلى حجم غرفة كبيرة ، وبعض هذه الحواسيب يحتاج إلى حفظها في بيئة محكمة لإبقائها باردة وبعيدة عن الغبار، وذات تكلفة عالية وتمتلك إمكانية أداء عالية ، ويمكن استعمال الحاسوب الكبير كحاسوب مركزي ضمن شبكة حواسيب صغيرة ، واستخدم لمعالجة المعطيات قبل ظهور الحواسيب الميكروية مثل : IBM/370 لشركة IBM ، ويبين الشكل (٤٧-١) الحاسوب الكبير .



الشكل (١-٤٧) - يبين الحاسوب الكبير

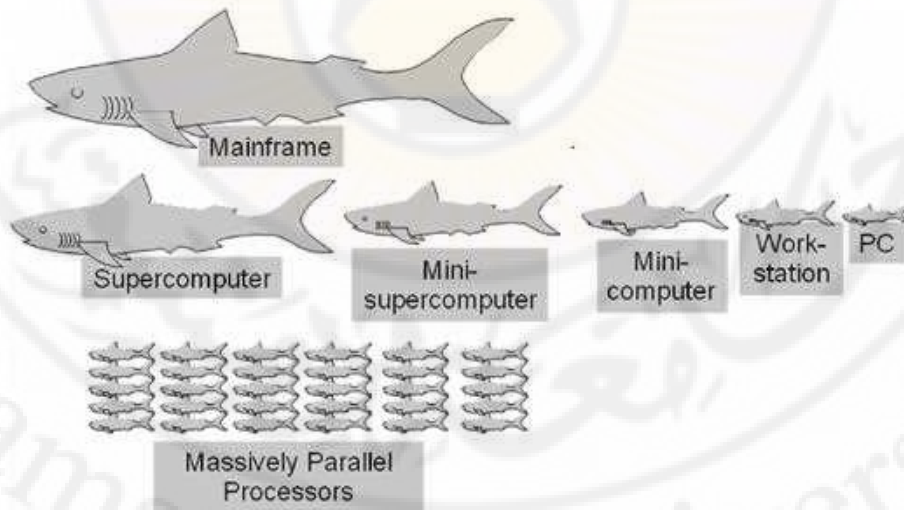
د- الحواسيب الفائقة Super Computer

وهي حواسيب كبيرة جداً تكلف ملايين الدولارات وذات سرعة هائلة، تستخدم في مؤسسات البحث العلمي وعمليات الطيران والفضاء والتنبؤ الجوي، ويبين الشكل (١-٤٨) الحواسيب الفائقة .

وتتماز هذه الحواسيب بأنها أكبر حجماً وأعلى سرعة وأكثر تكلفة وتمتلك مقدرة حسابية ضخمة مثل : CRAY-1 لشركة CDC . وتستعمل في المؤسسات الضخمة ومراكز الأبحاث وغيرها ، ويبين الشكل (١-٤٩) تسلسل تطور الحواسيب .



الشكل (٤٨-١) - يبين الحواسيب الفائقة



الشكل (٤٩-١) - يبين تسلسل تطور الحواسيب

The background of the page features a large, faint watermark of the Damascus University logo. It is a circular emblem with a central yellow and white geometric design resembling a stylized sun or a flower. The top half of the circle contains the Arabic text 'وقل رب زدني علما' (Qul Rabbi Zidni 'Alima) in a calligraphic font. The bottom half contains the Arabic text 'جامعة دمشق' (Jama'at Dimashq) in a similar font. Below the circle, the words 'Damascus University' are written in a serif font.

الفصل الثاني
أنظمة العد وتمثيل المعطيات



أنظمة العد

Number Systems

٢-١- مقدمة

الحاسوب جهاز إلكتروني قابل للبرمجة يستخدم لتخزين المعطيات والتعليمات ومعالجتها، مكون من مجموعة من الدارات الإلكترونية الرقمية التي تستطيع معالجة المعطيات ذات التمثيل الثنائي فقط ، لذلك فمن الضروري تحويل المعطيات المدخلة إلى الحاسوب بأنواعها المختلفة سواء أكانت تعليمات أو أعداد صحيحة أو كسرية أو معطيات غير رقمية (حروف ، صور ، أصوات) بسلاسل من الخانات الثنائية (Bits) والتي تأخذ إحدى الحالتين 0 أو 1 باستخدام الدارات الإلكترونية المستخدمة في بناء العناصر الأساسية للحاسوب.

٢-٢- أنواع أنظمة العد

Type of Number Systems

قبل اختراع الكتابة كان القدماء يقومون بالعد على أصابع اليد فكانت اليد تمثل الأعداد وعند انتهاء الأصابع كانوا يحتاجون إلى شخص آخر ليقوم بالعد فكان الأول يمثل الأحاد و الثاني يمثل العشرات ، وعند اختراع الكتابة اجتهد العلماء لاختراع منظومة أعداد بدلاً من الأصابع فكانت هذه الأعداد هي الأعداد الأساسية وهي (من 0 حتى 9) ، وبإضافة الواحد إلى الصفر يتكون العدد 10 وهذا ما اعتمد عليه العلماء فكانت كل الأعداد بعد العدد 9 مزيج من عددين أو أكثر مثل 10 و 100 و 6735 .

وباعتبار أن الإنسان يملك عشرة أصابع فكان نظام العد العشري أول نظام عد متطور وما يزال يستخدمه الإنسان حتى يومنا هذا.

بما أن نظام العد العشري لا يناسب الخاصية المادية أو الفيزيائية للعناصر المكونة للحاسوب الرقمي فقد تم تطوير أنظمة متنوعة أخرى للعد ، وسوف ندرس في هذا الفصل أنظمة العد الأكثر أهمية واستخداماً وهي:

- نظام العد العشري Decimal Number System.
- نظام العد الثنائي Binary Number System.
- نظام العد الثماني Octal Number System.
- نظام العد السداسي عشر Hexadecimal Number System.
- نظام العد العشري المرمز ثنائياً BCD (Binary Coded Decimal).

يمكن كتابة قانون الصيغة العامة للأعداد وذلك لتحويل الأعداد من نظام عد ما إلى نظام العد العشري كما يلي:

$$N = a_n \times R^n + a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \dots + a_1 \times R^1 + a_0 \times R^0 + a_{-1} \times R^{-1} + a_{-2} \times R^{-2} + \dots + a_{-m} \times R^{-m}$$

حيث:

- a_n : تمثل أحد أرقام نظام العد.
- R : تمثل أساس نظام العد.
- n : تأخذ القيم $(n=0,1,2,\dots,K-1)$ حيث K تساوي عدد الأرقام المكونة للجزء الصحيح للعدد.
- m : عدد الأرقام المكونة للجزء العشري للعدد.

٢-٣- نظام العد العشري Decimal Number System

النظام الرقمي الذي يتعامل معه الإنسان نظام العد العشري ويتكون من عشر أرقام وهي (0,1,2,3,4,5,6,7,8,9) ، ويتكون العدد في نظام العد

العشري من خانة واحدة أو عدة خانات عشرية يتم ترتيبها من اليمين إلى اليسار وتسمى الخانة العشرية الواقعة أقصى يمين العدد بالأحاد ثم يليها العشرات فالمئات فالآلاف وهكذا ، فعلى سبيل المثال: يتكون العدد (517) من ثلاثة أرقام، يمثل الرقم (7) خانة الأحاد، ويمثل الرقم (1) خانة العشرات بينما يمثل الرقم (5) خانة المئات ، ويطلق على خانة الرقم (7) في العدد العشري (517) بخانة الدلالة الدنيا (Least significant Digit) LSD ، بينما يطلق على خانة الرقم (5) في العدد العشري (517) بخانة الدلالة العليا (Most Significant Digit) MSD ، ويمكن التعبير عن العدد العشري بشكل عام حسب قانون الصيغة العامة للأعداد كما يلي:

$$N_{10} = a_n \times 10^n + a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} + \dots + a_1 \times 10^1 + a_0 \times 10^0 + a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \dots + a_{-m} \times 10^{-m}$$

حيث :

a_n : تمثل أحد أرقام نظام العد العشري.

$R=10$: تمثل أساس نظام العد العشري.

وبشكل عام يمكن القول إن أساس أي نظام عد Base يساوي عدد الأرقام المستعملة لتمثيل الأعداد فيه، وهو يساوي كذلك أكبر رقم في النظام مضافاً إليه واحد ، وفي النظام العشري أساسه $B=10$ وأرقامه العشر هي (0,1,2,3,4,5,6,7,8,9) ، ويمكن تمثيل العدد العشري (624.582) باستخدام قانون الصيغة العامة للأعداد كما يلي:

$$624.582 = 6 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 8 \times 10^{-2} + 2 \times 10^{-3}$$

$$624.582 = 600 + 20 + 4 + 0.5 + 0.08 + 0.002 \\ = 624.582$$

تعريف نظام العد :

هو نظام عد يتكون من محددين أساسيين هما :

قاعدة العد (B) Base و الرموز (الأرقام) Symbols

نظام العد العشري : قاعدته $B=10$ وعدد الرموز عشرة من 0 إلى 9 .

مثال (2-1)

الأعداد التالية :

1010 , 9910867 , 0 , 128

هي أعداد بالنظام العشري .

مثال (2-2)

العدد 128 يمكن تمثيله كما يلي :

$$128 = 1 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$$

٢-٤ - نظام العد الثنائي Binary Number System

تستخدم الحواسيب الأعداد الثنائية على الرغم من صعوبة التعامل معها ، وذلك لأنها تتطلب ثلاثة أو أربعة أضعاف عدد الأرقام التي تحتاجها مكافئتها في أنظمة العد الأخرى ، وبما أن مختلف العناصر المكونة للحاسوب الرقمي مبنية على أساس الدارات المنطقية وهي مبنية على أساس الأعداد الثنائية ، والتي تأخذ حالة واحدة من أصل حالتين (0) و (1) فقط ، فقد تم استخدام نظام العد الثنائي الذي يتكون من رقمين هما (0) و (1) ، ويمكن نشر العدد الثنائي حسب قانون الصيغة العامة للأعداد كما يلي:

$$N_B = a_n \times 2^n + a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \dots + a_1 \times 2^1 + a_0 \times 2^0 + a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \dots + a_{-m} \times 2^{-m}$$

حيث :

a_n : يأخذ إحدى القيمتين (0) أو (1).

ويمثل 2 : أساس نظام العد الثنائي .

ويطلق على الخانة الثنائية الواقعة في أقصى اليمين للعدد الثنائي الخانة

ذات الوزن الأدنى (Least Significant Binary) LSB ، بينما يطلق على

الخانة الثنائية الواقعة في أقصى اليسار للعدد الثنائي ذات الوزن الأعلى MSB

(Most Significant Binary) ، ويمكن أن تكون القوة الموجودة على

الرقم (2) تمثل رقم الخانة أو وزنها .

مثال (2-3)

أوجد القيمة العشرية المكافئة للعدد الثنائي التالي $(11101.101)_2$:

الحل:

باستخدام قانون الصيغة العامة للأعداد نجد:

$$\begin{aligned} (11101.101)_2 &= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + \\ &\quad 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 16 + 8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 \\ &= 29.625 \end{aligned}$$

مثال (2-4)

الأعداد التالية :

110011 , 110000 , 000 , 10

هي أعداد في النظام الثنائي .

أما الأعداد التالية :

1200 , 1002 , 001 9

ليست أعداد بالنظام الثنائي (علل ذلك ؟) .

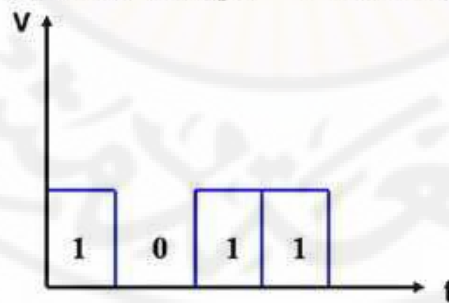
مثال (2-5)

مثل العدد الثنائي $(1011)_2$ بيانياً .

الحل :

يمكن تمثيل العدد $(1011)_2$ بيانياً كما يلي :

0 indicates low voltage, 1 indicates high voltage



٢-٤-١ - التحويل من النظام الثنائي إلى النظام العشري

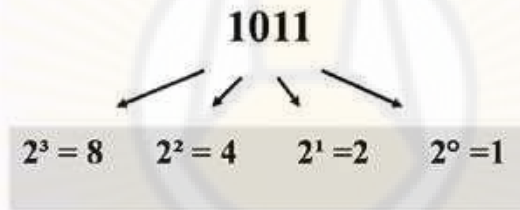
Converting Binary To Decimal Numbers

عند تحويل العدد الثنائي إلى عدد عشري يتطلب جمع جميع أوزان قيم الخانات التي تتضمن (1) ، ويبين المثال (2-6) كيفية تحويل العدد الثنائي إلى مكافئه العشري .

مثال (2-6)

أوجد العدد العشري المكافئ للعدد الثنائي $(1011)_2$ ثم بين أوزان العدد ضمن جدول .

الحل :



$$\begin{aligned}(1011)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 8 + 0 + 2 + 1 \\ &= (11)_{10}\end{aligned}$$

يبين الجدول التالي أوزان العدد الثماني

Binary Digit	1	0	1	1
Weight	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$

مثال (2-7)

أوجد العدد العشري المكافئ للعدد الثنائي $(10111)_2$.

الحل :

$$\begin{aligned}(10111)_2 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 16 + 0 + 4 + 2 + 1 \\ &= (23)_{10}\end{aligned}$$

مثال (2-8)

أوجد العدد العشري المكافئ للعدد الثنائي $(10111)_2$.

الحل :

$$\begin{aligned}(101010101)_2 &= 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 \\ &\quad + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 256 + 0 + 64 + 0 + 16 + 0 + 4 + 0 + 1 \\ &= (341)_{10}\end{aligned}$$

٢-٤-٢ - التمثيل الثنائي للأعداد العشرية

Binary Representation for the Decimal Numbers

لتمثيل الأعداد العشرية من 0 حتى 9 بحاجة إلى أربع خانات (4 bits) ،
وبيين الجدول (٢-١) تمثيل الأعداد العشرية بالنظام الثنائي .

Decimal Numbers	Number in Binary System (using 4 digits)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

الجدول (٢-١) - يبين التمثيل الثنائي للأعداد العشرية

٢-٤-٣ - إجراء العمليات الحسابية على الأعداد الثنائية الموجبة

يمكن إجراء العمليات الحسابية من جمع و طرح و ضرب وقسمة كما هو

الحال في النظام العشري مع مراعاة أن أساس النظام المستعمل هنا هو 2.

١ - عملية الجمع : إذا جمعنا عددين ثنائيين A , B وكان كل منهما يتكون من

خانة واحدة فقط bit ، وبما أن كل خانة يمكن أن تكون إما 0 أو 1 فإنه يوجد

للعددين معاً أربع احتمالات كالآتي:

A	B	المجموع S= A+B	الحامل Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

أما إذا كانت الأعداد الثنائية مكونة من أكثر من خانة واحدة فإن عملية الجمع تنفذ بنفس طريقة الجمع في النظام العشري مع مراعاة أن أساس نظام العد المستعمل هو 2.

مثال (2-9)

أوجد ناتج جمع العددين الثنائيين : $(101)_2 + (011)_2 = (?)_2$

الحل :

$$\begin{array}{r} 101 \\ + 011 \\ \hline 1000 \end{array}$$

$$(101)_2 + (011)_2 = (1000)_2$$

٢- عملية الطرح (في حالة المطروح أقل من المطروح منه): إذا طرحنا عددين ثنائيين A , B وكان كل منهما يتكون من خانة واحدة فقط، فإنه توجد الاحتمالات التالية لعملية الطرح وتكون كالآتي:

A	B	الفرق D=A-B	الاعارة Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

مثال (2-10)

أوجد ناتج طرح العددين الثنائيين : $(101)_2 - (011)_2 = (?)_2$
الحل :

$$\begin{array}{r} 1 \ 0 \ 1 \\ - \ 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 0 \end{array}$$

$$(101)_2 - (011)_2 = (100)_2$$

٣- عملية الضرب: وهي تشبه عملية الضرب في النظام العشري .

مثال (2-11)

أوجد ناتج ضرب العددين الثنائيين : $(101)_2 \times (11)_2 = (?)_2$
الحل :

$$\begin{array}{r} 1 \ 0 \ 1 \\ \times \quad 1 \ 1 \\ \hline 1 \ 0 \ 1 \\ 1 \ 0 \ 1 \\ \hline 1 \ 0 \ 1 \ 1 \end{array}$$

$$(101)_2 \times (11)_2 = (1011)_2$$

٤ - عملية القسمة: وهي تشبه عملية القسمة في النظام العشري .
أوجد ناتج قسمة العددين الثنائيين : $(1001)_2 \div (11)_2 = (?)_2$

الحل :

$$\begin{array}{r}
 11 \\
 11 \overline{) 1001} \\
 \underline{11} \\
 11 \\
 \underline{11} \\
 00
 \end{array}$$

$$(1001)_2 \div (11)_2 = (11)_2$$

تمارين

١ - حول ما يلي من النظام الثنائي إلى مكافئه بالنظام العشري

a. $(1000000001)_2 = (\dots)_{10}$

b. $(110110101)_2 = (\dots)_{10}$

c. $(111111)_2 = (\dots)_{10}$

d. $(1010100)_2 = (\dots)_{10}$

e. $(1010001)_2 = (\dots)_{10}$

٢-٥- التحويل من النظام العشري إلى النظام الثنائي

Conversion From Decimal To Binary

يتم تحويل (تحويل) العدد العشري إلى عدد ثنائي وذلك بتحويل الجزء الصحيح منه بنقسيمه على 2 (أساس نظام العد الثنائي) مع الاحتفاظ بالباقي الصحيح ، وتكرار عملية التقسيم للباقي الصحيح حتى تصبح نتيجة التقسيم صفراً سواء أكان الباقي 0 أو 1، ويبين المثال (2-12) كيفية التحويل .

مثال (2-12)

حول العدد 26_{10} إلى مكافئه بالنظام الثنائي .

الحل:

Integer	Remainder
26	
13	0
6	1
3	0
1	1
0	1

LSD

MSD

- يمثل باقي القسمة الأخير قيمة الخانة الأكثر أهمية MSD
- بينما أول باقي قسمة يمثل قيمة الخانة الأقل أهمية LSD

Answer

$(11010)_2$

$$(26)_{10} = (11010)_2$$

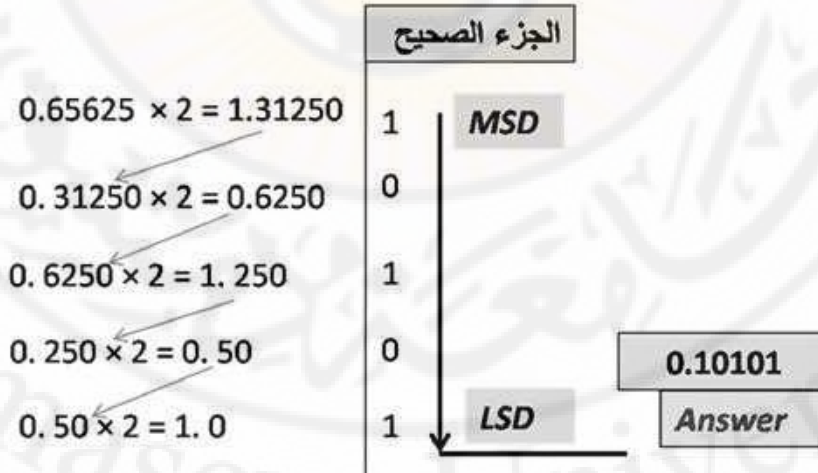
بينما تتم عملية تحويل الجزء العشري (الكسري) منه بضربه بالعدد 2 (أساس نظام العد الثنائي) مع الاحتفاظ بالجزء الصحيح الناتج ، وتكرار عملية الضرب للجزء العشري الناتج ، وعندما تكون النتيجة أقل من 1 نضع نتيجة الضرب 0 ، أما إذا تجاوزت قيمة نتيجة الضرب القيمة 1 ، عندها نطرح من النتيجة القيمة 1 ونضع إلى جانب نتيجة الضرب القيمة 1، ويبين المثال (2-13) كيفية التحويل .

مثال (2-13)

حول العدد $_{10}(26.65625)$ إلى مكافئه بالنظام الثنائي .

الحل :

- نحول الجزء الصحيح كما هو مبين في المثال (2-12)
- نحول الجزء العشري كما يلي :



- نجمع الجزأين الصحيح والعشري ونحصل على الجواب كما يلي :

$$(26.65625)_{10} = (11010.10101)_2$$

تمثل الأرقام 0 و 1 المكتوبة من الأعلى إلى الأسفل إلى جانب نتائج الضرب المتتابع بالقيمة 2 (أساس نظام العد الثنائي) بالعدد الثنائي الناتج ، وعند حساب قيمة تقريبية فإنه كلما زدنا في عمليات الضرب بـ 2 نحصل على دقة أفضل للجزء العشري (الكسري) للعدد $(26.65625)_{10}$.

ملاحظات :

١. للتحويل من أي نظام عد (الثنائي، الثماني، السداسي عشر) إلى نظام العد العشري نستخدم قانون الصيغة العامة للأعداد ، وذلك بنشر العدد على شكل سلسلة من القوى وجمع كافة الحدود .
٢. للتحويل من نظام العد العشري إلى أي نظام عد (ثنائي، ثماني، سداسي عشر) تتم بالتقسيم على أساس ذلك النظام بالنسبة للجزء الصحيح (Integer Part) ، وذلك بان نقوم بتقسيم ذلك الجزء ونواتج عملية القسمة (Quotients) المتتالية على أساس ذلك النظام وأخذ الباقي (Remainders) من كل عملية قسمة ، وتستمر هذه العملية إلى أن يصبح ناتج القسمة الصحيح 0 ، والضرب بأساس هذا النظام بالنسبة للجزء الكسري (Fraction Part) ، وذلك بان نقوم بضرب ذلك الجزء بأساس ذلك النظام للحصول على جزء صحيح وجزء كسري ، و يضرب الجزء الكسري الجديد بأساس ذلك النظام للحصول على جزء صحيح وجزء كسري جديدين ، وتستمر هذه العملية حتى يصبح الجزء الكسري مساوياً للقيمة 0 أو حتى تحقق الأرقام المأخوذة دقة أفضل .

مثال (2-14)

حول العدد $(274.1875)_{10}$ إلى مكافئه بالنظام الثنائي .

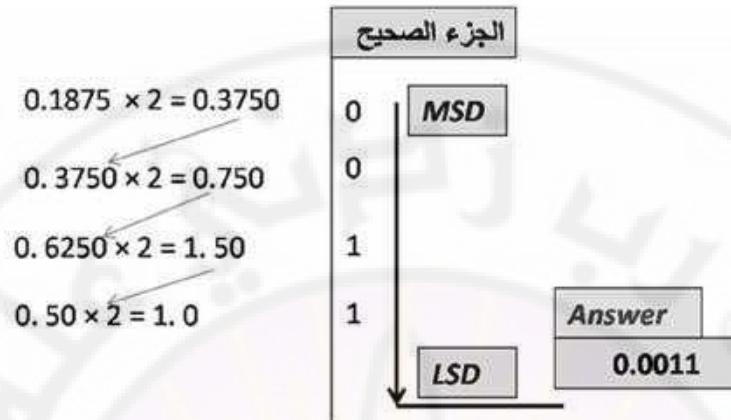
الحل :

- نحول الجزء الصحيح كما يلي :

Integer	Remainder
274	
137	0
68	1
34	0
17	0
8	1
4	0
2	0
1	0
0	1

LSD
 ↑
Answer
 $(274)_{10} = (100010010)_2$
MSD

- نحول الجزء العشري (الكسري) كما يلي :



- نجمع الجزأين ونحصل على الجواب كما يلي :

$$(274.1875)_{10} = (100010010.0011)_2$$

مثال (2-15)

حول العدد $(41.125)_{10}$ إلى مكافئه بالنظام الثنائي .

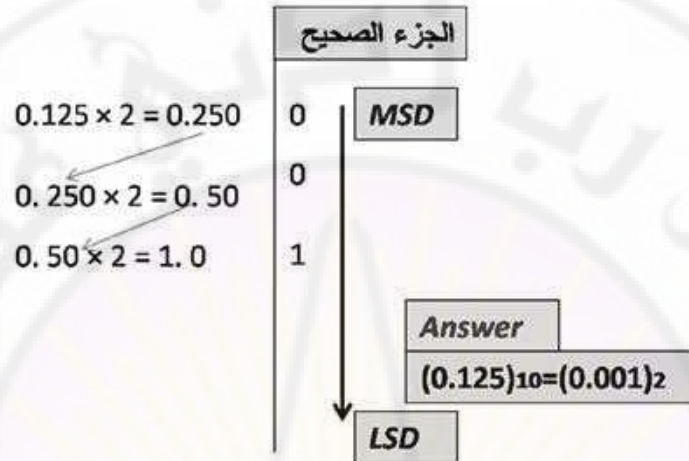
الحل :

- نحول الجزء الصحيح كما يلي :

Integer	Remainder
41	
20	1
10	0
5	0
2	1
1	0
0	1

LSD
 Answer
 $(41)_{10} = (101001)_2$
 MSD

- نحول الجزء العشري (الكسري) كما يلي :



- نجمع الجزأين ونحصل على الجواب :

$$(41.125)_{10} = (101001.001)_2$$

مثال (2-16)

حول العدد $(124)_{10}$ إلى مكافئه بالنظام الثنائي .

الحل :

نحول الجزء الصحيح كما يلي :

$(124)_{10}$

$$124/2 = 62 \text{ remainder } 0$$

$$62/2 = 31 \text{ remainder } 0$$

$$31/2 = 15 \text{ remainder } 1$$

$$15/2 = 7 \text{ remainder } 1$$

$$7/2 = 3 \text{ remainder } 1$$

$$3/2 = 1 \text{ remainder } 1$$

$$1/2 = 0 \text{ remainder } 1$$

LSD

MSD

Answer $(124)_{10} = (1111100)_2$

تمارين

١- حول ما يلي من النظام العشري إلى مكافئه بالنظام الثنائي

a. $(29)_{10} = (\dots)_2$

b. $(2013)_{10} = (\dots)_2$

c. $(2005)_{10} = (\dots)_2$

d. $(1960)_{10} = (\dots)_2$

e. $(450)_{10} = (\dots)_2$

٢-٦- نظام العد الثماني

Octal Number System

يستخدم نظام العد الثماني ثمانية أرقام هي: (0، 1، 2، 3، 4، 5، 6، 7)
وأساس العد هو ($B = 8$) ونلاحظ أن هذه الأرقام تمثل بالنظام الثنائي باستخدام
ثلاث خانات ثنائية (3 bits) .

سبب استخدام هذا النظام ناتج عن قلة عدد الخانات اللازمة لتمثيل أي عدد
بالنظام الثماني عن تمثيله بالنظام الثنائي ، وإن نظام العد الثماني هو نظام عد
متداول بين المختصين أو المتعاملين مع نظم الحواسيب الرقمية .
يمكن التعبير عن العدد الثماني حسب قانون الصيغة العامة للأعداد كما يلي:

$$N_B = a_n \times 8^n + a_{n-1} \times 8^{n-1} + a_{n-2} \times 8^{n-2} + \dots + a_1 \times 8^1 + a_0 \times 8^0 + \\ a_{-1} \times 8^{-1} + a_{-2} \times 8^{-2} + \dots + a_{-m} \times 8^{-m}$$

حيث :

a_n : تأخذ أحد القيم (0,1,2,3,4,5,6,7) .

ويمثل 8: أساس نظام العد الثماني .

مثال (2-17)

١- الأعداد التالية :

1001 , 345 , 0 , 17365

هي أعداد بنظام العد الثماني (علل ؟) .

٢- الأعداد التالية :

8510, 19000 , A1010

ليست أعداد بالنظام الثماني (علل ؟) .

٢-٦-١ - التمثيل الثنائي لنظام العد الثماني

Binary Representation for the Octal Numbers

لتمثيل الأعداد بالنظام الثماني إلى ما يكافئه بالنظام الثنائي والذي يتألف من الأرقام (1,2,3,4,5,6,7) ، بحاجة إلى ثلاث خانات ثنائية (3bits) ،
ويبين الجدول (٢-٢) تمثيل الأعداد الثمانية بالنظام الثنائي .

Decimal Numbers	Number in Binary System (using 3digits)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

الجدول (٢-٢) - يبين تمثيل الأعداد الثمانية بالنظام الثنائي

٢-٦-٢ - التحويل من النظام الثماني إلى العشري

Conversion From Octal To Decimal

يتم تحويل العدد في نظام العد الثماني إلى مكافئه في نظام العد العشري وذلك بجمع جميع أوزان قيم الخانات بعد ضربها بأرقام العدد، حيث تكون القوة موجبة للجزء الصحيح، وسالبة بالنسبة للجزء الكسري ، وذلك باستخدام قانون الصيغة العامة للأعداد.

مثال (2-18)

أوجد العدد العشري المكافئ للعدد الثماني $(1010)_8$ ، ثم مثل أوزان العدد $(725)_8$ في جدول .

الحل :

$$(1010)_8 = 1 \times 8^3 + 0 \times 8^2 + 1 \times 8^1 + 0 \times 8^0 \\ = (520)_{10}$$

يبين الجدول التالي تمثيل أوزان العدد $(725)_8$

Position	2	1	0
weight	$8^2=64$	$8^1=8$	$8^0=1$
Octal digit	7	2	5

Answer

$$D = 5 \times 8^0 + 2 \times 8^1 + 7 \times 8^2 = 5 + 16 + 448 = (469)_{10}$$

مثال (2-19)

أوجد العدد العشري المكافئ للعدد الثماني $(243.44)_8$

الحل :

$$\begin{aligned}
 (243.44)_8 &= 2 * 8^2 + 4 * 8^1 + 3 * 8^0 + 4 * 8^{-1} + 4 * 8^{-2} \\
 &= 2 * 64 + 4 * 8 + 3 * 1 + 4 * \frac{1}{8} + 4 * \frac{1}{64} \\
 &= 128 + 32 + 3 + \frac{4}{8} + \frac{4}{64} \\
 &= 163 + 0.5 + 0.0625 \\
 &= 163.5625
 \end{aligned}$$

٢-٦-٣ - التحويل من النظام العشري إلى الثماني

Conversion From Decimal To Octal

يتم تحويل العدد في نظام العد العشري إلى مكافئه في نظام العد الثماني وذلك بإجراء عملية التقسيم على أساس نظام العد الثماني من أجل الأعداد الصحيحة ، بينما تستخدم عملية الضرب من أجل الأعداد الكسرية كما هو الحال للتحويل من نظام العد العشري إلى النظام الثنائي ولكن تتم القسمة في هذه الحالة على القيمة 8 (أساس نظام العد الثماني) ويقع باقي القسمة بين القيمة 0 والقيمة 7 ، وذلك يتحدد لدينا مجموعة من الأرقام التي تمثل نظام العد الثماني ، و يبين المثال (2-20) كيفية التحويل .

مثال (2-20)

حول العدد العشري $(122)_{10}$ إلى مكافئه الثماني .

الحل :

122		LSD
15	2	
1	7	
0	1	MSD

العدد هو : $(122)_{10} = (172)_8$

مثال (2-21)

حول العدد $(653.40625)_{10}$ إلى مكافئه الثماني .

الحل :

- نحول الجزء الصحيح كما يلي :

653		LSD
81	5	
10	1	
1	2	
0	1	MSD

العدد هو : $(625)_{10} = (1215)_8$

- نحول الجزء الكسري كما يلي :

	الجزء الصحيح
$0.40625 \times 8 = 3.250$	3 MSD
$0.250 \times 8 = 2.0$	2 Answer
0	(0.40625) ₁₀ =(0.32) ₈
	LSD

الجواب هو :

$$(653.40625)_{10} = (1215.32)_8$$

٢-٦-٤ - التحويل من النظام الثماني إلى الثنائي

Conversion From Octal To Binary

إن أهمية استخدام النظام الثماني لتسهيل التخاطب ما بين الإنسان والحواسيب الرقمية ، وفي هذه الحالة يتم تبديل كل رقم ثماني بما يكافئه من الأرقام الثنائية ويعطى كل رقم ثماني بثلاث خانات ثنائية.

الطريقة :

✓ نبذل كل رقم من العدد الثماني ما يقابله من النظام الثنائي من

الجدول (2-2) .

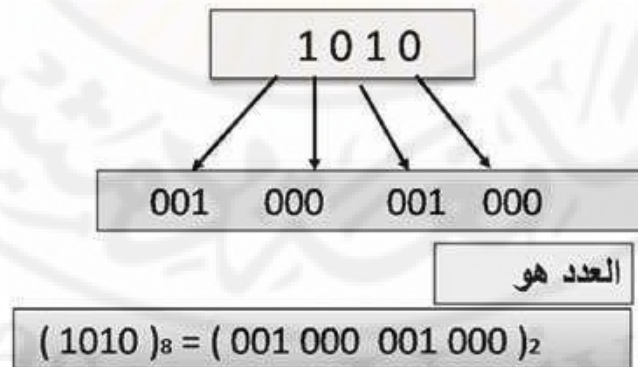
✓ الرقم الكلي الناتج يمثل العدد بالنظام الثنائي .

وبين المثال (2-22) كيفية التحويل .

مثال (2-22)

أوجد العدد الثنائي المكافئ للعدد الثماني $(1010)_8$.

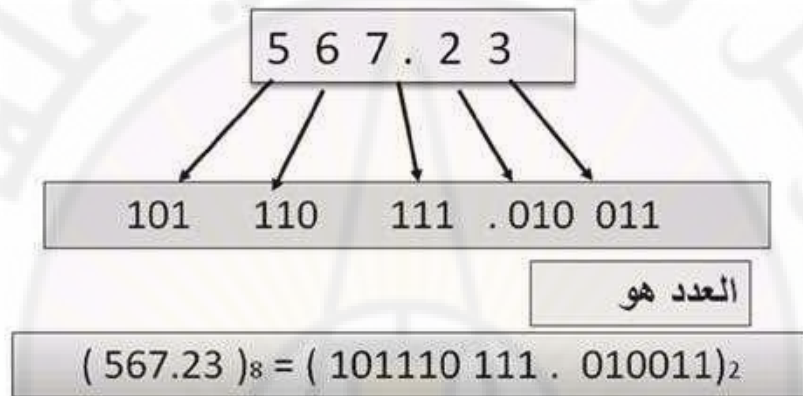
الحل :



مثال (2-23)

أوجد العدد الثنائي المكافئ للعدد الثماني $(576.23)_8$.

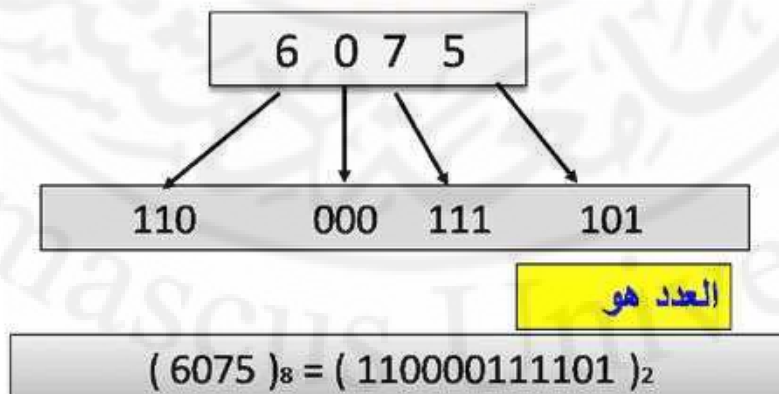
الحل :



مثال (2-24)

أوجد العدد الثنائي المكافئ للعدد الثماني $(6075)_8$.

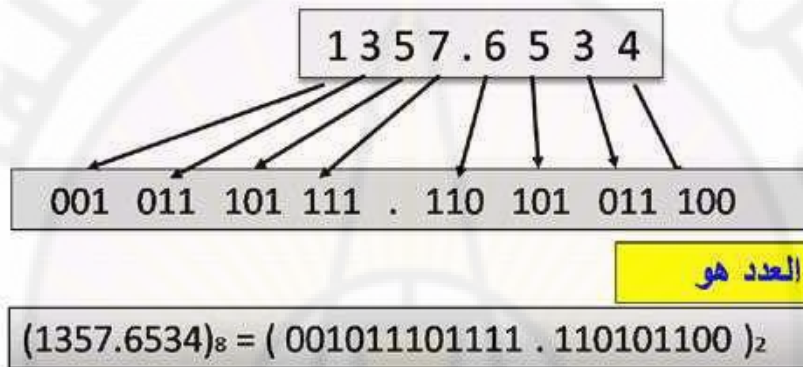
الحل :



مثال (2-26)

أوجد العدد الثنائي المكافئ للعدد الثماني $(1357.6534)_8$.

الحل :



٢-٥- التحويل من النظام الثنائي إلى الثماني

للتحويل من نظام العد الثنائي إلى نظام العد الثماني نقوم بتقسيم العدد الثنائي المعطى إلى مجموعات وكل منها يتألف من ثلاث خانات ثنائية وذلك بدءاً من أقصى اليمين إلى اليسار بالنسبة للجزء الصحيح ، وبتبدأ من أقصى اليسار متجهين نحو اليمين بالنسبة للجزء الكسري وذلك بإيجاد الرقم الثماني المكافئ لكل مجموعة من الخانات الثنائية حسب وزن الخانات $2^0, 2^1, 2^2$ (1 ، 2 ، 4) . في حالة عدم وجود ثلاث خانات أقصى اليسار تضاف أصفار لتكملة العدد وهذا لا يغير من قيمة العدد ، ثم نقوم باستبدال كل مجموعة بما يكافئها في النظام الثماني من الجدول (2-2) ، وتبين الأمثلة التالية كيفية التبديل .

مثال (2-27)

أوجد العدد الثماني المكافئ للعدد الثنائي $(111101001111)_2$.
الحل :

011 110 100 111

⏟ ⏟ ⏟ ⏟

3 6 4 7

العدد هو

$$(111101001111)_2 = (3647)_8$$

مثال (2-28)

أوجد العدد الثماني المكافئ للعدد الثنائي $(1110101.10111)_2$.
الحل :

001 110 101 . 101 11 0

⏟ ⏟ ⏟ ⏟ ⏟

1 6 5 . 5 6

العدد هو

$$(1110101.10111)_2 = (165.56)_8$$

٢-٦-٦- جمع وطرح الأعداد الثمانية

١- جمع الأعداد الثمانية

عند جمع الأعداد الثمانية نتبع الطريقة نفسها في حالة الأعداد العشرية مع مراعاة أن أساس نظام العد هو $B=8$.

مثال (2-29)

أوجد ناتج جمع العددين الثمانيين التاليين :

$$(176.7)_8 + (52.2)_8 = (\dots)_8$$

الحل :

$$\begin{array}{r} 176.7 \\ + 052.2 \\ \hline 251.1 \end{array}$$

$$(176.7)_8 + (52.2)_8 = (251.1)_8$$

٢- طرح الأعداد الثمانية

عند طرح الأعداد الثمانية نتبع الطريقة نفسها في حالة الأعداد العشرية مع مراعاة أن أساس نظام العد هو $B=8$.

مثال (2-30)

أوجد ناتج طرح العددين الثمانيين التاليين :

$$(2005.7)_8 - (756.2)_8 = (\dots)_8$$

الحل :

$$\begin{array}{r} 2005.7 \\ - 0756.2 \\ \hline 1027.1 \end{array}$$

$$(2005.7)_8 - (756.2)_8 = (1027.1)_8$$

تمارين

١- حول ما يلي من النظام الثماني إلى مكافئه بالنظام الثنائي

- a. $(2046)_8 = (\dots)_2$
- b. $(2013.407)_8 = (\dots)_2$
- c. $(2005)_8 = (\dots)_2$
- d. $(1560.065)_8 = (\dots)_2$
- e. $(450)_8 = (\dots)_2$

٢- حول ما يلي من النظام الثنائي إلى مكافئه بالنظام الثماني .

- a. $(11001011011)_2 = (\dots)_8$
- b. $(10101110001.011001110)_2 = (\dots)_8$
- c. $(1000111011101)_2 = (\dots)_8$
- d. $(101101110000)_2 = (\dots)_8$
- e. $(110100001101.11000101)_2 = (\dots)_8$

٢-٧- نظام العد السداسي عشر

Hexadecimal Number System

يتألف نظام العد السداسي عشر من 16 رقماً وهي :

(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , A , B , C , D , E , F)

ومن أساس قاعدة العد $B = 16$.

إن سبب استخدام هذا النظام ناتج عن قلة عدد الخانات اللازمة لتمثيل أي عدد بالنظام السداسي عشر عن تمثيله بالنظام الثنائي ، وكل 4 خانات في النظام الثنائي تمثل برقم واحد في النظام السداسي عشر ، فهو نظام عد متداول بين المختصين أو المتعاملين مع نظم الحواسيب الرقمية ، ويمكن التعبير عن العدد السداسي عشر حسب قانون الصيغة العامة للأعداد كما يلي:

$$N_B = a_n \times 16^n + a_{n-1} \times 16^{n-1} + \dots + a_1 \times 16^1 + a_0 \times 16^0 + a_{-1} \times 16^{-1} + a_{-2} \times 16^{-2} + \dots + a_{-m} \times 16^{-m}$$

حيث : a_n : تأخذ أحد القيم (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) .

ويمثل 16: أساس نظام العد السداسي عشر .

مثال (2-31)

١- الأعداد التالية :

1001 , 3A 5 , 0 , 19965 , AB16 , F3C

هي أعداد بنظام العد السداسي عشر (علل ؟) .

٢- الأعداد التالية :

AACC#0, 1M000 , K1010

ليست أعداد بالنظام السداسي عشر (علل ؟) .

٢-٧-١ - التمثيل الثنائي لنظام العد السداسي عشر

Binary Representation for the Hexadecimal Numbers

لتمثيل الأعداد بالنظام السداسي عشر إلى ما يكافئه بالنظام الثنائي والذي تتألف من الأرقام (1,2,3,4,5,6,7,8,9,A,B,C,D,E,F) ، بحاجة إلى أربع خانات ثنائية (4bits)، ويبين الجدول (٢-٣) تمثيل الأعداد بالنظام الثنائي .

Hexadecimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

الجدول (٢-٣) - يبين تمثيل الأعداد بالنظام الثنائي

٢-٧-٢ - التبدل من النظام السداسي عشر إلى العشري

Conversion From Hexadecimal To Decimal

يتم تحويل العدد في نظام العد السداسي عشر إلى مكافئه في نظام العد العشري وذلك بجمع جميع أوزان قيم الخانات بعد ضربها بأرقام العدد، حيث تكون القوة موجبة للجزء الصحيح، وسالبة بالنسبة للجزء الكسري ، وذلك باستخدام قانون الصيغة العامة للأعداد.

مثال (2-32)

أوجد العدد العشري المكافئ للعدد السداسي عشر $(1010)_{16}$ ، ثم مثل أوزان العدد $(725)_8$ في جدول .

الحل :

$$\begin{aligned}(1010)_8 &= 1 \times 16^3 + 0 \times 16^2 + 1 \times 16^1 + 0 \times 16^0 \\ &= 4096 + 0 + 16 + 0 \\ &= (40977)_{10}\end{aligned}$$

يبين الجدول التالي تمثيل أوزان العدد $(725)_{16}$

Position	2	1	0
Weight	$8^2=64$	$8^1=8$	$8^0=1$
Octal	7	2	55

Answer

$$D = 5 \times 8^0 + 2 \times 8^1 + 7 \times 8^2 = 5 + 16 + 448 = (469)_{10}$$

٢-٧-٣- التحويل من النظام العشري إلى السداسي عشر

Conversion From Decimal To Hexadecimal

يتم تحويل العدد في نظام العد العشري إلى مكافئه في نظام العد السداسي عشر وذلك بإجراء عملية التقسيم على أساس نظام العد السداسي عشر من أجل الأعداد الصحيحة ، بينما تستخدم عملية الضرب من أجل الأعداد الكسرية كما هو الحال للتحويل من نظام العد العشري إلى النظام الثنائي أو الثماني ، ولكن تتم القسمة في هذه الحالة على القيمة 16 (أساس نظام العد السداسي عشر) ويقع باقي القسمة ضمن المجموعة { 0,1,2,3,4,5,6,7,8,9,A,B,C,E,F } ، وبذلك يتحدد لدينا مجموعة من الأرقام التي تمثل نظام العد الثماني ، و يبين المثال (2-33) كيفية التحويل .

مثال (2-33)

حول العدد العشري $(72)_{10}$ إلى مكافئه السداسي عشر .

الحل :

72		LSD
4	8	
0	4	

MSD

Answer

العدد هو : $(72)_{10} = (48)_{16}$

مثال (2-34)

حول العدد العشري $(1256.12)_{10}$ إلى مكافئه السداسي عشر .

الحل :

- نحول الجزء الصحيح كما يلي :

1256		LSD
78	8	
4	E (14)	
0	4	MSD

Answer

العدد هو : $(1256)_{10} = (4E8)_{16}$

- نحول الجزء الكسري كما يلي :

الجزء الصحيح

1 MSD

E

B

8 LSD

Answer

$(0.12)_{10} = (0.1EB8)_{16}$

- نحصل على النتيجة بجمع الجزأين :

$$(1256.12)_{10} = (4E8.1EB8)_{16}$$

٢-٧-٤ - التحويل من النظام السداسي عشر إلى الثنائي

Conversion From Hexadecimal To Binary

إن أهمية استخدام النظام السداسي عشر لتسهيل التخاطب ما بين الإنسان والحواسيب الرقمية ، ولتحويل أي عدد من النظام السداسي عشر إلى النظام الثنائي نتبع ما يلي .

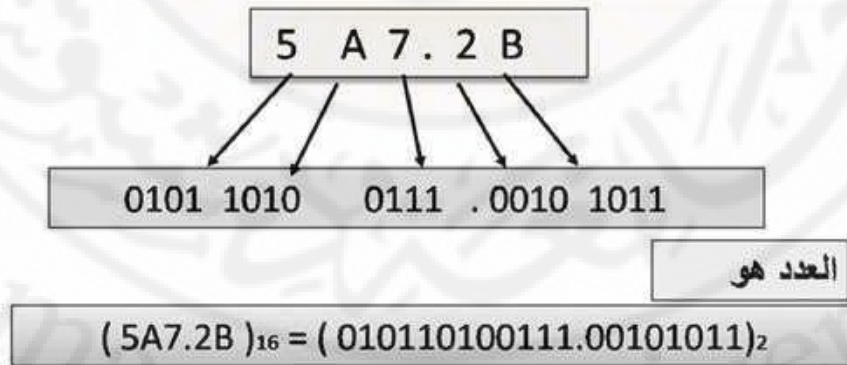
الطريقة :

- ✓ نبذل كل عدد من الرقم السداسي عشر ما يقابله من النظام الثنائي من الجدول (2-3) وكل رقم يتكون من أربعة خانات ثنائية (4bits) .
- ✓ نضم الأرقام الثنائية مع بعضها لنحصل على العدد الثنائي المطلوب .

مثال (2-35)

حول العدد السداسي عشر $(5A7.2B)_{16}$ إلى مكافئه الثنائي.

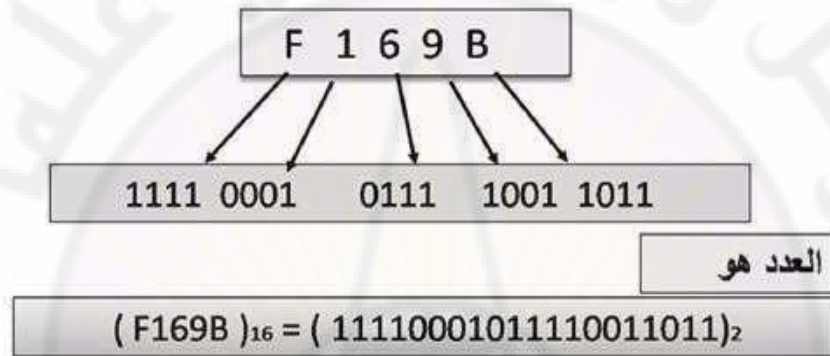
الحل :



مثال (2-36)

حول العدد السداسي عشر $(F169B)_{16}$ إلى مكافئه الثنائي.

الحل :



٢-٧-٥ - التحويل من النظام الثنائي إلى السداسي عشر

Conversion From Binary To Hexadecimal

للتحويل من نظام العد الثنائي إلى نظام العد السداسي عشر نقوم بتقسيم العدد الثنائي المعطى إلى مجموعات وكل منها يتألف من أربع خانات ثنائية وذلك بدءاً من أقصى اليمين إلى اليسار بالنسبة للجزء الصحيح ، و بدءاً من أقصى اليسار متجهين نحو اليمين بالنسبة للجزء الكسري ، وذلك بإيجاد الرقم السداسي عشر المكافئ لكل مجموعة من الخانات الثنائية حسب وزن الخانات الثنائية $2^0, 2^1, 2^2, 2^3$ (1 ، 2 ، 4 ، 8) ، وفي حال عدم وجود أربع خانات ثنائية في أقصى اليسار تضاف أصفار لتكملة العدد وهذا لا يغير من قيمة العدد.

مثال (2-37)

حول العدد الثنائي $(1010 \ 0110 \ 1111100100)_2$ إلى مكافئه السداسي عشر .

الحل :

لتحويل أي عدد صحيح موجب من النظام الثنائي إلى السداسي عشر نتبع الخطوات التالية :

١. نقسم العدد إلى مجموعات وكل مجموعة تتكون من أربع خانات ثنائية.
٢. نبدأ التقسيم من اليمين إلى اليسار
٣. إذا كانت المجموعة الأخيرة غير مكتملة نضيف أصفار إلى يسار العدد لتصبح المجموعة مكونة من أربع خانات .
٤. نقوم باستبدال كل مجموعة بما يكافئها في النظام السداسي عشر

0011	1110	0100	1010	0110
ب	ب	ب	ب	ب
3	E	4	.	A
6				

العدد هو

$$(1111100100 \ . \ 1010110)_2 = (3E4 \ . \ A6)_{16}$$

مثال (2-38)

حول العدد الثنائي $(1011011011001000111)_2$ إلى مكافئه السداسي عشر .

الحل :

0101	1011	0110	0100	0111
⏟	⏟	⏟	⏟	⏟
5	B	6	4	7

العدد هو

$$(1011011011001000111)_2 = (5B647)_{16}$$

تمارين

١- حول ما يلي من النظام السداسي عشر إلى مكافئه بالنظام الثنائي

- a. $(28AB7)_{16} = (\dots)_2$
- b. $(F013.B07)_{16} = (\dots)_2$
- c. $(2013)_{16} = (\dots)_2$
- d. $(2005AA.065)_{16} = (\dots)_2$
- e. $(CE450)_{16} = (\dots)_2$

٢- حول ما يلي من النظام الثنائي إلى مكافئه بالنظام السداسي عشر .

- a. $(11001011011)_2 = (\dots)_{16}$
- b. $(10101110001, 011001110)_2 = (\dots)_{16}$
- c. $(1000111011101)_2 = (\dots)_{16}$
- d. $(101101110000)_2 = (\dots)_{16}$
- e. $(110100001101, 11000101)_2 = (\dots)_{16}$

٢-٧-٦- التحويل من النظام الثماني إلى السداسي عشر

Conversion From Octal To Hexadecimal

لتحويل أي عدد من النظام الثماني إلى السداسي عشر نقوم بتحويله أولاً إلى النظام الثنائي وذلك بتبديل كل رقم من أرقام العدد الثماني إلى مكافئه الثنائي المكون من ثلاث خانات ثنائية ، وبعد ضم الأرقام الثنائية نقوم مرة أخرى بتقسيمها إلى مجموعات مكونة من أربع خانات ونستبدل كل مجموعة بما يكافؤها بالنظام الثماني وبذلك نحصل على العدد المطلوب ، ويبين الشكل (٢-١) طريقة التحويل.



الشكل (٢-١) يبين طريقة التحويل من النظام الثماني إلى السداسي عشر

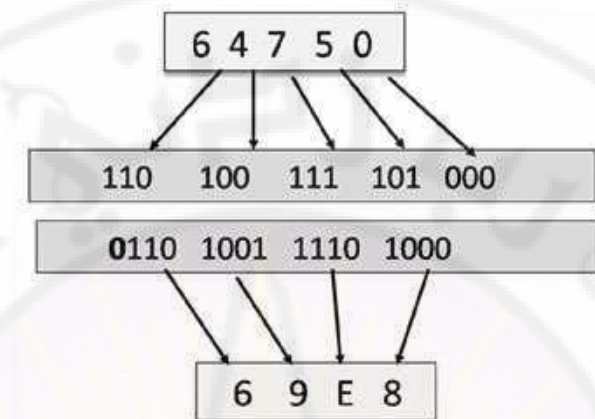
مثال (2-39)

حول العدد الثماني $(64750)_8$ إلى مكافئه السداسي عشر.

الحل :

الطريقة :

- ✓ نحول العدد الثماني إلى الثنائي .
- ✓ نحول العدد الثنائي إلى السداسي عشر .



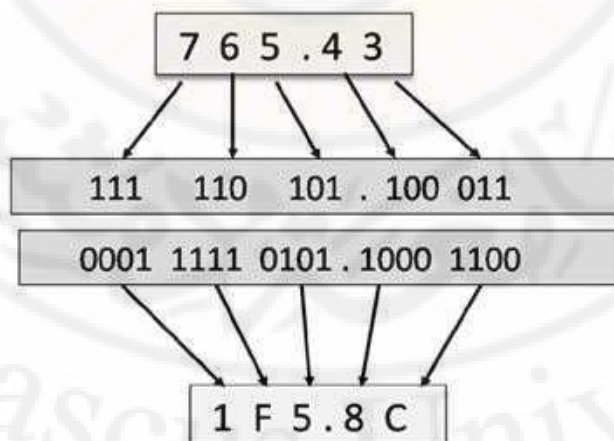
العدد هو

$(64750)_8 = (69E8)_{16}$

مثال (2-40)

حول العدد الثماني $(765.43)_8$ إلى مكافئه السداسي عشر.

الحل :



العدد هو

$(765.43)_8 = (1F5.8C)_{16}$

٢-٧-٧- التحويل من النظام السداسي عشر إلى الثماني

Conversion From Hexadecimal To Octal

لتحويل أي عدد من النظام السداسي عشر إلى الثماني نقوم بتحويله أولاً إلى النظام الثنائي وذلك بتبديل كل رقم من أرقام العدد السداسي عشر إلى مكافئه الثنائي المكون من أربع خانات ثنائية ، وبعد ضم الأرقام الثنائية نقوم مرة أخرى بتقسيمها إلى مجموعات مكونة من ثلاث خانات ثنائية، ونستبدل كل مجموعة بما يكافؤها بالنظام السداسي عشر وبذلك نحصل على العدد المطلوب ، ويبين الشكل (٢-٢) طريقة التحويل.



الشكل (٢-٢) - يبين طريقة التحويل من النظام السداسي عشر إلى الثماني

مثال (2-41)

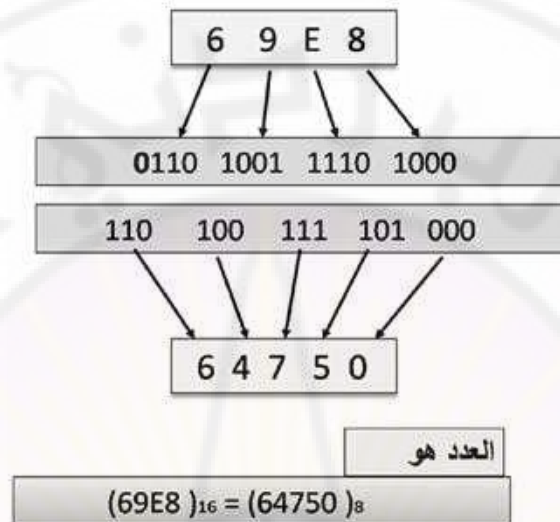
حول العدد السداسي عشر $(69E8)_{16}$ إلى مكافئه الثماني .

الحل :

الطريقة :

✓ نحول العدد السداسي عشر إلى الثنائي .

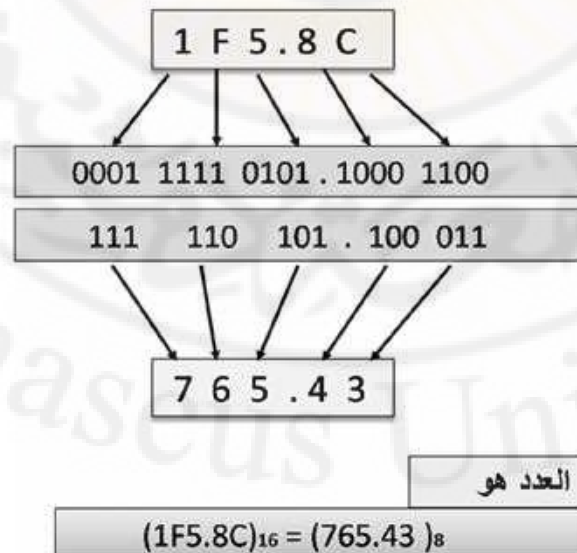
✓ نحول العدد الثنائي إلى الثماني .



مثال (2-42)

حول العدد السداسي عشر $(1F5.8C)_{16}$ إلى مكافئه الثماني.

الحل :



تمارين

١- حول ما يلي من النظام السداسي عشر إلى مكافئه بالنظام الثماني

- a. $(28AB7)_{16} = (\dots)_8$
- b. $(F013.B07)_{16} = (\dots)_8$
- c. $(2013)_{16} = (\dots)_8$
- d. $(2005AA.065)_{16} = (\dots)_8$
- e. $(CE450)_{16} = (\dots)_8$

٢- حول ما يلي من النظام الثماني إلى مكافئه بالنظام السداسي عشر .

- a. $(2007)_8 = (\dots)_{16}$
- b. $(20014 . 2005)_8 = (\dots)_{16}$
- c. $(65705)_8 = (\dots)_{16}$
- d. $(71073)_8 = (\dots)_{16}$
- e. $(608 . 407)_8 = (\dots)_{16}$



٢-٨- النظام العشري المرمز ثنائياً BCD

BCD(Binary Coded Decimal)

يتم في هذا النظام ترميز كل رقم عشري من 0 حتى 9 بعدد معطى بالنظام الثنائي مكون من أربع خانات ثنائية لتمثيل أي رقم عشري. إن استخدام أربع خانات ثنائية 4 bit يمكن من تشكيل 16 حالة مختلفة ($2^4 = 16$) ويتم تمثيل الأرقام العشرية في نظام الـ BCD كما هو مبين في الجدول (٢-٤) .

Decimal Numbers	Number in Binary System (using 4 digits)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

الجدول (٢-٤) - يبين التمثيل الثنائي للأعداد العشرية

ويستخدم الـ BCD نظام العد الثنائي لترميز أو لتمثيل الأعداد العشرية من 0 حتى 9 ، حيث :

- كل عدد عشري بحاجة إلى أربع خانات ثنائية 4 bits
- كفاءة الترميز BCD أقل من كفاءة النظام العشري .
- له ميزة إمكانية استخدامه في الحواسيب الرقمية لأن أرقامه مؤلفة من 0 و 1 ، وفي الجدول (٢-٥) يمثل الترميز BCD لبعض الأعداد العشرية .

Decimal	BCD
22	0010 0010
35	0011 0101
678	0110 0111 1000
9584	1001 0101 1000 0100

الجدول (٢-٥) - يبين الترميز BCD لبعض الأعداد العشرية

٢-٨-١ - عملية الجمع في نظام الـ BCD

عند إجراء عملية الجمع لعددتين مرمزين ثنائياً في نظام الـ BCD، فإذا كان ناتج الجمع أقل أو يساوي القيمة 1001 (أي 9) لا يتم إضافة أي شيء ولكن إذا كان ناتج الجمع أكبر من القيمة 1001، عند ذلك يجب تصحيح الناتج وذلك بإضافة القيمة 0110 (أي 6) والتي تسمى بالستة التصحيحية.

مثال (2-43)

أجرِ عملية الجمع التالية في نظام ترميز الـ BCD.

$$\begin{array}{r}
 0101 \\
 + 0111 \\
 \hline
 1100 \\
 + 0110 \\
 \hline
 10010
 \end{array}
 \quad
 \begin{array}{r}
 5 \\
 + 7 \\
 \hline
 12
 \end{array}$$

1 0010
└──┘ └──┘
1 2

إن ناتج عملية الجمع هو أكبر من القيمة 1001 (أي 6) لذلك يجب إضافة الستة التصحيحية لتصحيح الناتج.
 تم إضافة القيمة 0110 (الستة التصحيحية) لأن الفرق بين أكبر قيمة لأكبر رقم عشري مرمز ثنائياً في ترميز الـ BCD وبين القيمة العظمى المشكّلة من أربع خانات ثنائية وهي 1111 تعادل القيمة 0110 (الستة التصحيحية).

٢-٨-٢ - ملاحظات حول القيمة 0110 (الستة التصحيحية)

- ١- يتم إضافة القيمة 0110 إذا كان ناتج الجمع أكبر من القيمة 1001 ويقع بين القيمة 10 والقيمة 15.
- ٢- يتم إضافة القيمة 0110 إذا كان هناك حامل من الخانة الثالثة إلى الرابعة أي أن ناتج الجمع أكبر من القيمة 15.

مثال (2-44)

أجري عملية الجمع التالية في ترميز الـ BCD:

$$\begin{array}{r}
 + \quad 0111 \\
 1001 \\
 \hline
 1 \quad 0000 \\
 \leftarrow \text{حامل من الخانة الثالثة إلى الخانة الرابعة} \\
 + \quad 0110 \\
 \hline
 1 \quad 0110 \\
 \leftarrow \text{تضاف الستة التصحيحية بسبب وجود حامل من الخانة الثالثة إلى الرابعة} \\
 \hline
 1 \quad 6
 \end{array}$$

يحتوي هذا المثال على حالة وجود حامل من الخانة الثالثة إلى الخانة الرابعة (أكبر من القيمة 15) ولتصحيح ناتج الجمع ينبغي إضافة الستة التصحيحية.

مثال (2-45)

أجر عملية الجمع التالية في ترميز الـ BCD.

$$\begin{array}{r}
 + \begin{array}{ccc} 1001 & 1001 & 1001 \\ 0010 & 0110 & 1001 \end{array} \\
 \hline
 + \begin{array}{ccc} 100 & 0000 & 0010 \\ 0110 & 0110 & 0110 \end{array} \\
 \hline
 \begin{array}{cccc} & 1 & & 1 \\ \underbrace{1}_{1} & \underbrace{0010}_{2} & \underbrace{0110}_{6} & \underbrace{1000}_{8} \end{array}
 \end{array}$$

$$\begin{array}{r}
 + 999 \\
 + 269 \\
 \hline
 1268
 \end{array}$$

مثال (2-46)

أجر عملية الجمع التالية في ترميز الـ BCD .

$$\begin{array}{r}
 + \begin{array}{cc} 1000 & 0111 \\ 0111 & 0011 \end{array} \\
 \hline
 + \begin{array}{ccc} & 1 & \\ 1111 & \uparrow & 1010 \\ 0110 & & 0110 \end{array} \\
 \hline
 \begin{array}{ccc} 1 & 0110 & 0000 \\ \underbrace{}_{1} & \underbrace{}_{6} & \underbrace{}_{0} \end{array}
 \end{array}$$

$$\begin{array}{r}
 + 87 \\
 + 73 \\
 \hline
 160
 \end{array}$$

مثال (2-47)

أجر عملية الجمع التالية في ترميز الـ BCD

$$\begin{array}{r} + \quad 0010 \quad 1000 \\ \quad 0111 \quad 0011 \\ \hline \end{array}$$

$$\begin{array}{r} + \quad 28 \\ \quad 73 \\ \hline \end{array}$$

101

$$\begin{array}{r} + \quad 1001 \quad 1011 \\ \quad \quad \quad 0110 \\ \hline \end{array}$$

$$\begin{array}{r} 1010 \quad 0001 \\ 0110 \quad 0000 \\ \hline \end{array}$$

$$\begin{array}{r} + \quad \begin{array}{c} 1 \\ 1 \end{array} \begin{array}{c} 0000 \\ 0001 \end{array} \begin{array}{c} 1 \\ 0001 \end{array} \\ \begin{array}{c} \underbrace{\quad} \quad \underbrace{\quad} \quad \underbrace{\quad} \\ 1 \quad \quad 0 \quad \quad 1 \end{array} \end{array}$$

=====



٢-٩- الترميز وتمثيل المعطيات (Code)

ترميز المعطيات يعني استخدام رموز أخرى للدلالة على الأرقام أو الحروف، وكما ذكرنا سابقاً عند دراستنا للأعداد الثمانية فإن الترميز الثماني لكلمة ثنائية يمكن أن يتم بتشكيل مجموعات طول كل منها ثلاث خانات مبتدئين من اليمين إلى اليسار ويعطى كل مجموعة رقمها المكافئ الثماني وبهذه الطريقة نحصل على الترميز الثماني للكلمة الثنائية، وبالنسبة للترميز السداسي عشر يتم تشكيل مجموعات طول كل منها أربع خانات مبتدئين من اليمين نحو اليسار ويتم تبديله مباشرة إلى مكافئه الثنائي وبهذه الطريقة نحصل على الترميز السداسي عشر للكلمة الثنائية، ومن ميزات هذا الترميز سهولة قراءة الأعداد، وكذلك العشري المرمز ثنائياً (Binary Coded Decimal) ويرمز له BCD، ويستخدم نظام العد الثنائي لترميز أو لتمثيل الأعداد العشرية من 0 حتى 9. ويوجد بالإضافة إلى الترميز الثماني والسداسي عشر وال BCD ترميز ASCII و ترميز EBCDIC.

٢-٩-١- ترميز ASCII

يعد ترميز ASCII من أهم أنواع الترميز ويسمى " الترميز المعياري الأمريكي لتبادل المعلومات ASCII"، أي:

American Standard Code for Information Interchange

وقد استخدم هذا الترميز لتوحيد التعامل مع الحاسوب باستخدام شيفرة معيارية، وفي هذا الترميز يمثل كل رمز بواسطة 7 خانات أساسية والخانة الثامنة تستعمل كخانة تطابق (Parity Bit) وفي بعض الأجهزة يتم إهمالها، ويوجد نموذج آخر من ASCII يستعمل 8 Bits لتمثيل الرموز.

يمكن تصنيف هذه الرموز (المحارف) إلى مجموعات كما يلي:

١. مجموعة الأرقام من 0 حتى 9 وهي 10 أرقام .
 ٢. مجموعة الأحرف الأبجدية الكبيرة وهي 26 حرفاً من A حتى Z .
 ٣. مجموعة الأحرف الأبجدية الصغيرة وهي 26 حرفاً من a حتى z .
 ٤. مجموعة الرموز الخاصة الموجودة في لوحة المفاتيح وعددها 25 رمزاً .
 ٥. ثلاثون عملية تحكم .
- يوجد عادة جدول خاص لترميز ASCII يبين تمثيل مجموعات الرموز على لوحة المفاتيح المستعملة في أجهزة الحاسوب ، ونلاحظ من الجدول (2-6) أن الرقم العشري " 0 " ممثل بالرقم الثنائي " 00110000 " أو بالنظام السداسي عشر 30H أما الحرف A فممثل بالعدد السداسي عشر 41H والحرف B بالعدد 42H . هذا الترميز يستطيع ترميز $2^7 = 128$ رمزاً أو حرفاً وترميز ASCII يستطيع ترميز $2^8 = 256$ رمزاً أو حرفاً .

٢-٩-٢ - ترميز EBCDIC

يسمى ترميز EBCDIC بالترميز الموسع للأرقام العشرية الممثلة بالنظام الثنائي لتبادل المعلومات أي :

Extended Binary Coded Decimal Interchange Code

هذا الترميز يشبه ترميز ASCII إلا أن عدد الخانات المستخدمة لتمثيل المعلومات (الرموز) هو 8 خانات (8 bits) وأيضاً إلى اختلاف مجموعات الرموز (Character Set) ويستطيع ترميز $2^8 = 256$ رمزاً ، ويمتاز هذا الترميز بما يلي:

١. تستخدم 8 خانات لتمثيل جميع الرموز .
٢. تخصص 4 خانات للنطاق و 4 خانات للرمز .

٣. عدد الرموز الممكن ترميزها 256 رمزاً ولهذا سمي بالترميز الموسع .

٤. تقسم الرموز إلى مجموعات قد يصل عددها إلى 16، وكل مجموعة يمكن أن تحتوي على 16 رمزاً .

٥. يوجد جدول خاص لترميز EBCDIC ، ويبين الجدول (٢-٦) ترميز EBCDIC .

ASCII	SYMBOL	EBCDIC	ASCII	SYMBOL	EBCDIC
00110000	0	11110000	01001110	N	11010101
00110001	1	11110001	01001111	O	11010110
00110010	2	11110010	01010000	P	11010111
00110011	3	11110011	01010001	Q	11011000
00110100	4	11110100	01010010	R	11011001
00110101	5	11110101	01010011	S	11100010
00110110	6	11110110	01010100	T	11100011
00110111	7	11110111	01010101	U	11100100
00111000	8	11111000	01010110	V	11100101
00111001	9	11111001	01010111	W	11100110
01000001	A	11000001	01011000	X	11100111
01000010	B	11000010	01011001	Y	11101000
01000011	C	11000011	01011010	Z	11101001
01000100	D	11000100	00100001	!	01011010
01000101	E	11000101	00100010	"	01111111
01000110	F	11000110	00100011	#	01111011
01000111	G	11000111	00100100	\$	01011011
01001000	H	11001000	00100101	%	01101100
01001001	I	11001001	00100110	&	01010000
01001010	J	11010001	00101000	(	01001101
01001011	K	11010010	00101001	)	01011101
01001100	L	11010011	00101010	*	01011100
01001101	M	11010100	00101011	+	01001110

الجدول (٢-٦) - يبين ترميز ASCII وترميز EBCDIC

الفصل الثالث
البرمجيات ونظم التشغيل



البرمجيات ونظم التشغيل

١-٣ - البرمجيات Software

البرمجيات عبارة عن مجموعة من الأوامر المرتبة منطقياً ، يتم تنفيذها بواسطة وحدة المعالجة المركزية للحاسوب ، والتي تكون مخزنة على شكل ملفات في وحدات التخزين الثانوية ، وهذه البرامج تقوم بالإشراف على كافة المكونات الأساسية للحاسوب ، وتشرف على تسلسل تنفيذ عمليات المعالجة المختلفة ، فالتطور الذي يحدث في أجهزة الحاسوب ومكوناته يصاحبه أيضاً تطور وتحديث دائم في عالم البرمجيات . يمكن أن نعرف البرمجيات كما يلي :

البرمجيات اصطلاح يطلق على جميع البرامج اللازمة لتشغيل الحاسوب وتنظيم عمل وحداته وكذلك تنسيق العلاقة بين هذه الوحدات . ويشمل هذا التعريف نظم التشغيل وكذلك البرامج المعيارية التي يقوم مصنعو الحاسوب بإعدادها والتي تمكن المستخدمين من استغلال عمل الحاسوب على أفضل وجه ، وكذلك يشمل هذا التعريف على البرامج التطبيقية التي تلزم لاستخدام الحاسوب .

٣-١-١ - أنواع البرمجيات

وتتضمن هذه البرمجيات ما يلي:

١ - برمجيات نظم التشغيل Operating Systems

نظام التشغيل برنامج حاسوبي يقوم بضبط وإدارة التحكم بجميع الوحدات الأساسية للحاسوب (المكونات العتادية + وحدات التخزين الثانوية) ، ولا يمكن

الاستغناء عن نظام التشغيل في أي نظام حاسوبي، وبصورة أدق فإن الحاسوب لا يعمل على الإطلاق إذا لم يكن مزوداً بنظام تشغيل، و برمجيات ضرورية لتشغيل الحاسوب وتنظيم عمل الوحدات المحيطية به ، مثل :

Windows(98, 2000, XP,7, NT) , Unix , DOS ,Linux

٢ - البرمجيات التطبيقية Application Software

وهي مجموعة من البرامج الجاهزة يتم كتابتها بإحدى لغات البرمجة لتشكل بيئة يستطيع من خلالها المستخدم التعامل مع الحاسوب ليقوم بإجراء عمليات المعالجة للمعطيات وهذه البرمجيات تساعد الفرد على أداء الأعمال اليومية بكل يسر وسهولة ، نذكر منها :

أ- البرمجيات التطبيقية الجاهزة (Package Software)

وهي مجموعة من البرمجيات المتخصصة والجاهزة التي يمكن أن يستخدمها أي شخص لتنفيذ عمليات محددة على المعطيات، ويبين الشكل (٣-١) برمجيات مايكروسوفت 2007 ، وفيما يلي سنذكر بعض حزم البرمجيات الجاهزة:

- ✓ برمجيات معالجة النصوص Word.
- ✓ برمجيات معالجة الجداول الإلكترونية Excel .
- ✓ برمجيات التصميم AutoCAD .
- ✓ برمجيات قواعد المعطيات Access.
- ✓ برمجيات عرض الشرائح Power Point .
- ✓ برمجيات معالجة الصور Photo Shop .
- ✓ برمجيات التراسل الإلكتروني Outlook Express .



الشكل (١-٣) - يبين برمجيات مايكروسوفت 2007

ب- برمجيات النظم التطبيقية Application Systems

وهذه البرمجيات يتم تصميمها وتطويرها لانجاز عمل محدد ، مثل البرنامج المستخدم لإدخال درجات الطلاب برنامج نظام الرواتب والمحاسبة أو برامج مخصصة لتنفيذ أعمال أخرى .

٣- لغات البرمجة Programming Language

وهي لغات عالية أو متدنية المستوى يتم بواسطتها كتابة البرامج اللازمة للمستثمر لحل المسائل المطلوبة مثل لغات :
BASIC , Pascal , C++ ,
ومن خلالها يستطيع مستخدم الحاسوب كتابة جميع أنواع البرمجيات ، سواء أكانت برمجيات تطبيقية أو برمجيات أخرى .

٤ - البرمجيات الخدمية Utilities Programs

وهي برمجيات تقوم بإجراء الكثير من الوظائف والعمليات لمستخدم الحاسوب ، مثل التحكم في الحاسوب وصيانته بشكل سهل ، وتوفر عليه المعرفة الكاملة لأوامر نظام التشغيل اللازمة لأداء وتنفيذ هذه العمليات ، وكذلك تمكن المستخدم من أداء الكثير من الوظائف التي لا يقدمها نظام التشغيل مباشرة ، وتقوم بعض البرمجيات الخدمية بما يلي :

- ✓ تشخيص المشاكل المتعلقة بالأقراص وحلها .
- ✓ خدمة إصلاح القرص .
- ✓ إعادة بناء القرص واستعادة معطياته بعد مسحه عن طريق الخطأ .
- ✓ ضغط الملفات لتقليل حجم تخزينها في الأقراص .
- ✓ تسريع أداء الجهاز .
- ✓ تكوين النسخ الاحتياطية للأقراص بسرعة وسهولة .
- ✓ حفظ وتأمين الملفات بضغطها أو إخفائها أو حمايتها بكلمة مرور .

٥ - مترجمات لغات البرمجة

يوجد عدد من مترجمات لغات البرمجة ، ويكون لكل لغة برمجة مترجمها الخاص بها هي: المترجم Compiler و المفسر Interpreter والمجمع Assembler .



٣-١-٢- لغات البرمجة

٣-١-٢-١- تطور لغات البرمجة

استخدمت البرمجة كعلم مع ظهور أول حاسوب في الأربعينات حيث تم استخدام البرمجة اليدوية عن طريق اختيار المفاتيح المخصصة ، وبعد ذلك استخدمت اللوحات الإلكترونية بدل المفاتيح ، وتطورت البرمجة مع تطور الحاسوب حيث استخدمت الأرقام الثنائية لبرمجة الحاسوب ، واستخدمت الذاكرة لتخزين هذا البرنامج وقد سمي هذا البرنامج بلغة الآلة Machine Language نظراً لاستخدامها الأرقام الثنائية ، ومع اتساع استعمال الحاسوب أصبح من الصعب استخدام لغة الآلة نظراً لصعوبة تذكر الأرقام ولكبر حجم البرنامج وظهر ما يسمى بلغة الاختصارات ، وفي الفترة التي ظهرت فيها لغة الاختصارات (1952) تم استخدام المترجمات (Compilers) لتقوم بتحويل لغة الاختصارات إلى لغة الآلة ، ثم طرأ بعد ذلك تحسين على هذه اللغة وظهر ما يسمى بلغة التجميع Assembler ، واستمر الإنسان في تطوير لغات البرمجة وتقريبها من لغة الإنسان وظهرت مجموعة كبيرة من لغات البرمجة وسميت بلغات البرمجة عالية المستوى .

٣-٢-١- مستويات لغات البرمجة

تنقسم مستويات لغات البرمجة إلى مستويين :

١- لغات البرمجة المنخفضة المستوى LLL (Low Level Languages)
سميت هذه اللغة بمنخفضة المستوى نظراً لصعوبتها وابتعادها عن لغة الإنسان ، إذ يحتاج المبرمج إلى تحديد السجلات الداخلية للمعالج بشكل مفصل منها :

- لغة الآلة Machine Language
- لغة الاختصارات Mnemonic
- لغة التجميع Assembly Language

و للغات البرمجة منخفضة المستوى الخصائص التالية :

١. لغة صعبة التعلم والفهم .
٢. إن عملية البرنامج بهذه اللغة بطيئة ومتعبة للمبرمج .
٣. صعوبة تعديل البرنامج واكتشاف الأخطاء .
٤. يوجد لكل حاسوب لغة تجميع خاصة به ولهذا لا يمكن تنفيذ نفس البرنامج على أجهزة مختلفة .
٥. تتطلب من المبرمج معرفة كبيرة بالبنية الداخلية للحاسوب

٢- لغات البرمجة العالية المستوى (High Level Languages) HLL

سميت بلغات عالية المستوى نظراً لقربها من لغة الإنسان وهي مكتوبة بلغة مفهومة وتستخدم أوامر سهلة مثل : Read , write , print , go to , ولكن لكي يتم تنفيذ اللغة عالية المستوى في الحاسوب بحاجة إلى مترجم خاص يقوم بتحويل اللغة العالية المستوى إلى لغة الآلة ، على سبيل المثال :

✓ لغة بيزك BASIC

✓ لغة فورتران FORTRAN

✓ لغة Pascal

✓ لغة C++ و Java و C#



أما خصائص لغات البرمجة عالية المستوى هي :

١. سهولة تعلمها واستعمالها لأنها قريبة من لغة الإنسان .
٢. لا تتطلب من المبرمج معرفة كبيرة بالبنية الداخلية للحاسوب .
٣. سهولة كشف الأخطاء وتصحيحها .
٤. سهولة تعديل البرنامج .
٥. الزمن اللازم لكتابة البرنامج أصغر بكثير من اللغات منخفضة المستوى .
٦. معظم لغات البرمجة عالية المستوى لها خاصية التوثيق الداخلي .
٧. معظم لغات البرمجة عالية المستوى متخصصة بعضها يستعمل للأغراض التجارية وبعضها للأغراض الهندسية وبعضها للآتين معاً .

٣-٢-١- بعض أنواع لغات البرمجة

من أهم لغات البرمجة شائعة الاستخدام :

١. لغة البيسك BASIC

وهي لغة بسيطة عامة الأغراض وسهلة التعلم ، يستخدمها المبتدئون في جميع الأعمال، وخاصة في التطبيقات العلمية، وهي اختصاراً للمعنى Beginners All-Purpose Symbolic Instruction Code ولبسطة هذه اللغة واستخدامها في التعليم ظهرت لها عدة إصدارات منها BASIC, GWBASIC, TURBO BASIC, QUICK BASIC. كما ظهرت أيضاً لغة فيجوال بيسك (البيسك المرئي) Visual Basic وهي لغة برمجة مرئية وتعد لغة مطورة من لغة البيسك وهي خاصة لإنتاج برمجيات ذات قدرة عالية وتناسب مع بيئة برنامج ويندوز Windows .

٢. لغة C++/ C

تجمع لغة C ما بين اللغات البنوية Structured Languages ذات المستوى العالي ولغة التجميع Assembly Language ، وتتميز هذه اللغة بالقوة والمرونة والقدرة على إنتاج برمجيات متعددة وذات كفاءة عالية ، وقد ظهرت نسخة حديثة من لغة C ذات بيئة مرئية وهي لغة C++ تتميز بكونها لغة برمجة مرئية Visual تتعامل مع الأغراض Objects.

٣. لغة جافا Java Language

تُعد لغة جافا من اللغات عالية المستوى وتعرف بأنها من اللغات المرئية Visual والغرضية Objects ، وتشبه لغة جافا لغة C++ إلا أنها تتسم بالسهولة ، ولغة جافا من اللغات ذات الأغراض العامة والتي تستخدم لإنتاج برمجيات متنوعة، كما تستخدم لصياغة برامج صغيرة (برامج) تعرف بجافا أبلتس Java Applets.

٤. لغة الكوبول COBOL Language

تستخدم هذه اللغة بصفة رئيسية في الأعمال التجارية مثل البنوك والشركات وهي لغة واسعة الانتشار ، وكلمة كوبول مشتقة من الكلمة Common Business Oriented Language ، وبدأ ظهور هذه اللغة سنة 1959 وقد أجري عليها عدة تعديلات لزيادة كفاءتها وكان آخرها سنة 1974 .

٥. لغة باسكال PASCAL

سميت نسبة إلى العالم الفرنسي في علم الحاسوب Blaise Pascal وهي لغة حديثة يرجع تاريخها إلى 1973 وتستخدم للأغراض العامة وكلغة تعليمية، وعلى الرغم من وضوح بنائها إلا أنها أصعب في التعلم من لغة البيسك * ويبين الجدول (٣-١) برامج بلغتي باسكال وبيزك .

وتُعد لغة باسكال من لغات البرمجة الرئيسية التي تدرس لطلبة المدارس والكلية
نظراً لوضوح السمات الأساسية لتخطيط البرامج البنوية بها .

برنامج مكتوب بلغة باسكال PASCAL لايجاد القيمة العظمى بين قيمتين	برنامج مكتوب بلغة البيسك BASIC يجمع رقمين
<pre> Program pl (input, output); Var a,b, max: Real; Begin: Readln (a,b); IF a > b Then max: = a Else Max: = b; Writeln (max) End.</pre>	<pre> 10 Read A,B 20 Data 10,15 30 Sum = A + B 40 Print Sum 50 End</pre>

لجدول (٣-١) - يبين برنامج بلغتي باسكال وبيزك

٥. لغة اللوغو LOGO

وهي لغة تطبيقات علمية تتميز ببساطتها وسهولة تعلمها وقد صممت
خصيصاً ليستخدمها الأطفال فهي تشجع على الاستخدام المنطقي والتركيب
وتعتمد هذه اللغة على استخدام روبوت صغير يسمى بالسلفاة Turtle من أجل
إبراز استعمالاتها كتعلم الأفكار الحسابية مثل الزوايا والقياسات.

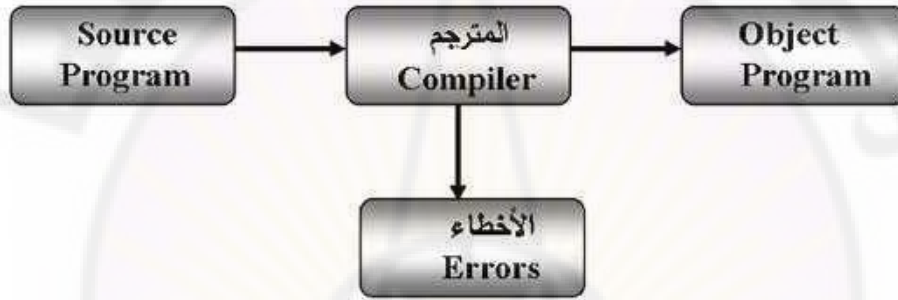
٦. لغات الذكاء الصناعي Artificial Intelligence

هي لغات خاصة بإنتاج حواسيب ذكية تحاكي الإنسان في قدراته الحركية
والبصرية والتحليل والاستنتاج واتخاذ القرارات بناء على نظم الخبرة التي ستغذي
بها الحواسيب ، ومن أهم هذه اللغات لغة برولوج Prolog (يطلق عليها اسم لغة
البرمجة المنطقية Programming in Logic)، ولغة ليسب Lisp (يطلق عليها
اسم لغة برمجة القوائم List Programming Language) .

٣-١-٢-٤ - مترجمات لغات البرمجة عالية المستوى

١. المترجم Compiler

هو عبارة عن برنامج يقوم بتحويل (ترجمة) البرنامج الأولي المكتوب من قبل المبرمج والمكتوب بإحدى لغات البرمجة عالية المستوى إلى برنامج مكتوب بلغة الآلة ، ويبين الشكل (٣-٢) المخطط الصندوقي للمترجم .



الشكل (٣-٢) - يبين المخطط الصندوقي للمترجم

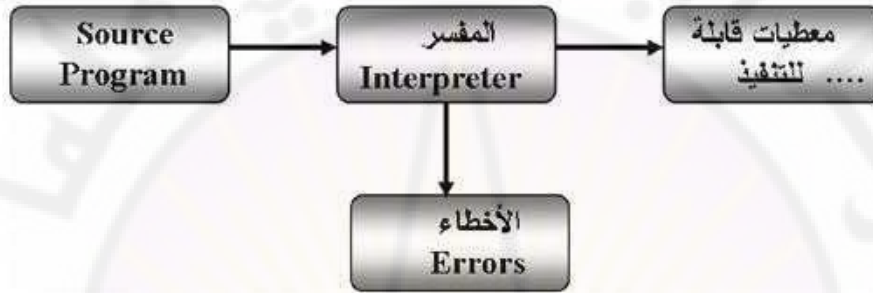
للمترجم عدد من الوظائف هي :

١. ترجمة البرنامج الأولي المكتوب بلغة عالية المستوى دفعة واحدة أي بشكل كامل (البرنامج كاملاً) .
٢. بيان مواقع الأخطاء الإملائية والقواعدية الموجودة في البرنامج وذلك من خلال إصدار قائمة بالأخطاء على الشاشة أو في ملف خاص .
٣. تزويد المستخدم ببعض المعلومات عن البرنامج المترجم كحجم البرنامج وزمن الترجمة .



٢- المفسر Interpreter

وهو عبارة عن برنامج يقوم بترجمة وتنفيذ تعليمات البرنامج الأولي (المصدر) الواحدة تلو الأخرى مع إمكانية اكتشاف الأخطاء الإملائية والقواعدية إن وجدت ، ويبين الشكل (٣-٣) المخطط الصندوقي للمفسر .



الشكل (٣-٣) - يبين المخطط الصندوقي للمفسر

أما الاختلاف بين المفسر والمترجم هي :

١. لا يقوم المفسر بتوليد برنامج الهدف ، لذا نحتاج إلى ترجمة جديدة لبرنامج المصدر عند تنفيذه مرة ثانية .
٢. زمن تنفيذ البرنامج باستخدام المفسر عادة أطول منه في حالة استخدام المترجم .
٣. في حالة استخدام المترجم نحتاج إلى ذاكرة لتخزين برنامج الهدف .
٤. نستخدم المفسر عادة في الحالات التي لا تحتاج فيها إلى تكرار تنفيذ البرنامج لفترات طويلة .
٥. يخزن المفسر عادة في ذاكرة الـ ROM بينما يخزن المترجم في وحدات التخزين المساعدة .
٦. يُعد المترجم أكثر تعقيداً وكلفة من المفسر .

٣-٢- نظم التشغيل Operating System

٣-٢-١- مقدمة

تعد برامج نظام التشغيل من أهم البرامج ويدونها لا يعمل جهاز الحاسوب ، وهو أول برنامج يتم تحميله إلى ذاكرة الحاسوب RAM ، وبعدها يصبح الحاسوب جاهزا لاستقبال وتنفيذ أوامر المستخدم وتحميل وتشغيل البرامج التطبيقية ، وهو الوسيط بين المكونات العتادية لجهاز الحاسوب Hardware وبين البرامج التطبيقية Software ، وبالتالي فهو يوفر بيئة أو واجهة عمل User Interface من خلال واجهة التطبيق السهلة الاستخدام التي تحتوي على النوافذ و القوائم والرموز وغيرها، والتي تمكن المستخدم من تشغيل البرامج التطبيقية دون الدخول في تفاصيل الكيفية التي تعمل بها هذه المكونات. وتقوم بالإشراف على عمليات وحدة المعالجة المركزية ووحدات الإدخال والإخراج، ويبين الشكل (٣- ٤) طبقات البرمجيات .



الشكل (٣- ٤) - يبين طبقات البرمجيات

٣-٢-٢- التطور التاريخي لنظم التشغيل

Historical Development Of Operating System

مرت الحواسيب منذ بداياتها إلى عهدنا الحاضر بالعديد من التطورات و الثورات على مختلف المجالات سواء كان هذا التطور من الناحية الوظيفية أو من الناحية المادية (العتاد) أو بالنسبة للتجهيزات البرمجية ، و أهم المراحل التي مرت بها نظم التشغيل باعتبارها إحدى مكونات التجهيزات البرمجية هي :

١ . لم يكن هناك نظم تشغيل في بداية عهد صناعة الحواسيب ، وذلك لعدم وجود ذاكرة رئيسية في الحواسيب آنذاك .

٢ . ظهر في عام 1953 عمل برمجي يشبه نظم التشغيل سمي بنظام العمل الواحد (Job by Job Processing)، ويُعد الأساس في بناء نظم التشغيل ، وقد اتصف هذا النظام ببطئه الشديد ، حيث كان يحتاج إلى مساعدة بالعمل اليدوي البشري ، سواء كان ذلك لأداء عمليات الإدخال أو عمليات الإخراج ، ولكنه يعد بحق البنية الأساس في بناء نظم التشغيل .

٣ . في عام 1956 ظهر نظام المعالجة الدفعية المبكر (Early Batch Processing) ، وهو نظام معالجة غير مباشر (Off Line) ، يقوم على معالجة أكثر من عمل على التوالي بشرط أن يفصل هذه الأعمال أوامر تحكم عرفت باسم (Job Control Cards) ، أو بطاقات التحكم بالعمل ، وتحمل كل بطاقة من هذه البطاقات أمراً واحداً فقط من أوامر التحكم بالعمل ، وترتبط هذه الأوامر بالبطاقات المنقبة التي تعد الأداة الأولية التي استخدمت لتخزين المعلومات.

٤ . في عام 1960 طرأ تطور كبير في وحدات الإدخال والإخراج ، فكان لزاماً أن يرافق ذلك التطور تطور آخر في برمجيات نظم التشغيل ، وذلك لخدمة وحدات الإدخال والإخراج ، وقد عرف هذا النظام بنظام التحكم بوحدات الإدخال والإخراج (Control Service I/O) ، وبعد عام 1960م بدأت تطورات عديدة تحدث على وسائط التخزين الثانوية المساندة ، ورافق ذلك تطورات على الذاكرة الرئيسية ، بحيث أصبح بمقدور الذاكرة الرئيسية استقبال أكثر من برنامج في نفس الوقت ، وعرف هذا المفهوم بتعدد البرامج (Multiprogramming) ، فظهر على إثر هذا التطور نظام تشغيل قادر على التعامل مع أكثر من برنامج في الوقت نفسه عرف بنظام تشغيل تعدد البرامج (Multiprogramming Operating System) ، ومع تطور معدات الحاسوب ، أصبح بالإمكان اشتراك أكثر من مستخدم الذاكرة نفسها ، بحيث يعطى كل مشترك شريحة زمنية من الوقت الكلي للحاسب ، بحيث ينفذ عمل كل مستخدم بمقدار تلك الفترة الزمنية الممنوحة له ، ويعطى مرة أخرى كل مستخدم لم ينتهي عمله مدة زمنية إضافية لينفذ جزءاً آخر من عمل مستخدم بنفس بمقدار الفترة الزمنية نفسها، وهكذا حتى تنتهي جميع الأعمال المرتبطة بالذاكرة الرئيسية ، وقد عرف هذا النظام بنظام المشاركة بالزمن (Time Sharing System) .

٥. في عام 1972 حدثت ثورة هائلة في تصنيع الذاكرة الثانوية (المساعدة) بحيث أصبح بالإمكان اعتبار الذاكرة الثانوية جزءاً من الذاكرة الرئيسية وذلك حسب ما يقتضي حجم الأعمال الداخلة إلى الذاكرة الرئيسية ، وتعرف هذه الذاكرة باسم الذاكرة الافتراضية (Virtual Memory) ، بحيث أصبح كل حاسوب يملك نظام التشغيل الخاص به .

مثل : AOS/VS , MS DOS , UNIX OS

٦. وفي عام 1985 ظهر أول إصدار لنظام التشغيل windows حيث لم يحقق الإصدار رقم 1 نجاحاً .

٧. في عام 1987 ظهر الإصدار رقم 2 كان يعمل مع كل أنواع الحواسيب الشخصية وفيه ظهرت الرسوم المصغرة وفتح أكثر من نافذة في آن واحد وأنتج له برامج Excel , word من مايكروسوفت .

٨. ثم صدر نظام تشغيل ويندوز رقم 2.10 الذي سمح بتشغيل word و Excel لنظام الـ DOS من windows وذلك في عام 1988 .

٩. في عام 1990 تم إصدار نظام تشغيل ويندوز 3.00 ، ويعد ظهوره البداية الحقيقية لانتشار نظام تشغيل windows ، ومن أهم الميزات :

- ✓ القدرة على العمل مع ذاكرة RAM (16 MB) .
- ✓ دعم العمل مع الشبكات .
- ✓ دعم لعدد اكبر من الطابعات .

١٠. في عام 1992 تم إصدار نظام تشغيل ويندوز رقم 3.10 ، وهو نسخة مطورة عن النسخة السابقة وفي هذا الإصدار قامت مايكروسوفت بتعريب نظام windows إلى اللغة العربية ليصبح هناك نسختين معربتين .

١١. في عام 1993 تم إصدار نظام تشغيل Windows 3.11 هذه النسخة لها إمكانية ذاتية للعمل كنظام تشغيل شبكات والمشاركة في الملفات والطابعات بين عدد من الحواسيب وقد تم تزويدها بالبريد الالكتروني e.mail وجدول تحديد المواعيد schedule .

١٢. في عام 1993 قامت شركة مايكروسوفت ببعض التعديلات على نظام التشغيل os/2 لتطرحه في الاسواق باسم NT أي Windows New Technology في عدة إصدارات:

windows NT 3.1 , windows NT, windows NT 3.5

١٣. في عام 1995 تم إصدار Windows 95 ، وتميز بتغيير واجهة الاستخدام ، ولا يعتمد على نظام تشغيل DOS وأصبح تحميله يتم تلقائياً عند تشغيل الحاسوب وله حماية كاملة للبرامج .
١٤. عام 1998 تم إصدار Windows 98 ، وهو تحسين للإصدار السابق ويتميز بدعم كبير من الملحقات كالطابعات printers ومشغلات الأقراص Disk Drivers ، وله قدرة على استخدام اتصال واحد بالانترنت .
١٥. في عام 2000 تم إصدار Windows 2000 هو الإصدار windows NT5 بعد تغيير اسمه وهو معد للشبكات LAN أو WAN .
١٦. في عام 2000 تم إصدار Windows Me ، وهو موجه لمستخدمي الحاسوب في المنزل ويتميز بعدد من الإضافات أهمها في مجال استخدام الوسائط المتعددة Multimedia .
١٧. في عام 2001 تم إصدار Windows XP فيه مئات من التحسينات والإضافات الجديدة في xp و لم يتجاهل البرامج التي تعمل من خلال DOS أو من خلال الإصدارات السابقة من windows ويتوفر في شكلين :

- ✓ نسخة الاستخدام المنزلي Home Edition .
- ✓ نسخة الأعمال للمحترفين Professional .

اتبعت مايكروسوفت أسلوب جديد لتوفير الحماية لمنتجاتها عبر فحص النسخ المباعة دورياً من خلال الانترنت فكل نسخة من windows xp يتم برمجتها لتقوم بعملية تنشيط على فترات زمنية محددة (30 يوم) يتم إجبار المستخدم على الاتصال بمايكروسوفت عبر الانترنت .

١٨. و في عام 2007 م تم تدشين ويندوز Windows Vista الذي اختلفت الآراء حوله حيث فشل في أول استخدام له عندما طرح بصورة تجريبية إلا أنه استطاع الصمود حتى نال ثقة الكثير من مصممي الحواسيب حول العالم.
١٩. و أصدرت الشركة نفسها نظام التشغيل Windows 7 ، الذي يعد من الإصدارات القوية والذي حمل الكثير من الميزات .

٣-٢-٣- تعريف نظام التشغيل

The Definition of The Operating System

يمكن القول بأن نظام التشغيل (Operating System) هو مجموعة من البرامج المتكاملة التي توفر الاتصال بين المستخدم والحاسوب ، وتستطيع التحكم بالحاسوب وملحقاته ، وتستخدم جميع الموارد الحاسوبية بالطريقة المثلى ، وتستطيع أن توفر واجهة تخاطب (Interface) سهلة وسلسة بين المستخدم والحاسوب بحيث يتمكن المستخدم من الوصول إلى الموارد كافة.

وبكلام آخر يمكن القول بأن نظام التشغيل (Operating System) هو منصة التحكم التي يستخدمها الإنسان (المستخدم العادي أو المبرمج) للسيطرة على الحاسوب وملحقاته ، ومن ثم استثماره وذلك باستخدام مجموعة منتهية من الأوامر (Commands) .

ويطلق على المبرمجين الذين يقومون بكتابة وتطوير برامج نظام التشغيل اسم مبرمجي النظم ، وهم أعلى مستوى في تصنيف المبرمجين ، وهم عادة يعملون في الشركات الكبيرة لتصنيع الحواسيب .

وتختلف مهام نظام التشغيل باختلاف أحجام الحاسوب ، فالحواسيب الكبيرة تحتاج إلى نظم تشغيل عالية الكفاءة تمكنها من التعامل مع العديد من وحدات الحاسوب والوحدات المحيطة به (مثل الطابعات ، ووسائط التخزين وغيرها) ،

وكذلك تمكّنها من توفير إمكانية التشغيل المتعدد المستخدمين للحاسوب في الوقت نفسه ، أما في الحواسيب الشخصية فإن نظم التشغيل تكون أقل تعقيداً ، حيث إنها تتعامل عادة مع حاسوب واحد لمستخدم واحد .

ويمكن تعريف نظام التشغيل على أنه مجموعة البرمجيات الجاهزة المسؤولة عن ضبط وإدارة التحكم بكافة الوحدات الأساسية المكونة للحاسوب وما تحتويه هذه الوحدات من معلومات ومعطيات ، مثل :

- إدارة الذاكرة الرئيسية

- إدارة المعالج

- إدارة وحدات الإدخال والإخراج

- إدارة المعلومات والمعطيات ووسائط التخزين

إذاً هو عبارة عن حزمة من البرامج تقوم بدورين أساسيين:

١. إدارة مكونات الحاسوب المادية (المعالجات ، الذاكرة ،

الأقراص بأنواعها المختلفة) و البرمجية (البرامج و

الملفات و التطبيقات المختلفة).

٢. ربط التطبيقات(البرامج)بالمكونات المادية .

□ عمل نظام تشغيل الحاسوب

يتم تخزين نظام التشغيل عادة على قرص مرّن أو على قرص ضوئي أو على القرص الصلب ، وعند بدء تشغيل الحاسوب يقوم المعالج تلقائياً بالبداية بقراءة وتنفيذ التعليمات المخزنة على ذاكرة القراءة فقط (ROM) ، والتي تقوم بفحص المكونات المادية للجهاز والتأكد من سلامتها ، فإذا ما وجد عطل معين ، يتوقف الحاسوب عن العمل ويعرض رسالة على الشاشة عن العطل الحاصل ،

وأما إذا ما وجد المعالج جميع الوحدات سليمة فيقوم بالبحث عن نظام التشغيل ويبدأ أولاً بالبحث في الأقراص المرنة أو الضوئية على الجهاز ، فإن لم يجد نظام التشغيل هناك ، يقوم المعالج بالبحث في القرص الصلب وبعد التعرف على الوحدة التي تخزن نظام التشغيل يقوم المعالج بتحميله إلى الذاكرة الرئيسية (RAM) ، ثم يبدأ في تنفيذ تعليمات النظام واحدة تلو الأخرى وعادة يقوم النظام بعد تحميله بإعطاء الفرصة لمستخدم الجهاز لإدخال أوامره فمثلاً إذا ما طلب المستخدم تنفيذ برنامج معالج النصوص ، يقوم نظام التشغيل بالبحث عن هذا البرنامج في وحدات التخزين وإذا ما وجده يقوم بتحميله إلى مواقع مخصصة لذلك بالذاكرة الرئيسية (RAM) وينقل التحكم بالمعالج بعد ذلك إلى برنامج معالج النصوص حيث يتم تنفيذ تعليماته من قبل المعالج وعند الانتهاء من تنفيذ البرنامج التطبيقي يعود التحكم تلقائياً إلى نظام التشغيل والذي يسمح لمستخدم الجهاز بإعطاء أوامر جديدة وهكذا ... ، ويمكن تلخيص مهام نظام تشغيل للحاسوب بالخطوات التالية :

١. قراءة وتنفيذ التعليمات والأوامر المخزنة في ذاكرة القراءة فقط (ROM).
٢. فحص وحدات الحاسوب للتأكد من سلامتها .
٣. تحميل نظام التشغيل من الأقراص المرنة أو الضوئية أو من القرص الصلب .
٤. استلام أوامر مستخدم الجهاز .
٥. تحميل البرامج التطبيقية وتنفيذ تعليماتها .
٦. العودة إلى نظام التشغيل وانتظار أوامر المستخدم وتكرار الخطوات السابقة .

□ تصنيف برمجيات نظم التشغيل

تصنف برمجيات نظم التشغيل إلى أربع مجموعات برمجية هي :

١. برمجيات إدارة الذاكرة الرئيسية (Memory Management Programs)
 ٢. برمجيات إدارة المعالجات (Processes Management Programs) .
 ٣. برمجيات وحدات الإدخال والإخراج (Input/Output Management Programs) .
 ٤. برمجيات إدارة المعلومات (Information Management Programs)
- وتتم هذه البرمجيات بلغة الآلة (Machine Language) .

٣-٢-٤ - أنواع أنظمة التشغيل

Operating System Types

تنقسم نظم التشغيل من حيث قدرتها على تشغيل أكثر من برنامج للمستخدم في الوقت نفسه إلى قسمين:

١. أنظمة متعددة المهام Multi-Tasking
 ٢. أنظمة وحيدة المهام Single-Tasking
- كما تنقسم نظم التشغيل من حيث قدرتها على السماح لأكثر من مستخدم بتشغيل برامجها في الوقت نفسه إلى قسمين:
١. أنظمة متعددة المستخدمين Multi-user
 ٢. أنظمة وحيدة المستخدم Single-user
- كما تنقسم إلى :

أ - أنظمة وحيدة المستخدم (Single User O.S)

وهو النظام الذي يسمح لمستخدم واحد فقط بالوصول إلى موارد الجهاز وبرامجه ، ومن الأمثلة على هذه الأنظمة (Windows 95 - DOS) .

ب- أنظمة متعدد المستخدمين (Multi User O.S) :

وهو النظام الذي يسمح لأكثر من مستخدم بالوصول إلى موارد الجهاز وبرامجه ، ومن الأمثلة على هذه الأنظمة (Windows NT – Windows XP – Windows2000 - - Novell Unix) .

٣-٢-٥- واجهة المستخدم في نظم التشغيل

Operating System Modes

أ- الواجهة الخطية (Line Command Interface)

يقصد بالواجهة الخطية استخدام لوحة المفاتيح في إدخال الأوامر والعمليات إلى الحاسوب ، والتي هي مجموعة من النصوص المكتوبة ، وتظهر هذه الأوامر على الشاشة بالإضافة إلى إرسالها إلى الذاكرة الرئيسية ، وقد بدأ استخدام هذه الطريقة من الواجهات منذ بداية ظهور الحاسوب ، واستمر خلال مراحل تطوره حتى الوقت الحالي ، ولهذا فإن معظم نظم التشغيل تستخدم هذه الطريقة من الواجهات في مراحل تطور نظم التشغيل للحواسيب الشخصية ، ثم جرى استبداله فيما بعد بالواجهة الرسومية في نظم التشغيل الحديثة في الحواسيب الشخصية ، ومن المأخذ على هذه الطريقة من الواجهات ضرورة تدريب مستخدم الحاسوب على حفظ أوامر نظم التشغيل ، كما يجب عليه معرفة بعض المصطلحات الانكليزية حتى يستطيع كتابة الأوامر إلى الحاسوب .

وينتج عن استخدام هذه الطريقة من الواجهات كتابة (طباعة) جميع الأوامر والتعليمات ، ومن نظم التشغيل التي تمثل هذه الواجهة (DOS , Unix) .

ب - الواجهة الرسومية (Graphical User Interface)

وهي طريقة من الواجهات تستخدم فيها الفأرة في أغلب الأحيان بدلاً من لوحة المفاتيح ، وعند تشغيل جهاز الحاسوب يجب الانتظار حتى يتم تحميل نظام التشغيل ، وتظهر على الشاشة عدة أشكال على هيئة كائنات رسومية يطلق عليها الأيقونات (Icons) ، وتحت كل شكل اسم التطبيق باللغة المناسبة للمستخدم ، وكل ما على المستخدم هو تحريك مؤشر الفأرة فوق التطبيق المطلوب والضغط على زر الفأرة مرة واحدة ، أو مرتين متتاليتين فيتم تنفيذ التطبيق المطلوب .

لقد تبنت العديد من الشركات الصانعة لأجهزة الحاسوب هذه الطريقة من الواجهات في نظم تشغيل أجهزتها ، وحبذا استخدام هذه الطريقة المستخدمون بمختلف تخصصاتهم ، وذلك لسهولة استخدامها وإمكانية التدريب عليها دون الاعتماد على لغة معينة أو حفظ صيغ الأوامر .

وتظهر جميع الأوامر و التعليمات في هذا النوع من الواجهات على هيئة كائنات رسومية ، ويقوم المستخدم فقط باختيار الأمر (التعليمة) المناسبة لتنفيذها. ومن الأمثلة على هذا النوع (Windows (all versions) .

ت - واجهة القوائم (Menu Interface)

تستخدم هذه الطريقة من الواجهات مفاتيح الأسهم لحركة المؤشر على لوحة المفاتيح لكي يسمح للمستخدم باختيار أحد الخيارات المعروضة بشكل قائمة على شاشة العرض ، ويمثل هذا الخيار تطبيقاً معيناً أو أحد أوامر التشغيل للجهاز ، وبعد الاختيار يتم عرض قائمة أخرى تتضمن قائمة متفرعة عن القائمة الأولى ، وبالتالي يتم انتقال المستخدم من قائمة إلى أخرى عن طريق استخدام مفاتيح حركة الأسهم لتحريك المؤشر لاختيار أحد الاختيارات المعروضة أمامه ، والتي تؤدي إلى تنفيذ أوامر برمجيات معينة من قبل جهاز الحاسوب .

يختار المستخدم الأوامر والتعليمات من خلال القوائم المدرجة ، وتحتوي القوائم على قوائم فرعية وهكذا ، ومن الأمثلة على هذا النوع من الواجهات نظام (Novell) .

مقارنة بين الواجهات الثلاث

- ✓ توفر كلاً من الواجهة الرسومية و واجهة القوائم بيئة عمل سهلة وبسيطة حيث تظهر فيها نوافذ أو قوائم يمكن فتحها أو اختيارها للتشغيل .
- ✓ توفر كلاً من الواجهة الرسومية و واجهة القوائم باستخدام الفأرة أو الأسهم على لوحة المفاتيح للاختيار من قوائم الأوامر بدلاً من إدخالها عن طريق لوحة المفاتيح .
- ✓ توفر كلاً من الواجهة الرسومية و واجهة القوائم سهولة في تبادل المعلومات أو استخدام أكثر من برنامج في وقت واحد .
- ✓ توفر كلاً من الواجهة الرسومية و واجهة القوائم سهولة في الاستخدام إذا ما تم تعلم بعضها حيث تنتقل المهارات المكتسبة في تطبيق معين إلى تطبيق آخر دون الحاجة إلى تعلم أوامر خاصة بكل تطبيق .

٣-٢-٦- بنية أنظمة التشغيل العامة

يقوم نظام الـ BIOS بتحميل ملفات نظام التشغيل إلى ذاكرة الحاسوب الرئيسية (RAM) عند بدء تشغيل الحاسوب ، ووظيفة هذه الملفات هو أن تقوم بتحميل وتشغيل نظام الحاسوب أو إعادته لحالته الأولى، وبعضها يبقى في الذاكرة لإبقاء النظام عاملاً حتى يتم وقف التشغيل ، وباقي الملفات يتم تحميلها عند الحاجة إليها. وعلى العموم فإن أكثر أنظمة التشغيل لها بنية عامة مؤلفة من

مجموعة من الطبقات على شكل دوائر متحدة المركز ، الدائرة الداخلية هي المكونات المادية (العتاد) وهي كما يلي :

١ - النواة (The Kernel)

هو البرنامج الرئيسي القائد ، وهو المسؤول عن عمليات الدخل والخرج الأساسية ويشرف على التعامل مع وسائط التخزين ، وهي عبارة عن مجموعة الملفات التي تقوم بإدارة كل من أسس الأعمال التي لا نراها، مع أنها في الواقع هي الأساس في العمل.

٢ - برنامج الغلاف (The Shell)

هو البرنامج الذي يفسر أوامر المستخدمين ويحولها إلى تعليمات يفهمها المعالج وينفذها ، وتظهر نتائجها للمستخدم ، وهو الذي يعطي رسائل التحذير والخطأ .

٣ - البرامج التطبيقية المساعدة Tools and Application

وهي البرامج المتممة لعمل نظام التشغيل ، مثل برامج تنظيم القرص والتفحص وإلغاء تجزئة القرص وغيرها .

٣-٢-٧ - مهام نظام التشغيل الرئيسية

The Basic Missions of O.S

تختلف مهام نظام التشغيل من حاسوب إلى آخر ، وهي مهام ووظائف متعددة تتعلق بعمليات تشغيل الحاسوب في مكوناته وأجهزته المادية وملحقاته . ومن أهم وظائف نظام التشغيل ما يلي :

١ . التحكم في مسار المعطيات ، حيث يقوم نظام التشغيل بنقل المعطيات داخل الحاسوب من وحدة إلى أخرى ، كما يقوم بتنظيم تبادلها بين الوحدات المختلفة

بجهاز الحاسوب ، وكذلك ينظم عمليات حفظ المعطيات والبرمجيات ، ويحتفظ بمعلومات مفصلة عن حجمها وأماكن حفظها .

٢. تحميل البرامج التطبيقية إلى الذاكرة الرئيسية ، لأن من الوظائف المهمة التي يقوم بها نظام التشغيل تحميل البرامج التطبيقية إلى الذاكرة الرئيسية من الوحدات المحيطية أو من وسائط التخزين المرتبطة بالحاسوب . ويقصد بعملية تحميل البرامج هو نقلها من وسائط التخزين إلى الذاكرة الرئيسية ثم إلى وحدة المعالجة المركزية لتنفيذها ، وبعد تنفيذ البرنامج يقوم نظام التشغيل بإزالة البرنامج من الذاكرة الرئيسية ، وذلك إذا لم يكن البرنامج من البرامج الدائمة في الحاسوب ، وذلك لإفساح المجال لتحميل وتنفيذ برامج أخرى .

٣. تنظيم التسلسل الزمني لتنفيذ العمليات في الحاسوب ، فينظم ترتيباً (دوراً) للمسائل الواردة للمعالجة ويعيد نتائجها إلى أماكن الطلب .

٤. تنفيذ الأوامر الصحيحة الواردة من المستخدم ، وذلك لأن هذه الأوامر هي كلمات تكتب من لوحة المفاتيح أو تحكم بأيقونة أو رموز من واجهة تخطب ببيانة رسومية ، ودور نظام التشغيل هنا هو ترجمة هذه الأوامر إلى تعليمات يفهمها المعالج ويقوم بتنفيذها وإعطاء النتيجة (القيام بالعملية المطلوبة) للمستخدم .

٥. إدارة الذاكرات بمختلف أنواعها واستخدامها على الوجه الأمثل .

٦. إدارة عمليات التخزين على وسائط التخزين وفق بروتوكولات معينة والقراءة منها (إدارة التعامل مع الملفات) .

٧. الإشراف على جميع الموارد وتقاسمها وتنظيمها في حال تعدد المستخدمين أو تعدد الطلب عليها .

٨. التحكم في كامل عمليات الدخل والخرج المختلفة ومراقبتها ، وأيضاً التحكم في وحدات الإدخال والإخراج ، ويشمل ذلك عمليات التحكم في إدخال المعطيات

- عن طريق لوحة المفاتيح أو الفأرة أو غيرها ، وعمليات عرض المعلومات على الشاشة أو إرسالها إلى الطابعة أو أي وحدات أخرى .
٩. التعامل مع البرامج والتطبيقات المختلفة المنفذة على الحاسوب .
١٠. القدرة على التعامل مع شبكات الحواسيب وإدارتها .
١١. حماية التطبيقات والنظام نفسه في حالات التوقف المفاجئ (Abort) والعطب (Failure) ، وذلك عن طريق إضافة برنامج لإدارة المهام (Task Manager) لإغلاق البرمجيات التي لا تستجيب بدلاً من إعادة التشغيل كما في الإصدارات السابقة ، وذلك عن طريق الضغط بنفس الوقت على الأزرار الثلاثة Alt + Ctrl + Del .
١٢. القدرة على إضافة إمكانيات جديدة إلى نظام التشغيل أو إلغاء إمكانيات أصبحت غير مطلوبة (تجزئة نظام التشغيل) ، وذلك دون الحاجة إلى كتابة برنامج جديد لنظام التشغيل ، وتتعلق هذه الخاصية بتصميم برنامج نظام التشغيل حيث يتم تصميمه على شكل تركيبي (Structured) ، أي أن لكل وظيفة يؤديها نظام التشغيل برنامجاً خاصاً بها (Module) فعندما يراد إلغاء أحد الوظائف يتم فصل البرنامج الخاص بها ، كما يمكن ببساطة إضافة برنامج (Module) آخر يحقق وظيفة جديدة .
١٣. القدرة على أداء عدة وظائف في نفس الوقت (تعدد المهام) Multitasking . ويقصد بتعدد المهام هو قدرة نظام التشغيل على تنفيذ أكثر من عملية على الحاسوب في نفس الوقت ، فمثلاً عند تشغيل معالج النصوص ، يمكن في الوقت نفسه طباعة ملف على الحاسوب ، وسماع أغاني على السواق الليزرية .
١٤. التحكم بالوصول إلى الأقراص ووسائط التخزين .

إن المهام السابقة هي مهام عامة ، وقد لا توجد جميعها في نظام تشغيل واحد ولكن معظمها يمكن أن يكون في نظام تشغيل محدد .

□ تنظيم المعلومات داخل الحاسوب

إن أهم مميزات نظام التشغيل هي التخزين الجيد للمعلومات واسترجاعها عند الحاجة ، ويقوم بذلك بواسطة نظام خاص للخرن في ملفات Files ، ويتم تنظيم الملفات ضمن أدلة فرعية (مجلدات) تشكل بنية هرمية (شجرية) في التخزين عادة على الأقراص (مرنة ، صلبة ، ضوئية ، ليزيرية) ويستطيع نظام التشغيل انجاز جميع العمليات على الملفات من حذف وتعديل ونقل وغير ذلك .



٣-٢-٨- أنواع نظم التشغيل الشائعة

٣-٢-٨-١- نظام التشغيل (DOS)

إن كلمة DOS هي اختصار للجملة (Disk Operation System) والتي تعني نظام تشغيل الأقراص ، وهي عبارة عن مجموعة من البرامج و الأوامر التي تنظم طريقة عمل الحاسوب وتسمح للمستخدم من تشغيل الحاسوب الشخصي بحيث يمكن البرامج والتطبيقات من التداخل مع وحدة المعالجة المركزية للحاسوب. وتكون هذه البرامج على مجموعة من الأقراص المرنة ، ويجب أن تحمل إلى ذاكرة الجهاز ليكون بيئة نظام التشغيل لأي برنامج تطبيقي آخر وهناك طريقتان لجعل النظام يعمل تلقائيا بمجرد تشغيل الجهاز وهما :

الطريقة الأولى : وضع نظام التشغيل على القرص الذي يتم تشغيل

الجهاز منه .

الطريقة الثانية : وضع نظام التشغيل على الذاكرة (ROM) .

وبعد نظام التشغيل (DOS) أحد أنظمة التشغيل أحادية المستخدم ، والتي تعتمد بالأساس على الواجهة الخطية للمستخدم ، أي أن المستخدم يقوم بكتابة الأوامر والتعليمات بصورة خطية ، وقد ظهر نظام التشغيل (MS - DOS) لأول مرة في العام 1981 ، وبسرعة كبيرة استطاع هذا النظام أن يحتل المرتبة الأولى في سلم نظم التشغيل المصممة بالأساس للعمل على أجهزة الحاسوب الشخصية (PC) ، ولعل السبب الرئيسي الذي أدى إلى انتشار هذا النظام ونجاحه هو اعتماد أجهزة الحاسوب الشخصي المصنعة من قبل شركة (IBM) والأجهزة المتوافقة معها كنظام تشغيل لأجهزتها .

يطلق المصطلح (PC- DOS) على نظام التشغيل للأجهزة التي تنتجها شركة (IBM) ، في حين يطلق المصطلح (MS-DOS) على نظام التشغيل للأجهزة المتوافقة مع أجهزة (IBM) ، أي الأجهزة التي تستخدم نفس المعالج التي تستخدمه أجهزة (IBM) وتنتجها شركة (Microsoft) . وقد قدمت شركة (Microsoft) العديد من إصدارات هذا النظام وفقا لاحتياجات كل مرحلة ، وتمت عملية تطوير النظام في أكثر من ثماني إصدارات مختلفة حتى توقف إنتاجه في العام 2000 م.

□ الأدلة والملفات (Directories and File)

لكي يتم الوصول إلى المعلومات بسرعة لابد من التنظيم الجيد للعملية التخزينية على السواقات الصلبة أو المرنة أو حتى المدمجة ، فالقرص الواحد يمكن تقسيمه منطقيا إلى مجموعة من الأدلة (Directories) ، وعندها يمكن تخزين الملفات ذات العلاقة المتشابهة مع بعضها البعض ، كما يمكن تقسيم الدليل الواحد إلى مجموعة من الأدلة الفرعية (Subdirectories) .

□ أسماء الأدلة (Directory's Names)

الدليل (الفهرس) هو جزء منطقي يمكن أن تخزن فيه المعلومات على شكل ملفات ، ولكل مجلد أو دليل في البنية الهرمية اسم خاص الذي يميزه عن غيره ، وعند قيام المستخدم بتسمية دليل ما فهناك قواعد لا بد من إتباعها ، وهي :

- ✓ أن لا يزيد عدد الرموز المكون منها اسم الدليل عن ثمانية أحرف.
- ✓ أن لا يكون هناك فراغ (Space) بين هذه الرموز .
- ✓ أن لا نستخدم أيا من الرموز (< > [] ; * ? =) في الاسم .

□ أسماء الملفات (Files Name)

الملف هو تنظيم وتخزين المعلومات والمعطيات لازمان طويلة في الحاسوب وفي مجموعات منطقية مترابطة ، وينقسم اسم الملف بنظام (MS-DOS) إلى قسمين : الأول هو اسم الملف الرئيسي ، وهو الجذر Root في تسمية الملف (بدونه لا يوجد ملف أصلا) ، ولا يزيد عدد الرموز المكونة للاسم الرئيسي عن ثمانية رموز ما عدا الرموز (< > [] ; * ? =) ، والثاني هو اسم الملف الفرعي والذي يعرف اصطلاحا بالامتداد (Extension) ، ولا يزيد عدد الرموز المكونة لاسم الملف الفرعي عن ثلاثة رموز ، ويتم الفصل بين اسم الملف الرئيس واسم الملف الفرعي بالنقطة (.) ، ومن أشهر الملفات الفرعية المستخدمة في نظام التشغيل (DOS) هي :

(TMP,BAK,BIN,DOC,EXE,SYS,TXT,COM) ، بالإضافة

إلى اسم الملف لا بد لنظام التشغيل من معرفة الموقع (القرص و المجلد) الذي يخزن فيه هذا الملف .

□ ملفات نظام التشغيل (DOS) الأساسية

يتكون نظام التشغيل (DOS) من ثلاثة ملفات أساسية :

١. **Input/Output.SYS** : وهو الملف الذي ينظم عمليات الإدخال

والإخراج الأساسية من وحدات الإدخال والإخراج ، حيث يحتوي على التعليمات الخاصة بإدارة عمليات الإدخال والإخراج .

٢. **MS-DOS.SYS** : وهو يسهل تنفيذ برامج التطبيقات المختلفة على

الحاسب الآلي ، وهو البرنامج الخاص بالتعامل مع برامج التطبيقات لمعالجة العمليات الحسابية والمنطقية . ويتم تحميل هذين البرنامجين بمجرد تشغيل الجهاز بواسطة (Boot Disk) ويبقيان في ذاكرة الجهاز بصفة مستمرة ، ولا يظهران في قائمة الملفات عند إعطاء الأمر DIR ، لأنهما من الملفات المحجوبة حرصاً على سلامة التشغيل .

٣. **Command.com** : وهو البرنامج الذي يتلقى الأوامر ويقوم بتوجيهها

للتنفيذ ، وعند تشغيل الجهاز تبحث وحدة المعالجة المركزية عن نظام التشغيل الذي سوف تستخدمه ، حيث يعطي الجهاز الأمر إلى سواقة تشغيل القرص المرن للبحث عن نظام التشغيل ، وفي حالة عدم وجوده ينتقل البحث إلى القرص الليزري فيتم نقله إلى الذاكرة الرئيسية لكي يتلقى الأوامر ويعمل على تنفيذها ، وفي حالة عدم وجوده ينتقل البحث إلى القرص الصلب فيتم نقله إلى الذاكرة الرئيسية لكي يتلقى الأوامر ويعمل على تنفيذها ، أي أن نظام التشغيل يلعب دور المنسق لجميع عمليات الحاسوب ، وهذا الملف هو المسؤول عن تنفيذ أوامر المستخدم عن طريق برامج فرعية كل برنامج منها هو المسؤول عن تنفيذ عملية معينة .

□ أوامر نظام التشغيل (DOS)

تنقسم أوامر نظام التشغيل (DOS) إلى نوعين :

١. **ملفات الأوامر الداخلية :** وهي ملفات الأوامر التي يتم تحميلها لمرة واحدة فقط (عند تشغيل الحاسوب) من وحدة الذاكرة الثانوية إلى وحدة الذاكرة الرئيسية RAM وتبقى ملفات تلك الأوامر في الذاكرة الرئيسية طالما أن جهاز الحاسوب يعمل .

٢. **ملفات الأوامر الخارجية :** وهي الملفات التي يتم استدعاؤها من الذاكرة الثانوية إلى الذاكرة الرئيسية RAM عند الحاجة إليها فقط (طلبها من قبل المستخدم) ، ويقوم نظام التشغيل بالتخلص منها وحذفها من وحدة الذاكرة الرئيسية عند الانتهاء من استخدامها ، وإذا قام المستخدم بطلب أي من هذه الأوامر مرة أخرى أعيد تحميلها إلى الذاكرة الرئيسية وهكذا .

□ محث نظام التشغيل دوس (DOS Prompt)

بمجرد تحميل ملفات نظام التشغيل (DOS) وعمل إعادة تشغيل لجهاز الحاسوب يظهر محث (DOS) وهو الموضع الذي يبدأ منه كتابة أوامر (DOS) سواء الداخلية أو الخارجية ، وإذا كان الإقلاع (Booting) من سواقة الأقراص المرنة يظهر الشكل التالي :

A:\> :

وإذا كان من سواقة الأقراص الصلبة يظهر الشكل التالي :

C:\> :

ويمكن الانتقال بين سواقات الأقراص المرنة والصلبة بكتابة اسم السواقة المراد الانتقال إليها كما يلي :

C:\> A:

ثم الضغط على مفتاح الإدخال أو العكس كما يلي :

A:\> C:

ثم الضغط على مفتاح الإدخال .

دليل الجذر : يرمز له بـ \ ويلحق بعده اسم محرك الأقراص المغناطيسية B , A

... , E , C , .

□ وظائف نظام التشغيل

١. بدء عملية إقلاع (تشغيل) جهاز الحاسوب و القيام بمجموعة من

الاختبارات للتأكد من وجود عتاد الحاسب و سلامة التوصيلات الكهربائية

التي تصل الوحدات المختلفة بعضها بعضاً.

٢. قبول و تنفيذ أوامر التشغيل و أوامر الحاسوب التي يتم إدخالها عن طريق

لوحة المفاتيح والمسؤولة عن التحكم في جميع أجزاء الحاسوب.

```
VERKNÜ~1 LNK          375  29.10.01  19:49 Verknüpfung mit Nfs2se
11 Datei(en)          770.817 Bytes
2 Verzeichnis(se)     377.380.864 Bytes frei

C:\WINDOWS\Desktop>mem

Speichertyp      Gesamt  =  Belegt  +  Frei
-----
Konventionell    640K      63K      577K
Oberer           0K        0K        0K
Reserviert       0K        0K        0K
XMS-Speicher     65.472K   ?      129.860K
-----
Speicher gesamt  66.112K   ?      130.437K
Ges. unter 1 MB  640K      63K      577K

EMS-Speicher gesamt      64M (67.108.864 Bytes)
Freier EMS-Speicher      16M (16.777.216 Bytes)

Max. Größe für ausführbares Programm      577K (590.352 Bytes)
Größter freier oberer Speicherblock        0K      (0 Bytes)
MS-DOS ist resident im hohen Speicherbereich (HMA).

C:\WINDOWS\Desktop>
```

الشكل (٣-٥) يبين شاشة نظام DOS

٣. تحويل و نقل المعطيات و التعليمات بين الذاكرة الرئيسية و الأقراص المرنة.

٤. تمكين المستخدم من الحصول على نسخ إضافية من الملفات و نقلها من

٥. نقل التعليمات والمعطيات إلى الطابعة.

ويعين الشكل (٣-٥) شاشة نظام DOS

□ مميزات DOS بإصداراته المختلفة

١. السرعة العالية : حيث يعد نظام التشغيل DOS من أسرع النظم.

٢. صغر الحجم: يمتاز أصل النظام بالصغر كما يمتاز بالصغر عند التنفيذ

بحيث أنه يتناسب مع الأجهزة القديمة التي تحتوي على ذاكرة عشوائية صغيرة جداً.

٣. إمكانية التعامل مع الشبكات المحلية.

٤. إمكانية التعامل مع الأقراص المرنة ذات الوجهين.

٥. تم في الإصدار MS-DOS 5.0 إضافة تحسينات على برنامج DOS

SHELL و استخدام منسق نصوص جديد كما أضيفت تحسينات كثيرة

للاستفادة من الذاكرة والتعامل مع الذاكرة الإضافية، و أضيفت أوامر

جديدة لاسترجاع الملفات المحذوفة أو إعادة القرص إلى حالته السابقة قبل

إعادة تشكيله ، كما أضيفت تسهيلات في التعامل مع سطر الأوامر تتمثل

في الحصول على معلومات مساعدة عن أي أمر أو تخزين الأوامر

السابقة واسترجاعها و تعديلها بسهولة.

□ عيوب نظام DOS

١. استخدام سطر الأوامر .

٢. مشاكل في بعض اللغات (خاصة التي تكتب من اليمين إلى اليسار)

٣. الواجهة البدائية و عدم استخدام واجهات رسومية .
٤. عدم القدرة على استخدام ملفات الوسائط المتعددة و ملفات الصوت و غيرها
٥. لا يدعم الكثير من الأجهزة و الملحقات مثل : الفأرة .
٦. عدم توفر الألوان والأشكال ثلاثية الأبعاد .
٧. لا يستطيع أن ينفذ أكثر من مهمة في آن واحد .
٨. لا يدعم تعددية المستخدمين.

٣-٢-٨-٢- نظام تشغيل النوافذ (Windows)

لقد تمت محاولات عديدة لتسهيل استخدام نظام التشغيل (DOS) منها المحاولات التي أضيفت بغرض استخدام تقنية الفأرة والرموز الصورية في تسهيل عمليات التشغيل وتنظيم عرض محتويات القرص باستخدام البرنامج (DOS SHELL) ، وقد نجحت هذه الجهود بإنتاج نظام تشغيل النوافذ (Windows) الذي أنتجته شركة (Microsoft) أيضاً ، وقد وفر نظام تشغيل النوافذ (Windows) التسهيلات التالية :

١- إمكانية التشغيل المتعدد للبرامج والتطبيقات ، أي إذا كنا نستخدم برنامج محرر النصوص وأردنا الذهاب إلى برنامج آخر في الجهاز ، فلا داعي لإغلاق محرر النصوص ، ويمكن عمل ذلك ، ومن ثم يمكن العودة ثانية إلى محرر النصوص ، ولم تكن هذه الخاصية متوفرة من قبل عند نظام التشغيل (DOS) .

٢- إمكانية استخدام اللغة العربية بأنماط وأشكال متعددة للحروف .

٣- كما توفر مع نظام التشغيل (DOS) العديد من البرامج والتطبيقات المساعدة في التشغيل ، فان نظام تشغيل النوافذ (Windows) وفر العديد

منها ويمكن طلب أي منها بوضع سهم الفأرة فوق الشكل المخصص لها والضغط مرتين متتبعيتين على المفتاح الأيسر للفأرة ، وهذا أسهل بكثير من كتابة أمر التطبيق أو البرنامج .

٤- أسهم نظام تشغيل النوافذ (Windows) في توسيع حجم الذاكرة الرئيسية ، وذلك من خلال التعامل مع القرص الصلب كذاكرة إضافية مساندة للذاكرة الرئيسية ، ولكن يعاب على ذلك بطأ التشغيل ، حيث يفضل أن تكون الذاكرة الرئيسية كبيرة قدر الإمكان .

٥- لم تقتصر استخدامات الفأرة على اختيار التطبيق المطلوب وتنفيذه بل ساعدت في تنسيق شاشة العرض ، وذلك بنقل أحد التطبيقات من مكان لآخر ، أو إضافة تطبيق ما لشاشة العرض أو إلغاء تطبيق لا يستخدم .

٦- ظهرت العديد من التطبيقات التي تعمل تحت هذا النظام ، كما أمكن طلب التطبيقات القديمة التي كانت تعمل تحت نظام التشغيل (DOS) فقط ، إلا أن خاصية البرمجة المتعددة لا تشملها في أغلب الأحيان .

لقد ظهر من هذا النظام عدة إصدارات من أهمها :

١- نظام تشغيل ويندوز (Windows) تحت نظام التشغيل (DOS) :

يعد هذا النظام من أنظمة التشغيل ذات أسلوب المواجهة الرسومية ، حيث يتيح استخدام تقنية الفأرة والرموز الصورية ، ويعمل هذا النظام تحت نظام التشغيل (DOS) ، وبكلام آخر إن هذا النظام لا يعد نظام تشغيل مستقل بل يجب توفر نظام التشغيل (DOS 5.0) حتى يمكن تنفيذ نظام ويندوز (Windows) . وقد ظهر من هذا النظام عدة إصدارات أهمها نظام ويندوز (Windows 3.1) و نظام ويندوز (Windows 3.11) .

٢. نظام تشغيل ويندوز (Windows) كنظام تشغيل متكامل

لقد جرى تطوير هذا النظام للاستغناء عن نظام التشغيل (DOS) ونظام التشغيل النوافذ (Windows 3.1) معاً مع إضافة بعض الميزات الإضافية ، وقد ظهر من هذا النظام عدة إصدارات من أهمها :

أ- نظام تشغيل ويندوز 95 (Windows 95)

إن نظام تشغيل ويندوز 95 نظام حديث الإصدار لتشغيل الأجهزة المستخدمة للمعالجات X86 أو Pentium والذي جرى تطويره للاستغناء عن نظام التشغيل (DOS) ونظام التشغيل ويندوز (Windows 3.11) معاً ، حيث كانت الوظيفة الأساسية لنظام التشغيل (DOS) هي التعرف على وحدات الحاسوب وتنظيم التعامل مع البرامج والتطبيقات ، فأصبح نظام تشغيل ويندوز 95 يقوم بهذه المهام بالإضافة إلى المهام التي يقوم بها نظام التشغيل ويندوز (Windows 3.11) أو نظام تشغيل (Windows 3.1) القديم ، ويمكن تلخيص المزايا التي أضافها هذا النظام بالميزات التالية :

◀ إمكانية تسمية الملفات والمستندات باللغة العربية ، ويصل طول الاسم إلى 255 حرفاً .

◀ يوفر النظام إمكانية البرمجة المتعددة المهام بشكل جيد ومتكامل .

◀ توفير نظام وقواعد الاتصال بين جهاز الحاسوب وأجهزة الحواسيب الأخرى المرتبطة به عبر شبكة من الحواسيب .

◀ يوفر النظام تسهيلات كثيرة عند إضافة تطبيقات أو أجهزة محيطية للحاسوب جديدة لأول مرة دون الحاجة إلى تعريفها وذلك باستخدام خاصية ركب وشغل ((Plug and Play (PnP)) ، والذي يقوم بالتعرف التلقائي عليها دون أي مجهود يذكر من مستخدم الحاسوب ، وأيضاً يسهل هذا البرنامج عمل تشغيل أجهزة الوسائط المتعددة والتي تعرض الصوت

والصورة معاً ، بالإضافة إلى ذلك يتيح هذا النظام السرعة في تشغيل البرامج والتطبيقات .

◀ إمكانية نقل الملفات بسهولة بين الحاسوب الشخصي المكتبي والحاسوب الشخصي المحمول عن طريق برنامج حقيبة الملفات .

◀ إمكانية تبادل الرسائل الكترونية والفاكسات وغيرها عن طريق برنامج Microsoft Exchange .

◀ أصبحت عملية إدارة الملفات أسهل وأفضل باستخدام برنامج المستكشف Explorer .

◀ أصبحت إمكانية تغيير شكل الشاشة والألوان والوقت والتاريخ واللغة تتم بسرعة وسهولة .

◀ أصبح هناك إمكانية للكشف عن الأخطاء ومحاولة تصحيحها قدر الإمكان .

◀ إمكانية عمل نظام التشغيل (DOS) من خلال نظام تشغيل ويندوز 95 ، وذلك لان نظام تشغيل ويندوز 95 نظام متكامل يتضمن كافة الإمكانيات التي كان يقوم به نظام التشغيل (DOS) ، ولكن مع استخدام أسلوب الواجهة الرسومية لكافة عمليات النظام .

ب- نظام تشغيل ويندوز 98 (Windows 98)

هذا النظام جاء كتحديث للإصدار السابق نظام تشغيل ويندوز 95 مع إضافة بعض الخصائص من أهمها :

◀ إضافة برنامج متصفح الانترنت Internet Explorer .

◀ إمكانية فتح الإيقونات بنقرة واحدة بدلاً من نقرتين .

◀ إضافة خاصية التحكم في المجلدات Folder Option .

- ◀ إضافة خاصية البحث عن الملفات والمجلدات والأشخاص وهي موجودة في قائمة ابدأ .
- ◀ إضافة خاصية تحديث نظام التشغيل تلقائياً عن طريق الانترنت .

ج- نظام تشغيل ويندوز Me (Windows Millennium)

- تم إطلاق هذا الإصدار عام 2000 ، وقد تم تطويره ليصبح متخصصاً للاستخدام المنزلي والشخصي ، بينما تم تطوير نظام تشغيل ويندوز 2000 من أجل الاستخدامات الإدارية والشبكات ، وهو عبارة عن إصدار جاء كتحديث لنظام تشغيل ويندوز 98 السابق مع إضافة بعض الخصائص من أهمها :
- ◀ إمكانية استعادة النظام System Restore في حالة وجود أي أخطاء أو مشاكل في نظام التشغيل .
 - ◀ حماية ملفات النظام System File Protection من الكتابة عليها أو تعديلها .
 - ◀ التحديث التلقائي Auto-update لنظام التشغيل عن طريق موقع شركة Microsoft على الانترنت .
 - ◀ إضافة برنامج قارئ الوسائط المتعددة Media Player 7 لإدارة الوسائط الرقمية : مثل ملفات الفيديو ، والصوت ، وغيرها .
 - ◀ إمكانية التعرف مباشرة على الأجهزة الموصلة على المنفذ الـ USB .
 - ◀ تطوير خاصية ركب وشغل (plug and Play) .

د- نظام تشغيل ويندوز XP (Windows XP)

- يعد نظام تشغيل ويندوز XP من أحدث نظم التشغيل ، ولا يختلف عن أنظمة التشغيل السابقة من حيث طريقة العمل ، غير أنه توفرت فيه بعض المزايا منها :

- ◀ سهولة نقل الملفات من حاسوب إلى آخر .
- ◀ إمكانية توفير حسابات لمستخدمي الحاسوب مع تحديد صلاحية كل منهم
- ◀ إضافة برنامج لإدارة المهام (Task Manager) لإغلاق البرمجيات التي لا تستجيب بدلاً من إعادة التشغيل كما في الإصدارات السابقة .
- ◀ يدعم هذا النظام جميع لغات العالم تقريباً بما فيها اللغة العربية دون الحاجة لشراء نسخة خاصة .
- ◀ القدرة على توفير الدعم لأي أجهزة حديثة أو إضافات مستقبلية .
- ◀ نظام الأمان فيه متطور ، ما جعله يتفوق على الإصدارات السابقة .
- ◀ أصبحت الواجهة الرسومية أفضل وأجمل .

□ فوائد نظام التشغيل ويندوز Windows

لهذا النظام فوائد كثيرة منها:

١. سهولة الاستخدام.
٢. واجهة استخدام رسومية تمكن من النفاذ إلى جميع فعاليات النظام بسهولة بواسطة شريط المهام وقسم الملفات .
٣. المستكشف الذي يساعدنا في إدارة الملفات والبحث عنها .
٤. إمكانية استخدام أسماء طويلة للملفات وباللغتين العربية والإنكليزية.
٥. إمكانية التشغيل المتعدد للبرامج والتطبيقات.
٦. إمكانية استخدام اللغة العربية ولغات أخرى كواجهة تطبيق لنظام التشغيل.
٧. توجد العديد من البرامج الملحقة والتطبيقات المساندة لهذا النظام مثل برامج الصيانة، وتفحص الأقراص، ... وغيرها واستخدامات أخرى للفأرة مثل تثبيت الإعدادات ونسخ الملفات وحذفها ومجالات أخرى كثيرة.

٨. دعم كامل للوسائط المتعددة Multi-Media .
٩. تعدد المهام الذي يسمح باستخدام أكثر من برنامج في آن واحد .
١٠. مبدل مايكروسوفت Microsoft Exchange الذي يسمح العديد من أنماط الاتصالات ومنها الفاكس والبريد الإلكتروني e-mail .

❑ عيوب نظام التشغيل ويندوز Windows

١. نظام النوافذ ليس من النظم مفتوحة المصدر أي أنها مرتبطة بشركة مايكروسوفت فإن توقفت الشركة لأي سبب من الأسباب سيتوقف النظام بمرور الزمن.
٢. إصدارات Windows مكلفة مادياً و قد يفوق سعر النسخة من نظام التشغيل سعر الحاسوب نفسه.
٣. نظام Windows ضعيف من الناحية الأمنية خاصة في مجال الشبكات.
٤. الفيروسات : تنتوع الفيروسات التي تصيب هذا النظام و يصل عددها إلى آلاف الفيروسات و عدد الفيروسات التي تصيب النظم الأخرى قد لا يصل إلى 100 فيروس .
٥. معظم الإصدارات من Windows التي تباع في دول العالم (والدول التي لا تهتم بحقوق الملكية) هي إصدارات مزورة (خضعت لعمليات تعديل و قرصنة و قد يتوقف النظام في حالة تحديث Windows من موقع الشركة الأم في حالة كشفها) .
٦. يلزم Windows بطء في الإقلاع .
٧. بعض إصداراته ضعيفة في مجال الشبكات .





٣-٢-٨-٣- نظام التشغيل أبل ماكنتوش (MAC OS) Mac

تعد شركة أبل (Apple) (ورمزها من اسمها التفاحة) أول من بدأت باستخدام الواجهة الرسومية بالنسبة للحواسيب الشخصية ، وذلك عندما قدمت حواسيب ماكنتوش Macintosh عام 1984 ، وقد تطور نظام التشغيل ماكنتوش ليقدم بالمزيد من التسهيلات في كل مرة ، كما أصبحت أجهزة ماكنتوش الأجهزة المفضلة في المكاتب التي غالبية أعمالها تحرير النصوص ومعالجة الملفات وذلك للأسباب التالية :

- ◀ سهولة التعامل مع النظام الذي لا يحتاج إلى كتابة الأوامر بل إلى وضع مؤشر الفأرة فوق التطبيق الذي يتكون من رسم بسيط واسمه .
- ◀ موائمة النظام للعديد من التطبيقات شائعة الاستخدام في مجالات كثيرة بمكاتب الأطباء والصحافة وفي بعض مجالات إدارة الأعمال .
- ◀ يتميز نظام تشغيل ماكنتوش بوجود تعريب متكامل للنظام منذ بدء إنتاجه ، وسهولة استخدامه لتطبيقات الكتابة ، والإخراج المميز للمستندات باللغة العربية .
- ◀ يسمح نظام تشغيل ماكنتوش بربط أكثر من جهاز معاً ، والاشتراك في آلات الطباعة عبر شبكة خاصة لأجهزة ماكنتوش يطلق عليها شبكة (Apple Talk) .
- ◀ يسمح نظام تشغيل ماكنتوش بتعدد المهام لمستخدم واحد .
- ◀ القدرة العالية للتعامل مع الصور والرسوم .
- ◀ سهولة إضافة أجهزة جديدة على الحاسوب وإضافة برمجيات حديثة إلى القرص الصلب .
- ومع سهولة ومزايا نظام التشغيل ماكنتوش إلا أن أجهزة هذا النظام تعد أقل انتشاراً من الأجهزة المتوافقة مع الحاسوب الشخصي من إنتاج شركة (IBM) ،

وذلك لخصوصية نظام تشغيل ماكنتوش ، حيث حرصت شركة أبل (Apple)
المنتجة له على وضعه فقط في الأجهزة التي تنتجها دون أجهزة الشركات الأخرى
، وبالتالي لا يستطيع مستخدم جهاز الماكنتوش تشغيل برمجياته على أجهزة
تستخدم نظام تشغيل ويندوز أو نظام التشغيل دوس أو تشغيل برمجياته على
أجهزة ماكنتوش .

إلا انه مع تطور نظام التشغيل ماكنتوش ، منذ ظهور رقم 7.5 مروراً
بالإصدار 8 والإصدار 9 والإصدار 10 ونسخته الحديثة المسماة (Panther)
والتي تعني النمر باللغة العربية ، أصبح بإمكان أجهزة الماكنتوش قراءة أقراص
الأجهزة المتوافقة مع أنظمة دوس وويندوز ، كما يمكن بعد إضافة برنامج خاص
على جهاز ماكنتوش محاكاة أنظمة تشغيل دوس والنوافذ ، وبالتالي تشغيل
برمجياتها على جهاز ماكنتوش ، بالإضافة إلى أن شركة (Apple) سمحت
بالترخيص لشركات أخرى باستخدام نظام تشغيل ماكنتوش ، مما وفر بالأسواق
عدداً من الأجهزة المتوافقة مع نظام أبل ماكنتوش .

٣-٢-٨-٤ - نظام تشغيل Chrome OS

هو نظام تشغيل خاص بشركة (غوغل Google) المالكة لمحرك البحث ، و
كروم هو نظام تشغيل مبني على نواة يونيكس و هو أيضاً نظام مفتوح المصدر
وطرح مجاناً على الإنترنت في النصف الثاني من عام 2010 م .
و تزعم Google أن النظام الجديد منافس حقيقي لنظام النوافذ و أن النظام
الجديد يعد صفقة قوية لعملاقة البرمجيات مايكروسوفت ، و المثير في الأمر أن
شركة IBM الحليف لمايكروسوفت و بعض الشركات الأخرى ساندت (غوغل)
في مشروع نظام التشغيل الجديد، و قد ذكر بعض الخبراء في مجال نظم التشغيل
أن الفترة الزمنية التي قررتها شركة Google غوغل لا تكفي لإنتاج نظام تشغيل

مواكب لأنظمة النوافذ هذا الشيء الذي جعل الكثيرون يشككون في مدى مصداقية غوغل (Google) ، فإما أن غوغل (Google) قامت بشراء نظام تشغيل و تقوم بإجراء بعض التعديلات عليه أو أنها بدأت العمل في نظامها كروم منذ فترة زمنية ليست بالقصيرة .



٣-٢-٨-٥ - نظام التشغيل (UNIX)

تم إنتاج هذا النظام في معامل الهاتف للشركة الأمريكية AT&T عام 1969 لاستخدامه في تشغيل مخدمات الحواسيب ، ويعد نظام التشغيل UNIX أحد أنظمة التشغيل متعددة المستخدمين والتي تعتمد بالأساس على المواجهة الخطية ، أي أن المستخدم يقوم بكتابة الأوامر والتعليمات بصورة خطية ، و بدأ استخدام نظام التشغيل UNIX عام 1969 على الأجهزة الكبيرة Mainframe والمتوسطة (Minicomputer) وتم تعديله أخيراً للعمل على الحواسيب الشخصية (PC Computer) .

وهو يُعد من أقوى نظم التشغيل حيث أنه يعتمد في تصميمه على تجزئة البرامج إلى برامج صغيرة منفصلة (Modules) ما يسهل فصل أي برنامج فرعي أو إضافة برنامج جديد ، ويتميز هذا النظام عن أنظمة التشغيل الأخرى بالعديد من المزايا منها :

- ◀ يمكن استخدامه مع جميع أنواع الحواسيب الكبيرة الحجم والأصغر منها ، وهذا يعني سهولة كتابة وتشغيل التطبيقات والبرامج التي تعمل على الحواسيب الصغيرة مثلاً لكي تعمل على الحواسيب المتوسطة أو الكبيرة .
- ◀ كُتِبَ نظام التشغيل UNIX بلغة C بدلاً من لغة التجميع الصعبة ، مما ساعد على تسهيل العمل على تطويره ، حيث تتميز هذه اللغة بدقتها وقوتها في التشغيل .

◀ يعد نظام التشغيل UNIX أول نظام يقدم للحواسيب الصغيرة إمكانية البرمجة المتعددة المهام .

◀ توفر نظام أمني لحماية المستندات لمنع الآخرين من الاطلاع عليها .

وقد أصبح نظام التشغيل UNIX شائع الاستخدام لأن معظم البرمجيات التي تتعامل مع الشبكات وتديرها تعتمد على هذا النظام مثل بروتوكول التحكم بالإرسال TCP/IP المستخدم مع شبكات الانترنت ، ويتميز نظام التشغيل UNIX بسهولة استخدامه إلا أن أوامره يصعب تذكرها وخاصة إذا لم يستخدم المستخدم الحاسوب كثيراً ، ما يجعل هذا النظام غير مفضل عند الكثيرين من المستخدمين في الوقت الذي يفضله المهنيين في مجال الحاسوب ، وللتغلب على هذا الاعتراض توفر الكثير من الشركات نافذة استعلام لكي يتذكر المستخدم أوامر النظام . وقد صدرت عدة إصدارات لهذا النظام تعمل مع جميع أنواع الحواسيب على اختلاف أحجامها وأنواعها ، وتعمل على أنواع مختلفة من المعالجات ، كما يتوفر لهذا النظام أسلوبان للتعامل : أسلوب المواجهة الخطية وهو الشائع ، وأسلوب المواجهة الرسومية ، وهناك هناك أنظمة كثيرة نتجت من نظام التشغيل UNIX من أبرزها نظام التشغيل Linux .

وبعد المختصون بنظام التشغيل UNIX ، أن نظام التشغيل يتألف من ثلاثة أجزاء رئيسية، هي : النواة (Kernel) ، والواجهة (الغلاف) (Shell) ، ونظام الملفات (File system) .

◀ النواة (Kernel)

هي مجموعة الوظائف الأدنى مستوى في نظام التشغيل ، التي تُحمَل إلى الذاكرة ، كلما قمنا بإقلاع الجهاز ، وذلك مباشرة بعد أن تعمل بعض الوظائف الموجودة في الـ BIOS ، وتتولى الوظائف الموجودة في نظم التشغيل (DOS ، win NT ، win200 ، win95) معالجة عمليات نقل المعطيات بين مختلف

أجزاء نظام الحاسوب ، كعملية نقل المعطيات من القرص الصلب إلى الذاكرة RAM ثم إلى المعالج ، أو تحديد قطاعات من الذاكرة لمختلف عمليات النظام التي تحدث دائماً مادام الجهاز ينفذ عملاً معيناً ، أما برمجيات النواة Kernel مسؤولة أيضاً عن جدولة عمليات المعالج لتسمح لوظائف الواجهة بالوصول إليه ولها عدة مهام أساسية منها :

- ١- إدارة الأجهزة المتصلة بالحاسوب Devices وذلك بالاستعانة بتعريفات السواقات Drivers .
- ٢- إدارة نظم الملفات file systems management .
- ٣- إدارة و تنظيم العمليات التي ينفذها المعالج .
- ٤- إدارة وحدات الذاكرة المختلفة .
- ٥- تنفيذ نماذج الأمن التي يقوم مدير النظام بتحديد ها .



◀ واجهة مستخدم يونكس

Shell أو Command Line Interface

هي واجهة المستخدم الأساسية للنظام و هي تعمل بواسطة الأوامر النصية Commands وهي تشبه في نظم MS Windows واجهه الأوامر Command Prompt التي تعمل بنظام DOS و لكنها أقوى من نظام DOS و أكثر قابلية للتطوير، و في حالة دخول المستخدم إلي النظام باستخدام الواجهة النصية Single User Mode و هي الحالة الأساسية لنظام UNIX تكون Shell وهي وسيلة الاستخدام المتاحة ، و تُعد من أهم الوسائل في إصلاح أعطال النظام و لذا يعد تعلم الأوامر النصية الخاصة بنظام UNIX من الأشياء الأساسية سواء للمستخدم العادي User أو لمدير النظام Administrator

الواجهة هي عبارة عن برنامج يعمل في الطبقة العليا منه ، ويسمح للمستخدمين بإصدار الأوامر إليه ، ويوجد لنظام UNIX عدد من الواجهات ،
مثل واجهات : Bourne , Korn , Bourne Again
توفر أنظمة التشغيل نوعاً من العزل بين طبقتي التطبيقات والعتاد وذلك
عبر ما يسمى واجهة برمجة التطبيقات API (Application Programming Interface) التي توفر للمبرمجين أسلوباً قياسيلاً لاستدعاء
إجراءات عامة محددة الوظائف أو إجراءات تستدعي برامج جزئية تحكمية بعتاد
معين ، ويتم تحديث وظائف واجهة برمجة التطبيقات API كلما توفر عتاد جديد
، لكن هذا ليس كافياً للاتصال بالعتاد بشكل صحيح ، إذ يستخدم نظام التشغيل
برامج قيادة Drivers كتبت خصيصاً لتأمين الاتصال بين نظام التشغيل وقطعة
معينة من العتاد ، كبطاقة الرسومات على سبيل المثال .



❏ ميزات نظام التشغيل UNIX

1. خاصية تعدد المهام (Multitasking) : عندما يستخدم نظام ما نظام UNIX كنظام تشغيل فإن هذا النظام يتمكن من تشغيل أكثر من تطبيق واحد في نفس الوقت .
2. البناء الهندسي لـ 32 بت : يستفيد نظام UNIX من (32 bit) في تمرير المعطيات ومعالجتها ، ما يوفر سرعة أداء أكبر وقدرة على استخدام برامج ذات حجم أكبر أيضاً .
3. بروتوكولات الشبكة : تُعد بروتوكولات نظام (UNIX) من أشهر البروتوكولات التي تستخدم على مستوى العالم في مجال تطبيقات الحاسوب ومن أهم هذه البروتوكولات :

FTP : File Transfer Protocol .

TCP/IP : Transmission Protocol /Internet Protocol .

ويعتبر بروتوكول (TCP/IP) البروتوكول الرئيسي الذي تعمل به شبكة الإنترنت .



٣-٢-٨-٦- نظام التشغيل Linux

يُعرف نظام التشغيل لينوكس بأنه نظام تشغيل مجاني مفتوح المصدر (Open Source) يوفر كافة مزايا أنظمة التشغيل من تعدد المستخدمين Multi-user، وتعدد المهام Multitask، وذاكرة افتراضية Virtual Memory، وإدارة متطورة للذاكرة، والمشاركة بالمكتبات البرمجية Shared libraries، بالإضافة لدعم خاص للشبكات و البروتوكول TCP/IP.

وهو نظام تشغيل يصنف ضمن عائلة Unix تم تصميمه من قبل طالب فنلندي يدعى ليونس تورفالدز في جامعة هلسنكي عام 1990 م ، متأثراً بنسخة صغيرة من نظام التشغيل UNIX والتي كانت قد صممت لأغراض تعليمية والتي سميت minix ، وفي عام 1991 عرض "لينس" نظامه على مجموعة من المبرمجين و المطورين على إحدى مجموعات الـ newsgroups ، و كان اهتمام هؤلاء المطورين سبب لتطوير نظام التشغيل لينكس في نسخته الأولى .

سجل لينس نظامه باسم لينكس Linux و كانت نوعية الترخيص عامة GPL (General Public License) والتي تتيح استخدام النظام و توزيع نسخه مجانيا و التعديل عليه بحرية عالية ، ولقد حتمت رخصة البرمجيات من نوع GPL توفير الشيفرة المصدرية لهذه البرمجيات مجانياً أيضاً ، و لهذا فإن نظام التشغيل لينكس كاملاً متوفرٌ على شكل شيفرة مصدرية مكتوبة بلغة C++/C ، و هذا يتيح التعديل على النظام ليلاءم الاحتياجات والاستخدامات المختلفة ، و بسبب سياسة المصدر المفتوح التي أتبعها نظام التشغيل لينكس أهتم عشرات

الآلاف من المبرمجين، و قامت العديد من الشركات بتطوير نواة النظام و تطوير واجهته ليصبح على ما هو عليه الآن.

و زاد الاهتمام بنظام التشغيل Linux عندما اهتمت به شركات كبرى مثل IBM و Sun و Corel و DEC و DELL و SGI و غيرها الكثير ، و يرجع سبب الاهتمام إلى أن نظام التشغيل لينكس وفر ما لم يتوفر من قبل و لهذا بدأ يظهر كمنافس لنظام مايكروسوفت ويندوز (Windows) الأكثر انتشاراً في العالم .

على الرغم من اختلاف وتعدد إصدارات Linux و تنوعها ويمتاز Linux بأنه يدعم أكبر عدد من أنواع العتاد المختلفة وهذا ما جعله ينافس أكبر الأنظمة .
إن نظام التشغيل لينكس Linux هو نظام لإدارة الشبكات في الأساس مثل نظام التشغيل UNIX ، فإن نظام لينكس مبني في الأساس للعمل على الشبكات مثل الإنترنت، وقد وفر فيه المطورين مزايا عديدة للشبكات أهمها الأمن و الحماية، ويحظى نظام التشغيل لينوكس Linux بشعبية كبيرة لما يوفره من ميزات في حقل الأنظمة متعددة الأغراض ، باعتباره من البرمجيات مفتوحة المصدر ، ولهذا بدأ يظهر كمنافس لنظام مايكروسوفت Windows NT .

□ مميزات نظام التشغيل Linux

هناك الكثير من الأسباب لاستعمال نظام لينوكس كنظام تشغيل في الكثير من التطبيقات بدءاً من عناصر الشبكة إلى الخادمت الكبيرة "back-end"، وأهم ميزات نظام لينوكس هي :

١ - يُعد نظام التشغيل لينوكس Linux متعدد المستخدمين Multiuser

إن المعنى الحقيقي لنظام التشغيل متعدد المستخدمين الـ Multiuser أنه يسمح للعديد من المستخدمين بالدخول (log in) والعمل بتشغيل برامجهم في الوقت نفسه، ونظام التشغيل الذي يتصف بذلك يجب أن تكون لديه القدرة على

التمييز بين المستخدمين ليوفر الأمان والفصل بينهم ، ومن أمثلة هذه الأنظمة Linux, FreeBSD, SunOS، والكثير من أنظمة التشغيل المماثلة لـ Unix.

٢ - يُعد نظام التشغيل لينوكس Linux متعدد المهام Multi tasking
تعتبر خاصية تعدد المهام عامل مشترك بين أغلب أنظمة التشغيل الحديثة، وهذا يسمح للنظام بتشغيل أكثر من برنامج/ مهمة في الوقت نفسه ، وكانت أنظمة التشغيل القديمة مثل الـ MS-DOS لم تكن لتسمح بأكثر من تطبيق واحد في الوقت نفسه، بينما أغلب أنظمة التشغيل الحالية تنفذ خاصية تعدد المهام Multitasking ، فإنهم لا يدعمون ذلك بطريقة واحدة، ولكن الأداء والثبات يعاني أحيانا من سوء تطبيق خاصية تعدد المهام . Multitasking أو من المتطلبات الموضوعة في التصميم ، أما نظام التشغيل لينوكس Linux فله نظام جيد لخاصية تعدد المهام Multitasking والذي يوصف بالثبات والسرعة معاً.

٣ - يمتاز نظام التشغيل لينوكس Linux بالثبات
يتميز نظام التشغيل لينوكس Linux بأن لديه ثبات عالي المستوي وذلك نتيجة لتصميمه وتجزئته ليعمل كمجموعة من المكونات (Modularity) ، وأن فشل إحدى التطبيقات لا يعني بالضرورة فشل النظام ككل وأن يصبح النظام غير متزن، كما أن إضافة برامج أو تعديل المواصفات لا يتطلب إعادة تشغيل النظام وهذا يعني بالضرورة زيادة الوقت الذي يظل فيه النظام يؤدي خدماته للمستخدمين "uptime"، ويجعل الإشراف على النظام أكثر سهولة خاصة إذا كنا نتحدث عن مخدم Server.

٤- ممتلك نظام التشغيل لينوكس Linux مكتبة هائلة من البرامج في أغلب إصدارات نظام تشغيل لينوكس الموجودة حالياً، غالباً ما تحصل على نظام التشغيل بالإضافة لحوالي 6 أقراص مدمجة ممثلة بالبرامج، أيضاً البرامج متاحة بالمجان مع مصدر الكود الذي يشمل الكثير من البرامج التجارية المعروفة، وأغلب مخدمات الإنترنت التي تعمل على نظام تشغيل لينوكس تستعمل برنامج مجاني متوافق مع نظام التشغيل لينوكس الـ apache, BIND, SendMail.

٥- نظام التشغيل Linux مدعوم بنطاق واسع من الأنظمة والأجهزة القديمة يعمل نظام التشغيل لينوكس Linux على الكثير من الأنظمة والأجهزة القديمة التي خرجت من الخدمة في عرف أنظمة التشغيل الأخرى منذ فترة طويلة، فبينما حاسوب قديم معالجه من طراز 80486 لا يمكن أن يعمل عليه نظام تشغيل windows 98 (ناهيك عن windows XP)، فإن نظام التشغيل لينوكس Linux يستطيع أن يحول هذا الجهاز إلى مخدم في شبكة أو جدار ناري firewall، أو موجه Router . إذا كان لدينا أي من الأجهزة القديمة والتي لا تُعد حواسيب شخصية ولا يقوم منتجوها بدعمها حالياً، قد تجد دعماً ممتازاً في أوساط نظام التشغيل لينوكس Linux، وهذا يمكن أن يعيد الحياة إلى هذه الأجهزة ويجعلها مفيدة مرة أخرى.

٦- إن نظام التشغيل لينوكس Linux سريع في أغلب الأحوال إن نظام التشغيل لينوكس Linux سريع، والكثير من المقارنات (benchmarks) تظهر تفوق نظام التشغيل لينوكس Linux على الكثير من الأنظمة التقليدية كمخدم للشبكة أو الإنترنت، ويعود السبب في هذا إلى أنه بمجرد اكتشاف سبب الاختناق في الأداء (bottleneck)، فإنها أولاً يتم الإعلان عنها وتخضع للمعالجة بواسطة مجتمع نظام التشغيل لينوكس Linux

الواسع من دون انتظار لكي يقوم بإصلاحها مطور نظام التشغيل كما هو الحال مع الكثير من أنظمة التشغيل.

٧- من أهم ما يميز نظام تشغيل Linux أنه يعمل على عدد كبير من أنواع الأجهزة والمعالجات ، فهو يعمل على معالجات x86 و i64 من شركة Intel و SPARC من شركة Sun ، ومعالجات PowerPC و Motorola و MIPS و RISC و غيرها ، وهذا التنوع يغطي إمكانية الربط والانسجام لكافة الأنواع المختلفة للعمل على شبكة واحدة .

٨- يوفر نظام التشغيل Linux انفتاحاً على أنظمة أخرى مثل نظام Windows NT أو أي نظام تشغيل آخر من نوع UNIX ، مما يعطي القدرة على أن يكون نظام وسطي بين عدة أنظمة تشغيل في شبكة واحدة .

٩- يمكن في نظام تشغيل Linux تشغيل البرامج المكتوبة في نظام التشغيل UNIX لأنه في الأصل يعمل بنفس الطريقة ، وهناك برامج مثل WINE تتيح تشغيل برامج وألعاب لنظام التشغيل Windows للعمل في بيئة نظام التشغيل Linux .

١٠- يوفر استخدام نظام التشغيل Linux في الحواسيب الشخصية دعم واسع للإضافات والبطاقات Hardware من بطاقة صوت إلى بطاقة عرض فيديو وبطاقات دعم ثلاثية الأبعاد وبطاقات الـ FM ، ومنافذ الـ USB ، والمساحات والطابعات وغير ذلك من الإضافات المختلفة .



✕ أسباب تميز نظام Linux عن نظام التشغيل Windows

١. نظام تشغيل Linux مجاني يقوم بكل ما يقوم به نظام تشغيل Windows بثبات وأمان وبأقل تكلفة وأفضل أداء وبسعر مجاني ، وهذا العرض من الصعب مقاومته .

٢. الأمان المتوفر في نظام تشغيل Linux لأن بيئة Windows هي بيئة خصبة لانتشار وعمل الفيروسات ، بينما بيئة نظام Linux غير صالحة لحياة هذه الفيروسات وانتشارها . فعدد الفيروسات الموجودة في نظام تشغيل Linux لا يتجاوز العشرات وهي على الغالب ليست ذات طابع تخريبي ، بينما في بيئة نظام تشغيل Windows ملا يقل عن 58000 نوع من الفيروسات وذات طابع تخريبي ، ويتم كتابة ونشر 80 فيروس يومياً حول العالم .

٣. يعمل نظام تشغيل Linux بثبات ودون توقف لأي سبب كان وحفاظه على موارده ، بينما في نظام التشغيل Windows موارد الجهاز من الذاكرة يتم تسريبها واستنزافها وبالتالي بحاجة إلى إعادة تشغيل الجهاز لاستعادة الموارد ، بينما نظام Linux لا حاجة لإعادة تشغيل الجهاز لاستعادة الموارد حتى لو بقي الجهاز مشغولاً لمدة عام كامل .

٤. السياسات الاحتكارية لشركة Microsoft ، فاتجه عدد كبير من مستخدمي Windows لأنظمة أخرى مثل نظام Linux لأنه يوفر حرية الاختيار للمستخدم ، فهو يختار واجهة الاستخدام التي يريدتها ويحدد شكلها وملامحها ، وغير ذلك من حرية الاختيار المتوفرة .



٣-٢-٨-٧- أنظمة تشغيل الشبكات

يمكن عن طريق ربط أجهزة الحاسوب معاً إيجاد ما يطلق عليه شبكة الحواسيب ، ولقد ازدادت أهمية الشبكات في السنوات الأخيرة . ومن أهم العوامل التي ساعدت على ذلك :

- التطور السريع في تقنية الحواسيب الشخصية .
- التطور في تقنية الاتصالات وخطوط الهاتف .
- تطور قواعد المعطيات وتزايد الحاجة إليها .

فالأنظمة التي تقوم بتشغيل شبكات الحواسيب لا تختلف عن أنظمة التشغيل الأخرى إلا أنها مهياة بحيث تسمح لأجهزة الحاسوب المتصلة بالشبكة بتبادل المعلومات فيما بينها والاستفادة من الموارد المتوفرة مثل الطابعة أو الأقراص المدمجة ، بالإضافة إلى أن لها قدرات أمنية لمنع التسلل أو العبث بالأجهزة أو الاطلاع على المستندات أو الوثائق المخزنة على الحاسوب ، ويمكن لجميع أجهزة المتصلين بالشبكة والتي تعمل بنظام التشغيل الشبكي أن يستفيدوا من الخدمات التالية :

- استخدام البريد الالكتروني (E - mail) فيما بينهم .
- الاستفادة من وسائط التخزين الأخرى الموجودة على الشبكة سواء كان ذلك بنسخ البرامج المخزنة عليها إلى الأقراص الصلبة لحواسيبهم ، أو بتحميل البرامج والتطبيقات لتشغيلها .
- المشاركة في استخدام الطابعات .
- إمكانية تبادل البرامج والملفات بين المشتركين في الشبكة .

وهناك فوائد كثيرة أخرى للشبكات سوف ندرسها في مقرر شبكات الحواسيب وتراسل المعطيات للسنة الرابعة . ويختلف نظام التشغيل الشبكي عن نظام تشغيل

الحاسوب بأنه يسمح بتوزيع الملفات أو البرامج والمعطيات بين عدد من الأجهزة الموجودة على الشبكة ، وبالتالي عند تنفيذ هذا البرنامج يقوم نظام التشغيل بإدارتها معاً ، بينما في أنظمة الحواسيب يتم الاحتفاظ بنسخة الملف أو البرنامج كاملة على جهاز واحد دون توزيع محتوياته بين الأجهزة المتصلة بالشبكة وإنما يمكن نقله من جهاز لآخر فقط ، بعض الأمثلة على أنظمة تشغيل الشبكات :

✓ نظام النوافذ إن تي (windows NT) من إنتاج مايكروسوفت .

✓ windows 2000 و windows 2003

✓ نظام Net-ware من إنتاج شركة نوفل Novell .

✓ نظام UNIX

✓ نظام Linux

=====



الفصل الرابع
الشبكات الحاسوبية والانترنت



الشبكات الحاسوبية والانترنت

٤-١- مقدمة Introduction

أصبح العالم اليوم رغم ترامي أطرافه قرية صغيرة، إذ يستطيع الأفراد القيام بعمليات اتصال فورية وتبادل المعلومات عبر الحاسوب مع أي جهة في أي بقعة على الأرض ، وقد اسهم اندماج تكنولوجيا المعلومات والاتصالات كثيراً في التقريب ما بين الأفراد في شتى المجالات ، وأصبحت الشبكات جزءاً أساسياً في حياتنا الشخصية والمهنية فيمكنك اليوم أن ترسل رسالة مكونة من عدة صفحات وصور وأصوات ورسومات متحركة إلى مجموعة أشخاص في أي مكان دفعة واحدة وفي دقائق معدودة ، وأصبح من الممكن أن تتصل من حاسوب منزلك أو مقر عملك ببنوك المعلومات والشركات والمكتبات العالمية للحصول على المعلومات التي تهتمك ، كما أصبح متاحاً الآن عقد مؤتمرات دولية وندوات تفاعلية لأطراف متباعدة عبر شبكة الانترنت، وهناك أيضاً التعليم عن بعد والطب عن بعد والتجارة الإلكترونية والحكومة الإلكترونية كل ذلك لم يكن ممكناً بدون الاندماج بين تكنولوجيا الاتصالات وتكنولوجيا الحواسيب وإيجاد ما يسمى بشبكة الحاسوب .

ازداد استخدام الشبكات الحاسوبية خلال السنوات القليلة الماضية ، حيث أصبح من النادر وجود حاسوب في شركة أو مؤسسة غير متصل بشبكة حواسيب ويعود السبب في ذلك إلى ما وجدته هذه الشركات من فوائد تعود عليها من وجود هذه الشبكات والمشاركة في الأجهزة كالطابعات والراسمات وغيرها، وكذلك المشاركة في المعلومات التي تُعد العنصر الأهم لأي شركة.

٢-٤ - تعريف الشبكة الحاسوبية

هي مجموعة من الحواسيب المرتبطة معاً بطريقة معينة لإرسال وتبادل المعلومات أو الوصول إلى معطيات محددة وترتبط بخطوط اتصال بحيث يمكن لمستخدميها المشاركة في الموارد المتاحة .

- يمكن أن تكون مجموعة من الحواسيب الشخصية مرتبطة معاً ، أو حاسوباً كبيراً ترتبط به طرفيات (Terminals) .
- خطوط الاتصال التي تربط عناصر النظام ببعضها ، قد تكون سلكية أو لاسلكية ، وتحدد طريقة الربط شكل الشبكة وبنيتها .
- الموارد المتاحة : ويقصد بها المعدات والبرامج والمعلومات .

٣-٤ - أنواع الشبكات الحاسوبية

يمكن تصنيف شبكات الحواسيب إلى عدة أنواع ولكن يمكن تصنيفها حسب التوزيع الجغرافي إلى ما يلي :

١- الشبكات المحلية LAN

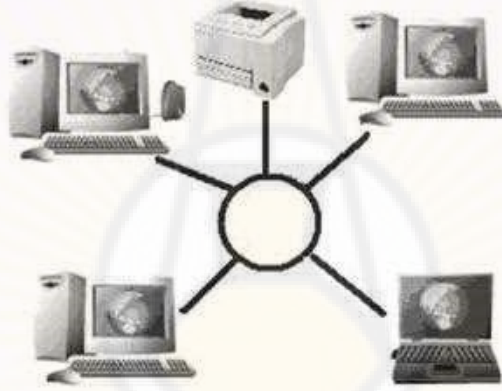
٢- شبكات المدينة MAN

٣- الشبكات الواسعة WAN

١-٣-٤ - الشبكات المحلية LAN (Local Area Networks)

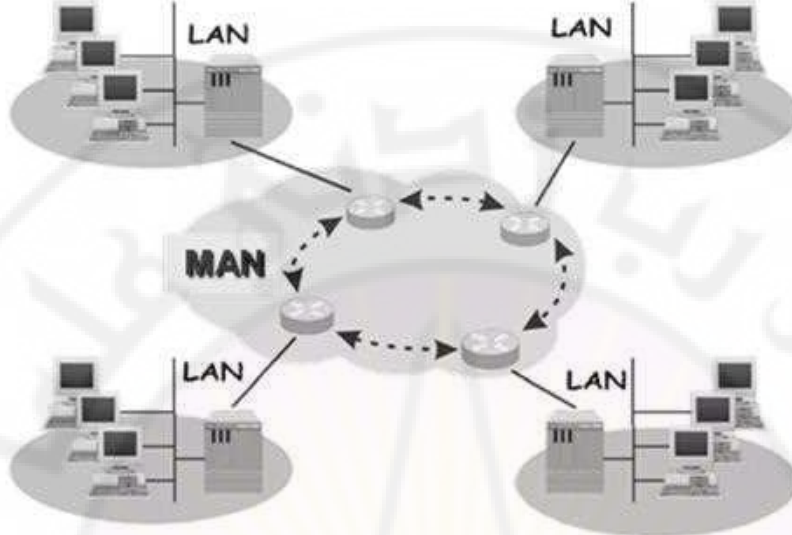
وهي الشبكات التي تربط مجموعة من الحواسيب المتواجدة في مكان واحد (معمل ، بناء) ، وهي أبسط أنواع شبكات الحواسيب و يمكن أن تحتوي على مئات الحواسيب المتصلة مع بعضها ضمن بناء أو أبنية متجاورة .

حيث يتم هذا الاتصال عن طريق وصلات سلكية مباشرة أو لا سلكية، وتستخدم هذه الشبكات في الشركات الصغيرة، المدارس، أو في المنازل وغيرها، ويجب أن لا تتجاوز المسافة العظمى الفاصلة بينهما 2.5 Km . والحاسوب المتصل بهذه الشبكة بإمكانه الوصول إلى الموارد الأخرى من المعلومات المتواجدة على أي حاسوب آخر كالبرامج و الملفات ، كما يمكن أن تتشارك هذه الحواسيب مع الأجهزة الملحقة مثل الطابعات و الراسمات وأجهزة الفاكس و غيرها ، ويبين الشكل (٤-١) شبكة LAN .



الشكل (٤-١) يوضح شبكة LAN

٤-٣-٢- شبكات المدينة (Metropolitan Area Networks) MAN
وهي الشبكات التي تربط عدد من الشبكات المحلية (LANs) المتواجدة في مدينة واحدة، وهي تعمل بسرعات عالية و تستخدم عادة كابلات الألياف الضوئية (Fiber Optic) و تغطي مساحة واسعة تتراوح بين 20Km إلى 100Km ، ويبين الشكل (٤-٢) شبكة MAN .



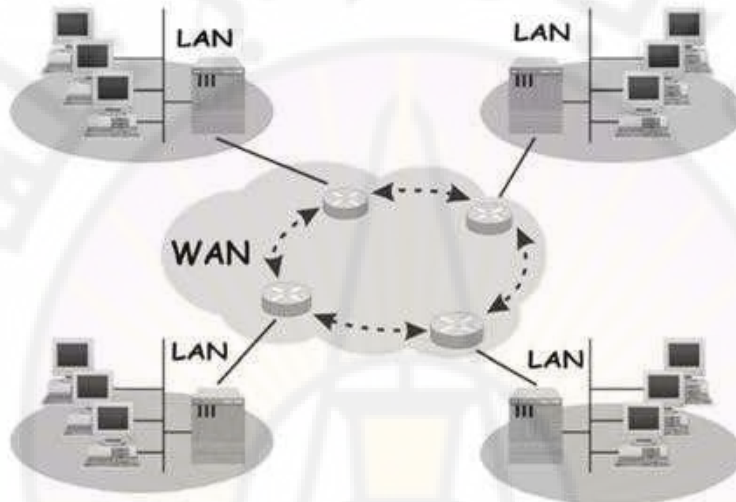
الشكل (٢-٤) - يوضح شبكة MAN

٤-٣-٣- الشبكات الواسعة (Wide Area Networks) WAN

في بداية ظهور شبكات الحواسيب ، لم تستطع الشبكات المحلية (LANs) دعم احتياجات الشبكات للشركات الكبيرة التي تتوزع مكاتبها على مساحات واسعة، في عدة دول، لهذا كان لابد من تطوير نوع جديد من الشبكات يقوم بربط الشبكات المحلية المتواجدة في أنحاء مختلفة من الدولة أو عدة دول ، وأطلق على هذا النوع من الشبكات اسم الشبكات الواسعة (WAN) و باستخدام هذا النوع تزايد عدد المستخدمين بشبكة الحواسيب في الشركات الكبيرة إلى آلاف الأشخاص.

وفي السنوات الأخيرة تلاشى دور شبكات المدينة (MAN) تقريباً وانقسمت الشبكات إلى شبكات محلية (LAN) أو واسعة (WAN)، ويبين الشكل (٣-٤) الشبكة WAN .

وتنظم عملية الاتصال بين هذه الشبكات بطرق اتصالات متعددة ، وذلك باستخدام خطوط الهاتف أو باستخدام الأقمار الصناعية ومثال على ذلك نظام حجز الطيران الدولي والشبكة الدولية (الانترنت) .



الشكل (٣-٤) - يوضح الشبكة WAN

٤-٤ - فوائد الشبكات الحاسوبية

إن الهدف الرئيسي لإنشاء شبكات الحواسيب هو الفوائد الكثيرة التي سيتم

جنيها ، ونذكر منها :

١. تبادل المعلومات :

إن من أهم الأسباب لبناء الشبكات تبادل المعلومات و الملفات بسهولة و

بسرعة عالية.

٢. المشاركة في استخدام موارد الشبكة

إن وجود الشبكة يؤمن التشارك في استخدام التجهيزات المادية المرتبطة بالشبكة (الطابعات ، الراسمات ، الماسحات الضوئية ، الخ).

٣. التشارك في البرمجيات

تؤمن شبكة الحواسيب للمستخدمين إمكانية التشارك في البرمجيات كقواعد المعطيات التي يمكن أن توضع على المخدمات (Servers) و توفر الشبكة إمكانية استخدامها من قبل كل أقسام الشركة.

٤. الإدارة المركزية

عندما يكون نظام تشغيل وإدارة الشبكات من نوع المخدم/ العميل (Client-Server) يتم إدارة الشبكة إدارة مركزية أي أن هناك مدير للشبكة (Administrator) هو الذي يستطيع إدارة و مراقبة الشبكات و التحكم بها من مكان مركزي ، وبالتالي إمكانية إدارتها بشكل جيد ورفع مستوى أداء العمل في الشبكة.

٥. تأمين التوافق بين التجهيزات و البرمجيات المختلفة

إن شبكة الحواسيب تسمح بربط التجهيزات المختلفة و الحواسيب ذات أنظمة التشغيل المختلفة بعضها ببعض ، والتخاطب فيما بينها.

٦. أمن وحماية المعلومات

يُعد أمن و حماية المعلومات من الأشياء المهمة في الشركات ، لذلك يجب حماية المعلومات من الأشخاص الدخلاء غير المسموح لهم بالوصول إليها وذلك عن طريق منح السماحيات و أذونات الوصول الملائمة لكل مستخدميه.

٧. تأمين الاتصال السهل

تؤمن شبكات الحواسيب سهولة إجراء الاتصالات وذلك من خلال توفير خدمة البريد الإلكتروني وكذلك يمكن أن توفر بعض شبكات الحواسيب خدمة المؤتمرات

الفيديوية والتي تتيح تشارك مجموعة من المستخدمين معاً في جلسة نقاش واحدة.

٤-٥ - نظم تشغيل وإدارة الشبكات

لكي تستطيع شبكات الحواسيب توفير الفوائد و الخدمات التي ذكرت سابقاً، لابد من توفر نظام تشغيل وإدارة للشبكات قادر على تحقيق ذلك ، لذلك يمكن تصنيف شبكات الحواسيب حسب نظام التشغيل وإدارة الشبكات إلى نوعين رئيسيين :

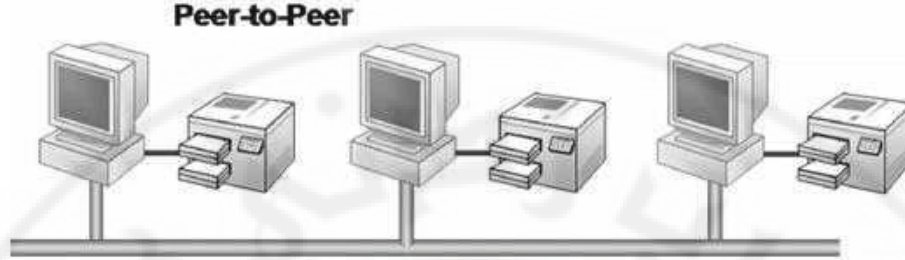
١- شبكات الند للند Peer-to-Peer.

٢- شبكات المخدم/العميل Client / Server.

٤-٥-١ - شبكات الند للند (Peer-to-Peer)

هي الشبكات التي تتكون من مجموعة من الحواسيب المتصلة معاً ولها حقوق متساوية و كل حاسوب بها يمكن أن يكون مخدم و عميل في نفس الوقت ، أي أن كل حاسوب على الشبكة يمكن أن يزود غيره من الحواسيب بالمعلومات و في الوقت نفسه ، ويمكن أن يأخذ المعلومات من غيره من الحواسيب المتصلة بالشبكة، وتعد أن كل حاسوب في الشبكة له إدارة مستقلة لموارده بحيث يقوم المسؤول عنه بتوزيع صلاحيات الدخول إلى حاسوبه من قبل الآخرين وتحديد نوع الخدمات التي يمكن أن يوفرها لغيره، ويبين الشكل (٤-٤) شبكات الند للند .

لا يحتوي هذا النوع من الشبكات على مخدم (Server) بل كل حاسوب في الشبكة يمكن أن يكون مخدماً وعميلاً في الوقت نفسه ، أي لا توجد إدارة مركزية أو حاسوب مركزي يتم تخزين أغلب التطبيقات المشتركة عليه .



الشكل (٤-٤) - يبين شبكات الند للند

ويطلق على هذا النوع من الشبكات اسم آخر هو مجموعة العمل (Workgroup) وذلك لأنها تحتوي مجموعة من الحواسيب التي تتعاون فيما بينها لإنجاز عمل معين.

ويعد هذا النوع من الشبكات مناسباً :

١. عندما لا يتجاوز عدد الحواسيب 10 أجهزة.
٢. عندما لا توجد إدارة مركزية.
٣. عندما لا يكون أمن الشبكات ذا أهمية وذلك لأن كل مستخدم في الشبكة يقوم بوضع نظام الحماية المناسب له.
٤. عندما يكون الحاجة إلى تكلفة محدودة .
٥. عندما لا نحتاج إلى برامج إضافية على نظام التشغيل .
٦. عندما يكون الموقع سهل التركيب والصيانة فيه.
٧. عندما لا نحتاج إلى حواسيب ذات مواصفات عالية وذلك لأن إدارة موارد الشبكة موزعة على حواسيب الشبكة.

أما السبب الرئيسية لهذا النوع من الشبكات هو أنها غير مناسبة للشبكات الكبيرة وذلك لأنه عند نمو و توسع الشبكة وزيادة عدد مستخدميها يؤدي إلى مشاكل كثيرة.

ومن أنظمة التشغيل التي تدعم هذا النوع من الشبكات :

- ✓ Windows For Workgroup 3.11.
- ✓ Windows 95,98.
- ✓ Windows NT Workstation.
- ✓ Windows me, XP.
- ✓ Windows 2000 Professional.

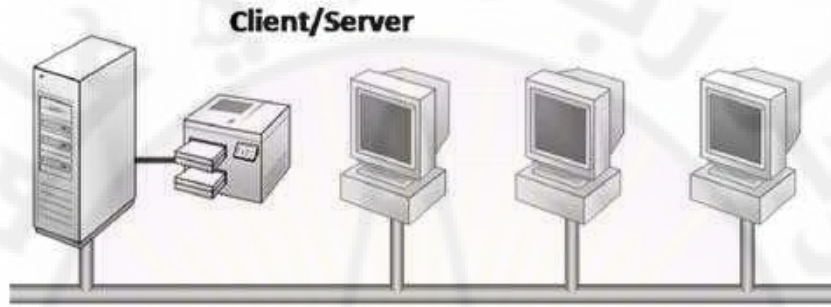
٤-٥-٢- شبكات المخدم/العميل (Client/Server)

هي الشبكات التي تعتمد على وجود حاسوب ذو مواصفات عالية ووحدات تخزين هائلة ، ويتم وضع البرامج التطبيقية المشتركة عليه وهذا الحاسوب يعمل كمخدم (Server) ، و يمكن أن يحتوي هذا النوع من الشبكات على أكثر من مخدم في حالة اتساع الشبكة ، وفي هذه الحالة تتوزع المهام على هذه المخدمات ما يزيد من كفاءتها ، ويبين الشكل (٤-٥) شبكات المخدم/العميل ، وهناك عدة أنواع للمخدمات حسب حاجة الشبكة :

- ◀ مخدم ملفات File Server .
- ◀ مخدم طباعة Printer Server .
- ◀ مخدم تطبيقات Application Server .
- ◀ مخدم قواعد معطيات Database Server .
- ◀ مخدم الاتصالات Communication Server .

وفي نظم التشغيل الجديدة لـ Windows يوجد نظام تشغيل Windows NT Server و نظام التشغيل Windows 2000 Server وهذان النظامان يدعمان المخدمات التالية :

- ◀ مخدم البريد الإلكتروني Mail Server .
- ◀ مخدم فاكس Fax Server .
- ◀ مخدم خدمات الدليل الفعال Active Directory Server .



الشكل (٤-٥) - يبين شبكات المخدم/العميل

يجب أن يكون نظام التشغيل قادراً على الاستفادة من كل قدرات المخدم ، وتتصف شبكات المخدم/العميل بالمواصفات التالية :

- ✓ يدعم هذا النوع من الشبكات آلاف المستخدمين .
- ✓ ليس من الضروري أن تكون حواسيب العملاء قوية وذات كفاءة عالية .
- ✓ توجد إدارة مركزية وجهاز واحد هو المخدم ويقوم بإدارة ومراقبة الشبكة.
- ✓ في هذا النوع من الشبكات يوفر المخدم حماية عالية من خلال السماح لشخص واحد هو مدير الشبكة (Administrator) بالتحكم في إدارة موارد الشبكة وإعطاء الصلاحيات و أذونات الوصول للمستخدمين.
- ✓ التركيب و الصيانة صعبة مقارنة بشبكات الند للند.
- ✓ التكلفة مرتفعة مقارنة بشبكات الند للند.

و من أنظمة التشغيل التي تدعم هذا النوع من الشبكات :

- ✓ Novell Netware Server .
- ✓ Unix Server.
- ✓ Windows NT Server .
- ✓ Windows 2000 Server .
- ✓ Windows 2003 Server .
- ✓ Windows 2008 Server .
- ✓ Windows 2010 Server .
- ✓ Windows 2012 Server .

٤-٦ - مكونات شبكات الحواسيب

أي شبكة حاسوبية تتكون مما يلي :

١. الحاسوب الرئيسي - المخدم Server : وهو الجهاز الرئيسي لتشغيل الشبكة

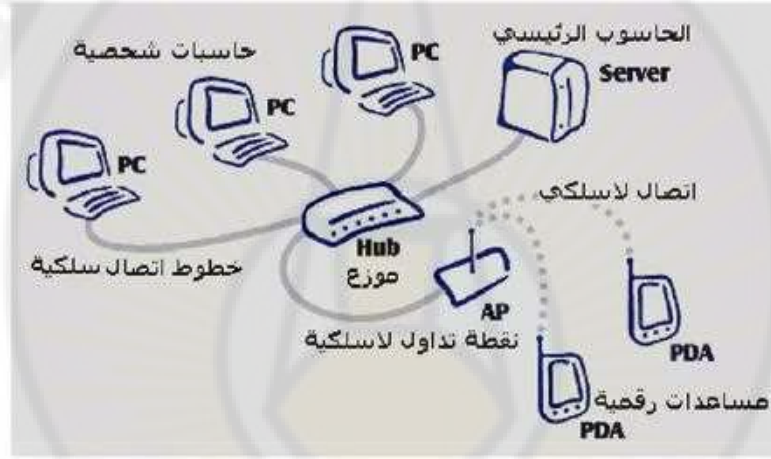
ويسمى جهاز الخدمة الرئيسي أو المخدم Server وهو عبارة عن حاسوب يتميز بالسرعة العالية والطاقة التخزينية الكبيرة لكي يستوعب المعطيات والبرمجيات التي سوف يتداولها المشاركون في الشبكة، ويقوم هذا الجهاز بالتحكم في جميع أجزاء الشبكة وذلك باستخدام برمجيات خاصة بتشغيل نظام الشبكة (Network Operating System) مثل : Unix , Novell Windows 2003 Server.

٢. محطات العمل Work Stations: وتسمى أيضاً Clients وهي الحواسيب

الشخصية بكافة أنواعها (مكتبية - محمولة - مساعدات رقمية - ...) أو الوحدات الطرفية Terminals والمتصلة بالجهاز الرئيسي ليستفيد

مستخدموها من المعطيات والبرمجيات المخزنة على جهاز الخدمة الرئيسي -
المخدم Server ، كما هو مبين في الشكل (٦-٤) .

٣. خطوط الاتصال **Communication Lines**: وهي الوسائل التي سيتم
بواسطتها تبادل المعطيات بين الحاسوب الرئيسي والحواسيب الفرعية
وتشمل الكبلات بأنواعها المختلفة كما تشمل الخطوط اللاسلكية
.Wireless



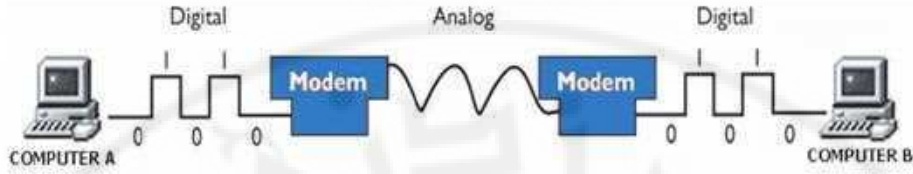
الشكل (٦-٤) -محطات العمل والمخدم وخطوط الاتصال في الشبكة

٤. بطاقة الشبكة **Network Interface Card**: هي بطاقة تثبت
بالحاسوب لتهيئته للاتصال بالشبكة، وتوجد البطاقة إما داخلية Internal
تثبت على اللوحة الأم Mother Board داخل الحاسوب أو خارجية
External ، ويبين الشكل (٧-٤) بطاقة الشبكة .



الشكل (٧-٤) - يبين بطاقة الشبكة

٦. **المودم Modem** : المودم عبارة عن لوحة أو شريحة إلكترونية تضاف إلى الحاسوب وتستخدم لتهيئة الحاسوب للاتصال بالانترنت من خلال خط الهاتف، ويقوم المودم بتحويل الإشارات الرقمية Digital Signals التي يستخدمها الحاسوب Modulate إلى إشارات تماثلية Analog Signals (نوع من الموجات المستخدمة في نقل الصوت كالتي يستخدمها الإنسان عبر خطوط الهاتف)، كما يقوم المودم بالعملية العكسية إذ يقوم بتحويل الإشارات التماثلية إلى رقمية Demodulate ، فكلمة مودم Modem هي اختصاراً لكلمتي (Modulate - Demodulate) أي التعديل وفك التعديل ، ويبين الشكل (٨-٤) طريقة عمل المودم .



الشكل (٤-٨) - يبين طريقة عمل المودم

٧. **الأجهزة الملحقة:** يمكن استخدام بعض الأجهزة وربطها بالشبكة مثل

الطابعات Printers وأجهزة الفاكس Fax والراسمات Plotters ويستطيع أي مشترك في الشبكة استخدام هذه الأجهزة.

٨. **مبدلات الشبكة Communication Switches:** هي عبارة عن أجهزة

تستخدم لربط حواسيب الشبكة بعضها ببعض وبين الشبكات ، ولتوجيه المعطيات بين حواسيب الشبكة، ومن هذه الأجهزة:

✓ الجسر Bridge

✓ والبوابة Gateway

✓ والموزع Hub

✓ الموجه Router.

٧. **برامج الشبكة:** هي برامج الاتصالات التي ستنحكم في تشغيل نظام

الشبكة ويتم تخزين هذه البرامج في الحاسوب الرئيسي Server (المخدم)،

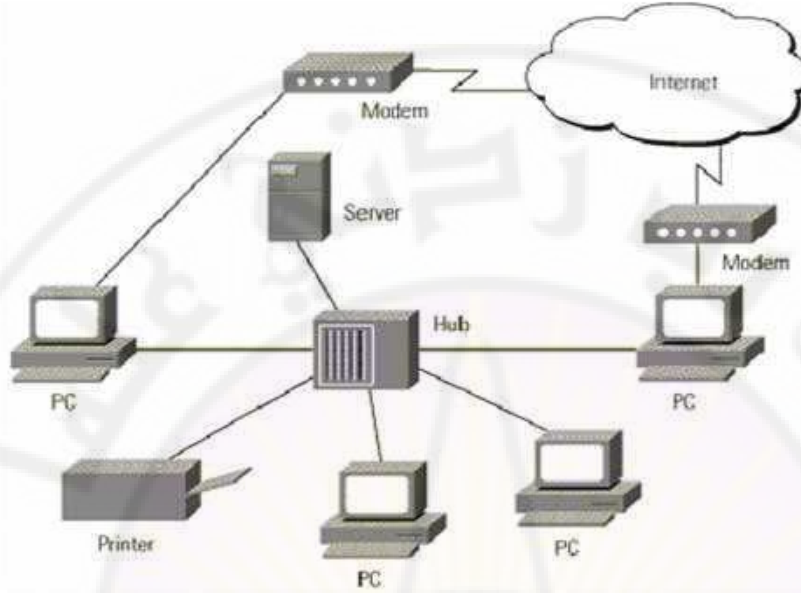
ومن أمثلتها :

✓ Novel

✓ Unix

✓ Windows 2003 Server.

ويبين الشكل (٤-٩) مكونات شبكة LAN .



الشكل (٩-٤) - يبين مكونات شبكة LAN

٤-٧- طوبولوجيا الشبكات المحلية

Local Area Networks Topology

يطلق اسم طوبولوجيا الشبكات Networks Topology على الطريقة التي يتم فيها تحديد كيفية ربط الحواسيب و المكونات الأخرى بالشبكة بعضها ببعض بحيث تعطي التوصيف السليم للبنية العامة لشبكة الحواسيب. إن اختلاف طريقة التوصيل يؤدي إلى اختلاف في المعدات التي تحتاجها الشبكة (نوع الكابلات ، نوع بطاقة الشبكة ، نوع الوصلات) كما يؤثر أيضاً على توسيع الشبكة مستقبلاً ، وبشكل عام يمكن تصنيف طوبولوجيا الشبكات المحلية إلى الطرق التالية :

- ١- طريقة التوصيل الخطي
- ٢- طريقة التوصيل النجمي
- ٣- طريقة التوصيل الحلقي
- ٤- طريقة التوصيل المتشابك

٤-٧-١ - طريقة التوصيل الخطي : Bus Topology

تعد هذه الطريقة من أبسط أنواع التوصيل ، يتم فيها ربط الحواسيب باستخدام كبل واحد يصل ما بين هذه الحواسيب، ويتم إرسال الإشارات إلى كل الحواسيب في الشبكة ، حيث يستطيع حاسوب واحد فقط الإرسال وباقي الحواسيب تنتظره حتى ينتهي من الإرسال ، ولكنها تؤخذ فقط من حاسوب واحد الذي يتوافق عنوانه مع العنوان الخاص بالإشارات .

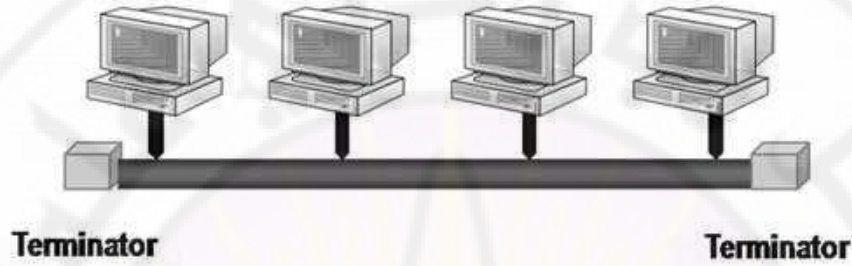
وكل حاسوب على الشبكة يبعث إشارات المعطيات في الاتجاهين على طول الكبل ، فإذا لم تجد هذه الإشارات نهاية ممانعة عندما تصل إلى نهاية الكبل فإنها تعكس اتجاهها في الكبل ما يمنع الحواسيب الأخرى من إرسال إشارتها ، لذلك يتم وضع نهاية ممانعة (Terminator) عند نهاية الكبل وذلك لإيقاف الإشارات ومنعها من الارتداد ، أي أنه يقوم بامتصاص الإشارات الحرة وهذا ينظف الكبل ما يسمح لحاسوب آخر بالإرسال، يبين الشكل (٤-١) طريقة التوصيل الخطي .

وإذا أردنا توسيع الشبكة أي زيادة الحواسيب المتصلة بالشبكة نقوم بإطالة الكبل بواسطة استخدام المعدات التالية :

١- الوصلات (Barrel Connection) : تقوم بوصل قسمين من الكبل لزيادة طوله.

٢- مقويات الإشارة (Repeaters) : تقوم بتقوية الإشارة لإرسالها لمسافة أطول.

و استخدام المزيد من الوصلات في الشبكة يؤدي إلى ضعف واضمحلال للإشارة قبل وصولها إلى الحاسوب المطلوب .



الشكل (١٠-٤) - يبين طريقة التوصيل الخطي

مميزات طريقة التوصيل الخطي

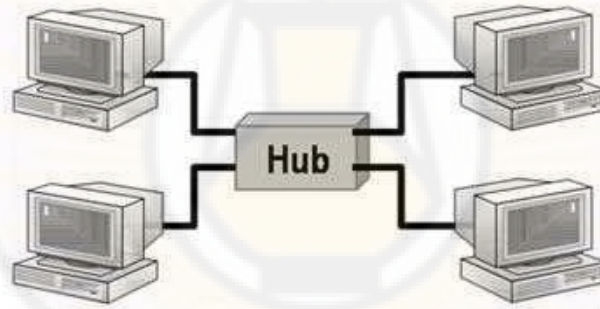
- ✓ إمكانية توسيع الشبكة بسهولة باستخدام الوصلات أو مقويات الإشارة.
- ✓ ذات تكلفة منخفضة لأنها تحتاج إلى أقل عدد من الكابلات مقارنة بطرق التوصيل الأخرى.
- ✓ تُعد أبسط وأسهل طرق التوصيل لأنها عبارة عن كابل واحد يصل ما بين جميع الحواسيب.

عيوب طريقة التوصيل الخطي

- ✓ كلما ازداد عدد الحواسيب، ضعفت الإشارة أكثر.
- ✓ صعوبة معرفة الأعطال.
- ✓ تتعطل الشبكة بشكل كامل إذا ما تعطلت وصلة الحاسوب أو حدث عيب في الكبل في أي منطقة.

٤-٧-٢- طريقة التوصيل النجمي (Star Topology)

تعد طريقة التوصيل النجمي من أشهر الطرق المعروفة لربط مكونات الشبكة بعضها ببعض ، ويتم فيها ربط كل حاسوب في الشبكة بكبل خاص إلى منافذ جهاز الـ (Hub) أو ما يسمى بوحدة التوصيل المركزية ، وفي النهاية يصبح الشكل النهائي للتوصيل يشبه النجمة Star لذا سميت هذه الطريقة بالتوصيل النجمي ، ويبين الشكل (٤-١١) طريقة التوصيل النجمي ، وفيها تنتقل الإشارات من الحاسوب المرسل الذي يرغب في إرسال المعطيات إلى جهاز الـ (Hub) ومنه إلى الحواسيب المرتبطة بها، وتحتوي الوحدة (Hub) على مفاتيح الكترونية Switches لتوصيل أي خط من الخطوط بخط آخر .



الشكل (٤-١١) - يبين طريقة التوصيل النجمي

عمل وحدة التوصيل المركزية Hub (المبدل)

تعمل وحدة التوصيل المركزية (Hub) بإحدى الطريقتين :

١. إما أن ترسل الإشارات إلى كل الحواسيب المرتبطة بها .
٢. أو أن ترسل الإشارات إلى الحاسوب المرسل إليه فقط وهذا النوع من الـ (Hub) يسمى بالـ (Switch) .

مميزات طريقة التوصيل النجمي :

- ✓ من السهل توسيع الشبكة (إضافة حواسيب جديدة) فكل ما نحتاجه هو كبل لوصل الحاسوب بأحد منافذ الـ (Hub).
- ✓ إذا تعطل أحد الحواسيب أو انقطع الكبل الخاص به استمرت الشبكة بالعمل ولا يتعطل غير هذا الحاسوب .
- ✓ يمكن تحريك الحواسيب من مكانها وصيانتها وتغيير التوصيلات دون أن تتأثر الشبكة.
- ✓ من السهل معرفة العطل وذلك بمجرد النظر إلى الـ (Hub) ، فكل منفذ يوجد بجانبه ديود ضوئي يضيء يشير في العمل ، فإذا انطفأ دل على أن الحاسوب المتصل به معطل.
- ✓ يمكن استخدام أنواع مختلفة من الكبلات في نفس الشبكة ، فإذا كنت تستخدم الكابلات (UTP) لربط الحواسيب بالـ (Hub) فيمكن استخدام الكابل المحوري Coaxial Cable أو الألياف الضوئية Fiber Optic لربط الـ (Hub) مع بعضها في حال وجود أكثر من (Hub).

عيوب طريقة التوصيل النجمي

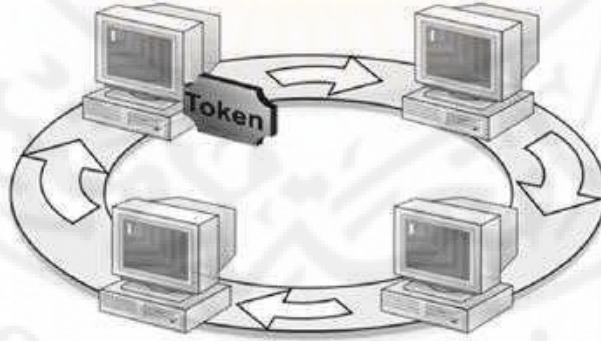
- ✓ تُعد هذه الطريقة ذات تكلفة مرتفعة لأنها تحتاج إلى كبلات أكثر ووحدات توصيل مركزية (Hub).
- ✓ تتعطل الشبكة بشكل كامل إذا تعطلت وحدة التوصيل المركزية (Hub).

٤-٧-٣- طريقة التوصيل الحلقي Ring Topology

يتم توصيل الحواسيب على كبل واحد على شكل حلقة، ويتم نقل المعطيات بين الحواسيب في اتجاه واحد عبر الكبل إلى أن تصل إلى الحاسوب المطلوب،

كما قد يكون الاتصال مزدوج الاتجاه حيث يمكن تمرير المعطيات في الاتجاهين مع عقارب الساعة أو عكسها ، ويتم تنسيق الاتصال خلال هذه الشبكة باستخدام بروتوكول خاص يسمى بروتوكول علامة المرور Token Passing Protocol . وفي هذه الطريقة يقوم كل حاسوب بإرسال المعطيات إلى الحاسوب الذي يليه، ويأخذ الحاسوب المعطيات ثم يعيد إرسالها مرة أخرى إلى الحاسوب الذي يليه ، وهكذا بحيث تنتقل الإشارات ضمن الحلقة باتجاه واحد وبترتيب واحد ، فكل الحواسيب متساوية ، وبما أن كل حاسوب يستقبل المعطيات ثم يعيد إرسالها على الشبكة إلى الحاسوب التالي، لذلك فكل حاسوب على الشبكة يقوم بعمل مقوي الإشارة ، حيث يقوم بتقوية الإشارة المارة من خلاله ، وهذا يؤدي إلى عدم ضعفها، كلما ازداد عدد الحواسيب يزداد زمن الاستجابة .

وتنتقل المعطيات باستخدام بروتوكول علامة المرور (Token Passing) التي تُعد وسيلة سريعة لنقل المعطيات ، وعندما يريد حاسوب ما على الشبكة إرسال معطيات ما فإن عليه الانتظار حتى يتسلم إشارة فارغة (Free Token) (تخبره على أنه قادر على إرسال معطياته على الشبكة، ويبين الشكل (٤-١٢) طريقة التوصيل الحلقي .



الشكل (٤-١٢) - يبين طريقة التوصيل الحلقي

مميزات طريقة التوصيل الحلقي

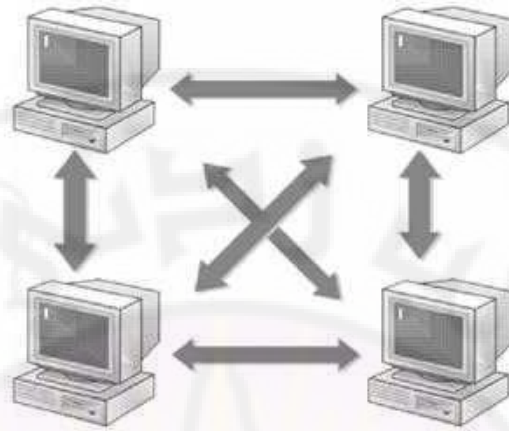
- ✓ تستخدم في الشبكات عالية الأداء والتي تحتاج إلى نقل كميات كبيرة من المعلومات.
- ✓ هذه الطريقة تستخدم بروتوكول علامة المرور (Token Passing) التي تُعد من الوسائل السريعة ، حيث تنتقل الإشارات من حاسوب إلى آخر بسرعة مقاربة لسرعة الضوء .
- ✓ تتمتع بدرجة عالية من الوثوقية ، فتعطل جهاز واحد لا يترتب عليه تعطل الشبكة .
- ✓ قابليتها للتوسع وإضافة أجهزة جديدة .
- ✓ تتميز بالسرعة والكفاءة .

عيوب طريقة التوصيل الحلقي

- ✓ إن إضافة أو إزالة حاسوب يؤثر على الشبكة.
- ✓ تتوقف الشبكة بالكامل عند تعطل إحدى الوحدات الطرفية .

٤-٧-٤ - طريقة التوصيل المتشابك Mesh Topology

في هذه الطريقة تتشابك كبلات توصيل الحواسيب بما يشبه خيوط شبكة الصيد ، إذ تمتد الكبلات من حاسوب لتصل إلى جميع الحواسيب وهكذا يتم توصيل كل حاسوب بجميع الحواسيب، ويبين الشكل (٤-١٣) طريقة التوصيل المتشابك .



الشكل (٤-١٣) طريقة التوصيل المتشابك

مميزات طريقة التوصيل المتشابك

- تعطل أحد الحواسيب أو أحد الوصلات لا يعطل الشبكة .
- يستطيع الحاسوب الاتصال بحاسوب آخر حتى لو انقطع الكبل الواصل بينهما وذلك من خلال الحواسيب الأخرى.

عيوب طريقة التوصيل المتشابك

- هذه الطريقة ذات تكلفة عالية حيث أنها تحتاج إلى كبلات كثيرة.
- صعوبة التركيب والإعداد.
- تكاليف الصيانة عالية بسبب تتبع المسارات.

٤-٨- وسائط نقل المعطيات في الشبكات

Communication Links

تنتقل المعطيات والمعلومات في شبكات الحواسيب عبر وسائل أو قنوات اتصال Communication Channels تربط بين عناصرها ، وتصنف وسائط الاتصال إلى نوعين :

- الوسائط السلكية

- الوسائط اللاسلكية

٤-٨-١- وسائط الاتصال السلكية

Wire Communications

وفيها يتم نقل المعطيات سواء كانت رقمية أو تمثيلية من خلال أسلاك متصلة مثل خطوط الهاتف أو الكابلات بأنواعها المختلفة ، ويطلق على الشبكات التي تستخدم الأسلاك بالشبكات السلكية ، وتنقسم الكابلات إلى ثلاثة أنواع :

١. الكابلات المزدوجة المجدولة Twisted Pair Cables .
٢. الكابلات المحورية Coaxial Cables .
٣. كابلات الألياف الضوئية Fiber Optics Cables .

العوامل التي تؤثر على اختيار الكابلات

لكي يتم اختيار أفضل نوع للكبل يناسب طبيعة الشبكة التي نقوم بتصميمها يجب دراسة مجموعة من العوامل التي يجب أخذها بعين الاعتبار عند الاختيار وهذه العوامل هي :

١. التكلفة Cost
٢. متطلبات التركيب Installing Requirements
٣. سرعة نقل المعلومات Transmission Speed
٤. التضاؤل Attenuation
٥. التداخل مع الموجات الكهرومغناطيسية EMI (Electro Magnetic Interference)
٦. أقصى طول كابل مسموح به Segment .
٧. أقصى عدد حواسيب لا Segment .

و في ما يلي شرح هذه العوامل :

١. التكلفة Cost

تلعب التكلفة دوراً كبيراً في اختيار نوع الكبل المستخدم ، فغالباً ما يكون هناك ارتباط بين سرعة نقل المعلومات وتكلفة الكبل ، فإذا كانت التكلفة الموضوعة لهذه الشبكة كبيرة يمكن اختيار كبل تكلفته عالية ولكنه ذو سرعة نقل عالية ، أما إذا كانت التكلفة الموضوعة قليلة فيجب مراعاة العلاقة بين سرعة النقل و تكلفة الكبل.

٢. متطلبات التركيب Installing Requirements

تختلف طريقة التركيب لكل نوع من أنواع الكابلات ، فمنها السهل و يمكن للشخص العادي تركيبها إلا التي تحتاج إلى خبرة في تركيبها والتي تحتاج إلى شركات خاصة.

٣. سرعة نقل المعلومات Transmission Speed

لكل نوع من أنواع الكابلات سرعة نقل معلومات مختلفة ، وتقاس سرعة نقل المعلومات بـ (Mega bit Per Second) Mbps .

٤. التضاؤل Attenuation

تنتقل المعلومات عبر وسائط نقل المعطيات (الكابلات) على شكل إشارات كهرومغناطيسية ، وهذه الإشارات تضعف قوتها على طول الكبل أي أن هناك علاقة عكسية بين التضاؤل وطول الكبل وبالتالي يتم تحديد طول الكبل بناءً على أثر عامل التضاؤل ولذلك يجب مراعاة الحد الأقصى لطول الكبل وذلك لتجنب التضاؤل.

٥. التداخل مع الموجات الكهرومغناطيسية (EMI)

هناك بعض الكابلات تتأثر بالموجات الكهرومغناطيسية ما يسبب تشويش على الإشارة المرسله داخل هذا الكبل ما يؤدي إلى حدوث أخطاء في المعطيات.

٤-٨-١-١ - الكابلات المزدوجة المجدولة

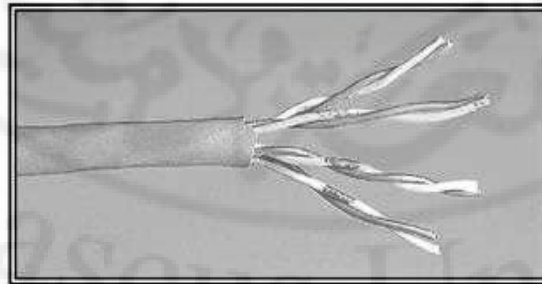
هي كابلات تشبه كابلات الهاتف العادي وهي أكثر وسائل الاتصال انتشاراً في الشبكات المحلية، فالكبل عبارة عن 4 أزواج من الأسلاك ملفوفة مع بعضها، وتنقسم الكابلات المزدوجة المجدولة إلى نوعين :

١. الكابلات المجدولة غير المغلفة (Unshielded Twisted Pair) UTP

٢. الكابلات المجدولة المغلفة (Shielded Twisted Pair) STP

١ - الكابلات المجدولة غير المغلفة UTP

الكبل المجدول غير المغلف عبارة عن عدد من الأزواج من الأسلاك النحاسية وكل زوج مكون من سلكين معزولين ومجدولين مع بعضهما ، و يتم وضع عدد من الأزواج المجدولة داخل غطاء بلاستيكي لين حسب نوع الكابل . إن عملية التجديل (Twisting) تقلل من تأثير عامل الـ Crosstalk وهو أثر التداخل بين الإشارات المارة في أحد الأسلاك مع الإشارات المارة في السلك الآخر، ويبين الشكل (٤-٤) الكبلات المجدولة .



الشكل (٤-٤) - يبين الكبلات المجدولة

قامت جمعية الصناعات الكهربائية EIA (Electrical Industrial Association) بتقسيم الكابلات المجدولة غير المغلفة (UTP) إلى الأصناف التالية:

الصف الأول : Category 1

يستخدم هذا النوع لنقل الصوت فقط و لا يستطيع نقل المعطيات .

الصف الثاني : Category 2

يستخدم لنقل المعلومات بسرعة تصل إلى 4 mbps وتتكون من أربعة أزواج مجدولة .

الصف الثالث : Category 3

يستخدم لنقل المعلومات بسرعة تصل إلى 10 mbps وتتكون من أربعة أزواج مجدولة .

الصف الرابع : Category 4

يستخدم لنقل المعلومات بسرعة تصل إلى 16 mbps وتتكون من أربعة أزواج مجدولة .

الصف الخامس : Category 5

يستخدم لنقل المعلومات بسرعة تصل إلى 100 mbps وتتكون من أربعة أزواج مجدولة .

الصف السادس : Category 6

يستخدم لنقل المعلومات بسرعة تصل إلى أكثر من 1000 mbps وتتكون من أربعة أزواج مجدولة وهي تستخدم في شبكات الـ Ethernet السريعة .

خصائص الكابلات المجدولة غير المغلفة UTP

أهم خصائص الكابلات ما يلي :

١. التكلفة

تعد الكابلات المجدولة غير المغلفة (UTP) من الكابلات ذات التكلفة المنخفضة .

٢. متطلبات التركيب

تعد الكابلات المجدولة غير المغلفة (UTP) من أسهل أنواع الكابلات في التركيب وأجهزته بسيطة.

٣. سرعة نقل المعلومات

يمكن للكابلات المجدولة غير المغلفة (UTP) نقل المعلومات بسرعات تختلف حسب الفئات التي تم تقسيمها من قبل جمعية الصناعات الكهربائية EIA.

٤. التضاؤل

بما أن الكابلات المجدولة غير المغلفة (UTP) مكونة من أسلاك نحاسية فهذا يعني أن انتقال الإشارة من خلال هذه الأسلاك يؤدي إلى ضعفها وتضاؤلها مع زيادة طول الكبل ولذلك تم تحديد طول الكبل المجدول غير المغلف (UTP) من الحاسوب إلى وحدة المبدل (Hub) بما لا يزيد عن 100 m وهذا ما يسمى بطول الـ Segment حيث تنتقل الإشارة خلال هذه المسافة بدون أن يطرأ عليها أي ضعف أو اضمحلال .

٥. أثر التداخل مع الموجات الكهرومغناطيسية (EMI)

إن الكابلات المجدولة غير المغلفة (UTP) سريعة التأثر بالتداخل مع الموجات الكهرومغناطيسية (EMI) وتعد الكابلات (UTP) من أكثر أنواع الكابلات تأثراً بالتداخل مع الموجات الكهرومغناطيسية (EMI).

٦. أقصى طول كابل مسموح به للقسم الواحد (Segment)

عند دراسة الكابلات لابد من معرفة معنى كلمة (Segment) ، وهو أقصى طول كبل مسموح بحيث تنتقل الإشارات دون أن يطرأ عليها اضمحلال أو تلاشي،

وطول هذا الـ Segment يختلف حسب نوع الكبل . وطول الـ Segment في الكابلات المجدولة غير المغلفة (UTP) هو 100 M .
٧ . عدد الحواسيب التي يمكن توصيلها للقسم الواحد (Segment) أقصى عدد حواسيب يمكن توصيلها للقسم الواحد (Segment) بدون استخدام مقويات للإشارة هو (1024) حاسوب .
٨ . الوصلات اللازمة للكابلات (UTP) تحتاج الكابلات (UTP) إلى وصلات RJ-45 .

٢ - الكابلات المجدولة المغلفة (STP)

لا تختلف الكابلات (STP) كثيراً عن الكابلات (UTP) إلا من حيث الحماية والتغليف حيث أن كل زوج مجدول محمي بطبقة من القصدير ثم بغلاف بلاستيكي خارجي .

خصائص الكابلات المجدولة المغلفة (STP)

أهم خصائص الكابلات ما يلي :

١ . التكلفة

تُعد الكابلات (STP) أكثر غلاءً من الكابلات (UTP) وذلك بسبب أغلفة الحماية .

٢ . متطلبات التركيب

إن تركيب الكابلات (STP) صعبة بسبب سماكتها التي تجعلها غير مرنة، كما أنها تحتاج إلى وصلات خاصة .

٣ . سرعة نقل المعلومات

تقارب سرعة نقل المعلومات في الكابلات (STP) سرعة نقل المعلومات في الكابلات (UTP) .

٤. التضاؤل

على الرغم من أغلفة الحماية ، فإنه لا تزال الإشارة المرسلة في الكبل (STP) عرضة للضعف و التلاشي لذلك فإن أقصى طول كبل للقسم الواحد (Segment) هو 100 M .

٥. أثر التداخل مع الموجات الكهرومغناطيسية (EMI)

إن أغلفة الحماية التي وضعت قللت من التأثير بالموجات الكهرومغناطيسية (EMI) إلا أنه لا يزال عرضة للتأثر بالـ (EMI)

٦. أقصى طول كابل للقسم الواحد (Segment)

إن أقصى طول كبل للـ Segment هو 100M وهو طول الكبل الذي تنتقل من خلاله الإشارات من دون أن تعاني ضعفاً أو تلاشياً.

٧. عدد الحواسيب الممكن توصيلها إلى الـ Segment الواحد

أقصى عدد حواسيب ممكن توصيلها إلى الـ Segment هو 1024 حاسوب وهذا العدد مشابه للعدد في الكبل (UTP) .

٤-٨-١-٢ - الكابلات المحورية Coaxial Cables

يتكون الكبل المحوري من سلك سميك من النحاس موجود داخل غلاف أبيض للحماية ثم يغلفه شبكة من الأسلاك الرفيعة محاط بطبقة عازلة تغطيها صفائح معدنية تغطي الموصل المركزي وتحميه من التيارات الكهربائية الخارجية وتأثير التداخل للموجات الكهرومغناطيسية (EMI) أو الـ CrossTalk ويحيط بها غطاء خارجي مصنوع من المطاط أو البلاستيك أو من مادة التفلون)

Teflon) ، ينقل هذا الكبل المعطيات بشكل إشارات رقمية كهربائية ويتميز بإرساله إشارات قوية وهي تشبه الكابلات المستخدمة في التوصيلات التلفزيونية. وهناك نوعين من الكابلات المحورية :

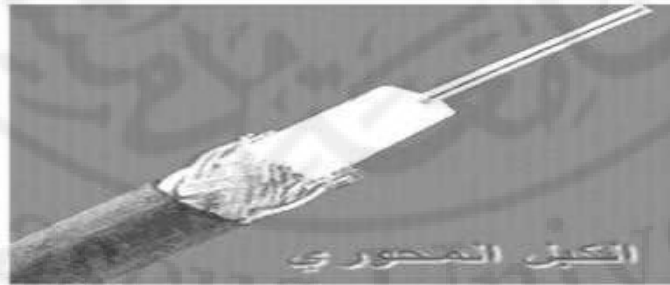
- الكبل المحوري الرفيع Thin Coaxial Cable .
- الكبل المحوري السميك Thick Coaxial Cable .

١- الكبل المحوري الرفيع

عبارة عن كبل مرن و رقيق سماكته 0.6 Cm ، ويستخدم هذا الكبل للإرسال العادي للمعطيات ، ويستخدم في ربط الحواسيب لمسافة تتراوح بين 15m إلى 185m .

٢- الكبل المحوري السميك

عبارة عن كابل غير مرن ، سميك ، تصل سماكته تقريباً إلى 1.3Cm وبما أنه أكثر سماكة من الكبل المحوري الرفيع فإنه يستطيع نقل الإشارات إلى مسافة أطول بدون أن يطرأ عليها ضعف أو تضائل ، فبينما لا يستطيع الكبل المحوري الرفيع نقل الإشارة لأكثر من 185M بدون أن يطرأ عليها أي ضعف أو تتضاءل ويستطيع الكبل المحوري السميك نقل الإشارة إلى 500 M ، ويبين الشكل (٤-١٥) الكبل المحوري .



الشكل (٤-١٥) - يبين الكبل المحوري

خصائص الكابلات المحورية

إن خصائص الكابلات المحورية هي :

١. التكلفة

يعد الكبل المحوري السميك أغلى من الكبل المحوري الرفيع ولكنه أرخص من كابلات الألياف الضوئية .

٢. متطلبات التركيب

الكبل المحوري الرفيع من السهل تركيبه بينما الكبل المحوري السميك تركيبه أصعب نظراً لسماكته .

٣. سرعة نقل المعلومات

كلاهما له سرعة نقل معلومات تصل إلى 10 Mbps .

٤. التضاؤل

عموماً تُعد الكابلات المحورية أقل تأثراً من الكابلات المجدولة ولكن الكبل المحوري الرفيع أكثر تأثراً من الكبل المحوري السميك ، لذلك فإن الكبل المحوري الرفيع يستطيع نقل الإشارة لمسافة 185M بدون أن يطرأ عليها أي ضعف بينما الكبل المحوري السميك يستطيع نقل الإشارة لمسافة تصل إلى 500 M .

٥. أثر التداخل مع الموجات الكهرومغناطيسية (EMI)

تعد الكابلات المحورية أقل تأثراً من الكابلات المجدولة ولكنها لا تزال تتأثر بـ EMI ولكن الكبل المحوري السميك أقل تأثراً من الكبل المحوري الرفيع.

٦. أقصى طول كابل للـ Segment

بما أن الكابلات المحورية السميكة أقل تأثراً بالتضاؤل من الكابلات المحورية الرفيعة فإن أقصى طول كابل محوري رفيع بدون مقوي إشارة هو 185M بينما أقصى طول كابل محوري سميك بدون مقوي للإشارة هو 500M.

٧. عدد الحواسيب التي يمكن توصيلها لل Segment الواحد
أقصى عدد حواسيب يمكن توصيلها بدون استخدام مقوي للإشارة للكبل
المحوري الرفيع هو 30 حاسوب ، بينما أقصى عدد حواسيب يمكن توصيلها
بدون استخدام مقوي للإشارة للكبل المحوري السميك هو 100 حاسوب.

□ الوصلات اللازمة للكابلات المحورية

تستخدم الكابلات المحورية بنوعيتها وصلات خاصة لتقوم بعملية الوصل بين
الكابلات و المكونات الشبكية ، ويوجد عدة أنواع من هذه الوصلات هي :

- **BNC T Connector** : وهي الوصلة التي تقوم بربط بطاقة الشبكة
مع الكبل .
- **BNC Cable Connector** : وهي الوصلة التي تربط نهاية الكبل مع
الوصلة T .
- **BNC barrel Connector** : تقوم بتوصيل قسمين من الكبل
المحوري الرفيع لزيادة طول الكبل .
- **BNC Terminator** : تقوم بإغلاق نهاية الكبل لامتناس الإشارات
الحرّة .

□ أصناف الكابلات المحورية

تم تصنيف الكابلات المحورية إلى صنفين وفقاً لتركيب الغلاف الخارجي و
المكان الذي سيركب فيه، وهذان الصنفان هما :

أ. PVC(Polyvinyl Chloride)

هو أحد أنواع البلاستيك الذي يستخدم في صناعة الغلاف الخارجي للكبل
وهو الأكثر استخداماً وهو مرن وسهل التركيب ولكنه في حالة الاحتراق يعطي
غازاً ساماً لذلك يفضل استخدامه في الأماكن المفتوحة والمعرضة للتهوية .

ب.البليونيوم Plenum

وهو مصنوع من مواد مضادة للحريق ، ويستخدم في المباني المتعددة الطوابق في الفراغ الذي يفصل السقف وأرض الطابق الذي فوقه ، كما يمكن أيضاً استخدامه في الجدران بدون مجاري أو قنوات ولكنه أقل مرونة وأعلى ثمناً .

٤-٨-١-٣ - كابلات الألياف الضوئية Fiber Optic Cables

يتكون الليف الضوئي من سلك من الزجاج رقيق جداً بسمكة شعرة الرأس ، يطلق عليه قلب زجاجي (Core) ويحاط القلب الزجاجي بدرع بلاستيكي يسمى الغلاف (Cladding) ، وهذا الغلاف مغطى بألياف تقوية (Kevlar) والتي بدورها تكون مغطاة بمادة من البلاستيك للحماية ، وبما أن القلب الزجاجي (Core) ينقل الضوء في اتجاه واحد فلا بد من استخدام سلكين واحد للإرسال والآخر للاستقبال ، يبين الشكل (٤-١٦) كبل الألياف الضوئية .



الشكل (٤-١٦) - يبين كبل الألياف الضوئية

يُعد من أفضل أنواع الكابلات حيث تستخدم الألياف الزجاجية التي تنقل خلالها المعطيات بشكل إشارات رقمية ضوئية ، ويتكون الكبل من حزمة من الموصلات الزجاجية المصنوعة من السليكون النقي والقادر على نقل الضوء ،

وتستخدم أنواع عديدة من الألياف البصرية ، أهمها الألياف الزجاجية التي تستطيع أن تنقل 160 مليون مكالمات هاتفية و 80 ألف قناة تلفزيونية في آن واحد لمستخدم النظام الرقمي ، وسرعتها تبلغ أضعاف سرعة الكبلات المحورية .

□ خصائص كابلات الألياف الضوئية

١. التكلفة : تُعد كابلات الألياف الضوئية الأغلى على الإطلاق على مستوى الكابلات .
٢. التركيب : صعبة التركيب وتحتاج إلى خبرة وأجهزة خاصة ، فتركيب هذه الكابلات وصيانتها أمر صعب فأى كسر أو انحناء يؤدي إلى توقف نقل البيانات.
٣. سرعة نقل المعلومات : توفر كابلات الألياف الضوئية سرعات كبيرة جداً مقارنة بالكابلات الأخرى ، وفي كل يوم ينتج أنواع كابلات جديدة ذات سرعات أعلى.
٤. التضائل : تأثير هذا العامل على كابلات الألياف الضوئية ضعيف جداً .
٥. التداخل مع الموجات الكهرومغناطيسية EMI : لا تتأثر مطلقاً بالـ (EMI) و تنقل ضوءاً وليس إشارات كهربائية ، ومستوى الأمن عالٍ ولا يمكن التجسس على البيانات المنقولة من خلالها.

٤-٨-٢- وسائل الاتصال اللاسلكية

Wireless Communications

تستخدم الأمواج الراديوية والأمواج الميكروية والأقمار الصناعية وغيرها من وسائل الاتصال اللاسلكية في نقل المعطيات والمعلومات سواء أكانت رقمية أو تمثيلية . أما أنواع وسائل الاتصال اللاسلكية هي :

- الأمواج الراديوية Broadcast Radio Transmissions
- والأمواج الميكروية Microwave System

- الأقمار الصناعية Satellite

٤-٨-٢-١ - الأمواج الراديوية

Broadcast Radio Transmissions

تستخدم الأمواج الراديوية اللاسلكية لإرسال المعطيات بأشكالها المختلفة (نصوص، صور، صوت، فيديو) من موقع إلى آخر ، ويمكن تصميم الشبكة المحلية لاسلكياً Wireless LAN وتستخدم في مقاهي الانترنت، والفنادق، والجامعات وغيرها وتعرف بالنقاط الساخنة Hotspots.

٤-٨-٢-٢ - الأمواج الميكروية Microwave System

يمكن استخدام موجات لاسلكية عالية التردد تعرف باسم الأمواج الميكروية Microwave في نقل المعطيات في الفضاء بين الحواسيب باستخدام هوائيات صحنية Dishes، ويتم تثبيتها في أماكن مرتفعة مثل قمم الأبراج والمباني لتوفير ممر صافٍ بين الهوائيات لنقل المعطيات، ويبين الشكل (٤-١٧) وسائط الاتصال بالأمواج الميكروية .

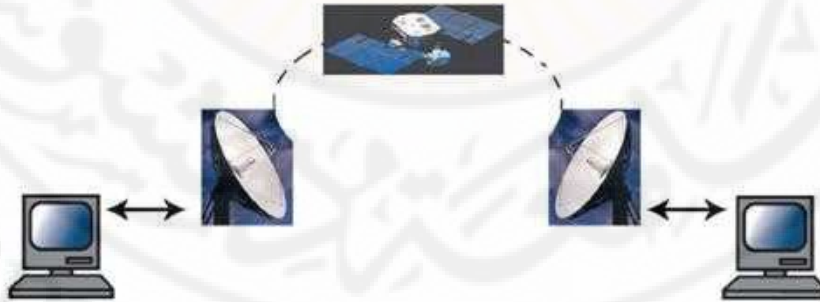


الشكل (٤-١٧) - يبين وسائط الاتصال بالأمواج الميكروية

ويستخدم هذا النظام عادة في الشبكات الواسعة WAN ، تستخدم لبث الصوت والمعلومات الصوتية مع استخدام محطات تقوية حيث تلتقط هذه الموجات ثم تعيد بثها بعد تقويتها ما يسمح بنقلها إلى مسافات بعيدة .
الأمواج الميكروية (الميكروويف) هي أمواج قصيرة ذات عرض مجال ترددي كبير (High Band Width) ، وتتميز بالسعة والسرعة الفائقة في حمل ونقل المعلومات .

٤-٨-٢-٣ - الأقمار الصناعية Satellite

تستخدم الأقمار الصناعية محطات أرضية لبث وتوزيع والتقاط المعطيات والمعلومات الصوتية والمرئية عبر الفضاء ، ويتم تثبيت هذه الأقمار في مدارات ثابتة فوق الأرض حيث تستقبل المعطيات المرسلّة من شبكة الحواسيب عن طريق المحطات الأرضية فتقوم بتقوية الإشارة وتغيير التردد وإعادة إرسال المعطيات إلى المحطة الأرضية المستقبلية التي تقوم بإرسالها إلى شبكة الحواسيب المستقبلية ،
ويبين الشكل (٤-١٧) وسائط الاتصال بالأقمار الصناعية .



الشكل (٤-١٧) - يبين وسائط الاتصال بالأقمار الصناعية

٤-٩- بروتوكولات شبكات الحواسيب

Computer Networks Protocols

٤-٩-١- مقدمة

يعرف البروتوكول بشكل عام بأنه مجموعة القواعد والقوانين التي تؤدي إلى تنفيذ العمل المنفق عليه بشكل سليم وخالي من الأخطاء.

إن عملية تبادل المعلومات الرقمية الصادرة من حاسوب إلى آخر لابد أن تخضع إلى مجموعة من القواعد والقوانين بحيث تصل بشكل سليم ، وبالتالي يمكن تعريف البروتوكول المستخدم في شبكات الحواسيب بأنه مجموعة من القواعد والقوانين التي تنظم عملية الاتصال بين الحواسيب بحيث يضمن تبادل المعلومات بشكل سليم وخالي من الأخطاء.

بسبب عدم قدرة شبكات الحواسيب الاتصال ببعضها البعض ، لتتمكن من تبادل المعلومات ، قامت عدة هيئات ولجان دولية بوضع معايير قياسية مناسبة لشبكات الحواسيب لجعل هذه الشبكات تتصل فيما بينها.

٤-٩-٢- المنظمات الدولية للمقاييس والمعايير

Standard Organizations

هناك عدد من الهيئات واللجان الدولية التي تهتم بوضع المعايير القياسية لمختلف أنواع المنتجات بدءاً من المنتجات الصغيرة إلى المنتجات الكبيرة . وفيما يلي سوف نستعرض أهم الهيئات واللجان الدولية التي تهتم بوضع المعايير القياسية الخاصة بشبكات الحواسيب.

١. المنظمة الدولية للمواصفات والمقاييس ISO

International Standard Organization

تهتم هذه المنظمة الدولية بتوحيد المعايير القياسية الدولية في مختلف المجالات ، وقد أصدرت آلاف المعايير في المجالات المختلفة ، وقسمت هذه

المنظمة إلى لجان فنية كل منها يهتم بوضع معايير قياسية في مجال معين ، وكل لجنة فنية تتكون من لجان فرعية ، وكل لجنة فرعية تتكون من عدة مجموعات عمل تهتم بإصدار المعايير المختلفة في المجالات المختلفة ، من بين هذه اللجان الفنية ، هناك لجان مختصة بوضع المعايير في مجال شبكات الحواسيب.

٢. الاتحاد العالمي للاتصالات ITU

International Telecommunication Union

يهتم (ITU) بوضع المعايير الخاصة بنظم الاتصالات العالمية المختلفة ، وبسبب التطور السريع في تقنيات الاتصالات والترابط الكبير بين شبكات الحاسوب ونظم الاتصالات ، اهتمت المنظمة (ITU) في نظم اتصالات المعلومات بطرق نقل المعلومات عبر شبكات الحواسيب .

٣. المعهد الوطني الأمريكي للمعايير القياسية ANSI

American National Standard Institute

يهتم هذا المعهد بوضع المعايير للمنتجات المختلفة في مختلف المجالات ، وهو يشابه المنظمة الدولية للمعايير القياسية ISO.

٤. هيئة الصناعات الكهربائية EIA

Electrical Industrial Association

تهتم هذه الهيئة بوضع المعايير الخاصة بمكونات الربط مع الشبكات ومعظم مواصفات الطبقة الفيزيائية في نموذج السبع طبقات (OSI)، فهي التي قسمت الكابلات المجدولة غير المغلفة (UTP) إلى فئات مختلفة حسب سرعة نقل المعلومات .

٥. معهد مهندسي الكهرباء والإلكترون IEEE

Institute of Electrical And Electronic Engineers

يهتم هذا المعهد بوضع المعايير الخاصة بالطبقة الفيزيائية وطبقة ربط المعطيات في نظام السبع طبقات (OSI) وتأمين النقل السليم للمعلومات، فهو الذي قام بتقسيم طبقة ربط المعطيات نظراً لأهميتها إلى طبقتين فرعيتين هما طبقة التحكم بالنفاذ إلى الوسط (MAC) وطبقة التحكم في الاتصال المنطقي (LLC)، وأيضاً هو المسؤول بأن يكون لكل بطاقة شبكة عنوان وحيد ومميز يسمى العنوان الفيزيائي (MAC Address).

يتبع اسم المعهد رقم يدل على رقم مجموعة العمل الذي أصدرت هذا المعيار، وهناك عدد من المعايير من أهمها المعيار IEEE 802 الذي صدر عنه عدد من المعايير أهمها :

- ✓ المعيار IEEE 802.3 الخاص بشبكات Ethernet .
- ✓ المعيار IEEE 802.4 الخاص بالشبكات التي تستخدم التوصيل الخطي وطريقة بروتوكول علامة المرور .
- ✓ المعيار IEEE 802.5 الخاص بشبكات Token Ring .

٤-٩-٣ - المهام الأساسية لبروتوكول شبكات الحواسيب

قبل أن نقوم بدراسة أهم البروتوكولات المستخدمة في شبكات الحواسيب المختلفة ، لابد من معرفة أهم المهام الأساسية التي يجب أن تتوفر في بروتوكول شبكات الحواسيب، وهذه المهام تختلف حسب ما يكون هذا الحاسوب هو الذي يقوم بإرسال المعلومات أو أن يقوم باستقبال هذه المعلومات.

١ - مهام البروتوكول في الحاسوب المرسل

يقوم البروتوكول في الحاسوب المرسل بالمهام التالية :

١. تقسيم المعلومات إلى رزم صغيرة.
 ٢. إضافة معلومات خاصة بالعنونة إلى الرزم.
 ٣. إضافة معلومات خاصة باختبار حدوث الخطأ في المعطيات.
 ٤. إضافة معلومات خاصة بنوع البروتوكول.
- بعد ذلك يتم إرسال هذه الرزم عبر بطاقة الشبكة ومن ثم عبر الكبل إلى الحاسوب المستقبل.

٢ - مهام البروتوكول في الحاسوب المستقبل

يقوم البروتوكول في الحاسوب المستقبل بالمهام التالية :

١. استلام الرزم من الكبل.
٢. نقل هذه الرزم عبر بطاقة الشبكة إلى الحاسوب.
٣. بعد التأكد من سلامة المعلومات يتم حذف معلومات التحكم.
٤. تجميع الرزم وإعادة تشكيل المعلومات الأصلية لإعطائها إلى البرنامج.
٥. اختبار سلامة المعلومات من الأخطاء بعد التجميع.

=====



٤-١٠- شبكة الانترنت

٤-١٠-١ تعريف شبكة الانترنت

شبكة الانترنت عبارة عن شبكة واسعة من الحواسيب الموصولة مع بعضها والموزعة في جميع أنحاء العالم وتخزن كمّاً هائلاً من المعلومات بأشكال مختلفة مثل النصوص وملفات الصوت والرسومات والصور المتحركة، وتوفر خدمة البريد الإلكتروني E-mail التي تعد من أسرع وسائل الاتصال الحديثة وأرخصها ، وهذه المعلومات تشمل كافة نواحي المعرفة من علوم وتكنولوجيا وصحافة وأعمال مصرفية وألعاب وغيرها ، ويمكن الوصول إليها من أي مكان في العالم حيث يتوفر فيه الحاسوب بمودم وبرمجيات الاتصال المناسبة وخط هاتفي .

٤-١٠-٢ التطور التاريخي لشبكة الانترنت

يعود إنشاء شبكة الإنترنت إلى عام 1970 تقريباً أبان الحرب الباردة بين القوتين العظميين في العالم الولايات المتحدة الأمريكية والاتحاد السوفييتي، حيث نشأت كمشروع لحساب وزارة الدفاع الأمريكية من أجل صناعة نظام يمكنه التصدي لأي هجوم سوفيتي محتمل على الولايات المتحدة ، وقد قامت وكالة ARPA المعنية بالمشروع بربط أربعة حواسيب في شبكة واحدة أطلقت عليها اسم ARPANET ، ويمكن لهذه الشبكة أن تعمل في ظل حدوث أي عطل يصيب حاسوباً أو حاسوبين ، وقد اعتمدت العديد من المؤسسات التخصصية تطبيق هذه التقنية في ربط حواسيبها ، و تضمن وصول المعلومة بين أطراف الشبكة بدون أي عرقلة في منطقة وسط الشبكة ، ويرجع النجاح في هذا الأسلوب من الربط إلى استخدام تقنية التبادل الرزمي (Packet Switching).

في أوائل الثمانينات قامت العديد من الوكالات الأمريكية الفيدرالية بإنشاء شبكة تعليمية لربط الجامعات المختلفة من خلال شبكة ARPANET ، وذلك

بالإضافة إلى اشتراك العديد من الشبكات التجارية خلال هذه الشبكة ، ما أدى إلى تكون العمود الفقري لشبكة الإنترنت في عام 1983، عندما انقسمت شبكة الـ ARPANET إلى قسمين : الـ ARPANET التي اقتصرت على الأبحاث المدنية والـ MILNET التي استخدمت في المجالات العسكرية .

في عام 1986 قامت مؤسسة العلوم الوطنية (NSF) National Science Foundation (بتشجيع الباحثين بعضهم ببعض في كافة أنحاء الولايات المتحدة باستخدام خمسة حواسيب عملاقة على هيئة شبكة باسم NSF، وتكونت هذه الشبكة من مراكز لخطوط الإرسال مستخدمة الألياف الضوئية والأسلاك العادية بمساعدة الأقمار الصناعية والأمواج الميكروية Microwave لكي تستطيع نقل كميات هائلة من المعلومات بسرعة ولمسافات بعيدة وكانت بذلك أساس الإنترنت التي بدأت عمليا عام 1985 .

وقد تكلفت وكالة العلوم الوطنية في الولايات المتحدة الأمريكية بالمحافظة على هذه الشبكة وصيانتها ، أما الشبكات الأخرى التابعة لشبكة الإنترنت فكل منها يتكفل بتكاليف شبكته سواء أكانت شبكة تجارية أم حكومية أو تعليمية . وتعد شبكة الإنترنت أحد أهم مصادر المعلومات على المستوى العالمي حيث يبلغ عدد المواقع الرئيسية Web Sites ملايين المواقع منها التجارية والتعليمية والعلمية والاقتصادية والحكومية والعسكرية ، بينما يبلغ عدد المشتركين بهذه الشبكة حالياً أكثر من ثلاثة مليارات شخص في أكثر من 160 دولة ، ويبلغ معدل الزيادة السنوي حوالي 30% من أجمالي المتعاملين مع الشبكة ، ولا يقتصر مجال عمل الشبكة على نقل المعطيات فحسب ، بل يمتد ذلك إلى نقل الصور والأشكال والأصوات ، وتبادل المحادثات التلفونية المرئية ، وزاد من فاعلية هذه الشبكة ارتباط الأنظمة الآلية لشركات الطيران وشركات النقل الدولية بها مما أصبح بإمكان أي جهاز متعامل مع الشبكة الاتصال بهذه الأنظمة

والحصول على الخدمات المطلوبة ، بالإضافة إلى اتصال دور نشر الصحف بالشبكة من خلال خدمة Usenet ، والتي توفر حالياً الإطلاع على أكثر من 20 ألف مجموعة أنباء ، وقد ظهرت الخدمات المختلفة لشبكة الانترنت بصورة تدريجية ، حيث تم التعامل مع خدمة Usenet خلال عام 1981 ، وخدمة Gopher في عام 1982 ، وظهرت خدمة الشبكة العنكبوتية WWW (World Wide Web) في عام 1989 والتي تتيح نقل الرسومات والصور والأصوات والفيديو عبر الانترنت.

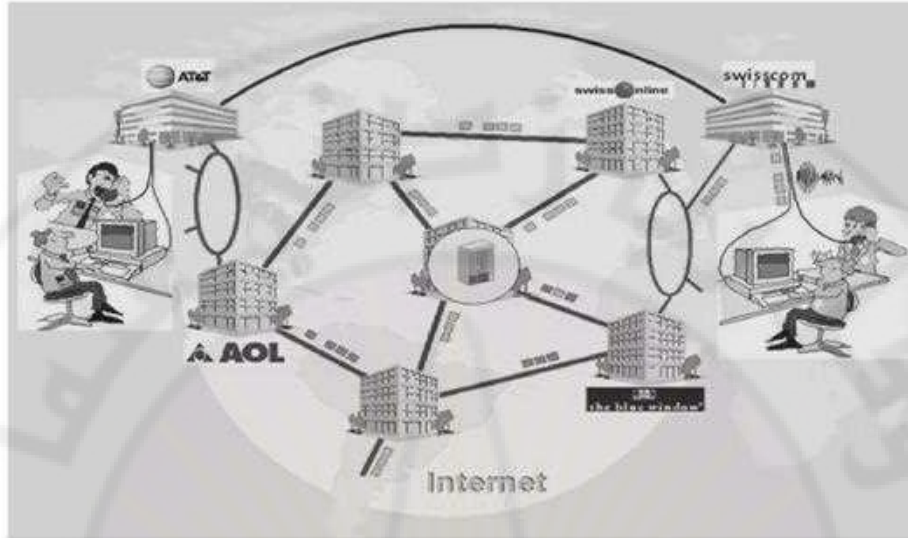
ومع هذا التعدد الهائل من الشبكات كان من الضروري أن يتم تطوير نظام اتصالات Communication Protocol يسمح بإيجاد وسيلة للتخاطب بين هذه الشبكات ، ولذلك فقد تم اعتماد نظام الاتصالات TCP/IP كنظام اتصال موحد بين أجزاء الشبكة المختلفة ، وفي التسعينات من القرن الماضي تلقت شبكة الانترنت دفعة هائلة حين أقر الكونغرس الأمريكي قانون تقنية الحواسيب المتقدمة ، مع تخصيص جزء كبير من شبكات الراديو الحكومية لاستخدام الشبكة.

وجدير بالذكر أن هذه الشبكة ، فضلاً عن كونها أهم مصادر المعلومات العالمية في كافة المجالات نظراً لتعاملها مع العديد من بنوك المعلومات العالمية ، فإنها تلعب دوراً أساسياً في تنمية البحث العلمي ، وتساعد الشركات التجارية على دخول الأسواق العالمية المختلفة ، وتقدم الشبكة خدمة البريد الإلكتروني ، التي تسمح بالتخاطب بين شخصين أو أكثر بواسطة الرسائل البريدية ، كما تقوم الشبكة بتوصيل النصوص الإلكترونية إلى ملفات بريد المتعاملين مع الشبكة ، وهذه النصوص تشمل المعلومات العلمية والتعليمية والاقتصادية والتجارية والسياسية والدورات اللغوية والأفلام وأسعار البورصات العالمية وأسواق المال والعملات والمعادن والسلع وغير ذلك .

وتسهم شبكات الإنترنت في المجالات التجارية في خفض إنفاق الشركات إلى حد كبير من متطلبات السفر للتفاوض بشأن الصفقات التجارية ، حيث يمكن إتمام هذه الصفقات بواسطة الشبكة وكذلك للمشاركين ، ويمكن للمشاركين فيها إرسال رسائل سريعة لعملائهم بالدول المختلفة ، وعرض منتجاتهم في كافة القطاعات التجارية .

وفي مجال التعليم والبحث العلمي فإن لشبكة الإنترنت أفقاً مفتوحة حيث إنها توفر كافة المعلومات العلمية والبحوث والدراسات التخصصية للباحثين في المجالات المختلفة ، كما تتيح التعاون المشترك بين أكثر من باحث في مواقع مختلفة ، بالإضافة إلى إمكانية عقد الندوات واللقاءات وتبادل الوثائق بين المشاركين بالشبكة .

وتتم معظم التعاملات في الإنترنت باللغة الإنكليزية لعدة أسباب منها أن الشبكة ظهرت في الولايات المتحدة الأمريكية بالإضافة إلى أن أجهزة الحواسيب المستخدمة على الشبكة انتشرت تجارياً أيضاً في الولايات المتحدة الأمريكية ، إلا أن العائق التقني والذي يتمثل في أن جميع المعايير المستخدمة في شبكة الانترنت مصممة حسب نظام (7Bit) ASCII ، المخصص أساساً للغة الإنكليزية ، ومن هذا المنطلق وأمام الانتشار الواسع للشبكة ، وجدت عدة شركات فرصة للاستثمار في هذا المجال ، بحيث يتم وضع نظام رقمي موحد ومعايير تكون مناسبة لجميع اللغات ، فمثلاً من البرامج التي ظهرت برنامج كوميونيك للبريد الإلكتروني الذي يتعامل مع نظام (8Bit) بحيث يكون مناسباً للغة الفرنسية ، إلا أنه من الناحية التقنية يمكن أن يخدم اللغة العربية مما يتيح إرسال واستلام الرسائل المطلوبة بين مستخدمي الشبكة باللغة العربية ، ويبين الشكل (٤-١٩) شبكة الانترنت .



الشكل (٤-١٩) - يبين شبكة الانترنت

٤-١٠-٣ - إدارة شبكة الإنترنت

لا توجد جهة مركزية تدير شبكة الإنترنت ، ولا يوجد إدارة معينة مسؤولة عنها ، فكل جهة مركزية Main Domain مرتبطة بهذه الشبكة مسؤولة من جهتها بواسطة وكيل محلي Provider Service يحدد قواعد التعامل مع المستخدمين المحليين ، ويباشر أمور العمل اليومي لهذا التعامل . ولكن من جهة أخرى فإنه توجد بعض المنظمات التي تعني بأمور الشبكة وذلك على النحو التالي :

- إن من يدير شبكة الانترنت ويتحكم بالمعايير الفنية النازمة لعملها هي جمعية الانترنت IOSC (Internet Society) وتتحصر مهمة هذه الجمعية في تأمين التنسيق والتعاون بين أطراف الشبكة ورسم سياسات واتجاهات تطورها في المستقبل .

- أما عن الاعتبارات الفنية المتعلقة باعتماد وتطوير أنظمة الاتصالات بين أجهزة الشبكة والتطبيقات الخاصة بها فيتم من خلال IAB Internet Architecture Board () ، وتباشر لجنة IETF (Internet Engineering Task Force) وضع القواعد القياسية لبعض الأمور الفنية المتعلقة بالشبكة ، وهي عبارة عن مجموعة من المختصين الذين يعملون على تطوير أداء الشبكة وتوسيع خدماتها .
- تختص Internet NIC بعملية تسجيل الشبكات المتصلة مع شبكة الإنترنت ، بالإضافة إلى تزويد المستخدمين بالخدمات الاستثنائية المتعلقة بالعمل على الشبكة .
- المركز الرئيسي للشبكة في الولايات المتحدة الأمريكية Backbone يتم مباشرته بما يسمى National Science Foundation .

٤-١-٤ - متطلبات الدخول إلى الإنترنت

تختلف متطلبات الدخول إلى الإنترنت بين كون المستثمر ينتمي إلى شركة أو مؤسسة لها شبكتها الخاصة المرتبطة بشبكة الإنترنت أو أن يكون المستثمر مستقلاً عن أي شبكة ويرغب في استخدام خدمات الإنترنت ، لكن يمكن تحديد المتطلبات المشتركة بما يلي :

- ١- أن تكون قابلاً للاتصال بأحد مخدمات الإنترنت .
 - ٢- أن تمتلك عنواناً خاصاً بك في الإنترنت كي يتم تبادل المعلومات معك .
 - ٣- أن تمتلك البرمجيات التي تساعدك للوصول والتجول في الإنترنت واستعراض المعلومات أو تبادلها .
- من المعروف أن الإنترنت تتألف من المكونات الفيزيائية للشبكة أي من خطوط اتصال وأجهزة رقمية تشكل الممر الواصل بين الحواسيب ، وتعد هذه

الحواسيب المخدمات المركزية التي يتم وضع المعلومات والبرمجيات والخدمات الضرورية التي يحتاجها المستثمرون لذلك يجب أن تكون هذه المخدمات ذات قدرة تخزينية ومواصفات فنية مناسبة إضافة إلى سرعة معالجة عالية ، حيث إن أول متطلبات الدخول إلى الإنترنت هو أن يمتلك المستثمر حق الدخول إلى أحد هذه الحواسيب (أي أن يكون له حساب على أحد المخدمات يمكنه من الدخول إلى هذا الحاسوب المرتبط مباشرة بشبكة الإنترنت ويسمح له باستخدام التطبيقات والخدمات المتوفرة) ، فإذا كان المستثمر يعمل في هيئة أو مؤسسة لها شبكتها الخاصة المحلية LAN التي تربط جميع حواسيب المستثمرين ببعضها وتمتلك بعض المخدمات المرتبطة بالإنترنت ، يمكن لذلك المستثمر الطلب من مدير شبكة المؤسسة أن يمنحه السماح في الدخول من خلال حاسوبه إلى ذلك المخدم عبر الشبكة المحلية ومن ثم العبور إلى أي مخدم في شبكة الإنترنت العالمية .

أما إذا كان المستثمر مستقلاً عن أي هيئة أو مؤسسة ، فعليه الاتصال بإحدى الجهات التي تمتلك مخدمات الإنترنت وطلب الاشتراك في استخدام مخدماتها للدخول إلى الإنترنت ، وهناك العديد من الشركات التجارية التي تقدم مثل هذه الخدمات وهو ما يطلق عليها اسم مزودي خدمة الإنترنت ISP إذ يمكن الاشتراك برسم شهري بسيط بحيث تؤمن عنواناً خاصاً بك عبر أحد مخدماتها ، ويمكنك من خلاله التخاطب مع الآخرين وتبادل المعلومات .

العديد من مزودي خدمة إنترنت والخدمة المباشرة يؤجرون مساحات تخزين في أجهزة الحواسيب الخاصة بهم لتمكين المشتركين من إنشاء مواقع لهم على الشبكة العالمية وبمجرد أن تقوم بتصميم (الصفحات) التي قد تضم معلومات عن أعمالك ومنتجاتك وما إلى ذلك سيكون عليك نقلها بكل بساطة لمزود الخدمة عندما تكون متصلاً به ، وتسمى هذه العملية بالتحميل (Uploading) ،

وعندما يزور الناس موقعك في الشبكة العالمية فإنهم يقومون بالإطلاع على ما وضعت من معلومات وتحميلها Download على حواسيبهم بالطريقة نفسها .

٤-١٠-٥- متطلبات الوصول إلى مزود خدمة الإنترنت

قد يكون مزود خدمة الإنترنت في المدينة أو في دولة أخرى مجاورة ، أو في دولة بعيدة جداً كالولايات المتحدة الأمريكية ، فإن كل ما نحتاجه هو جهاز يمكن من خلاله إرسال المعلومات الرقمية لمسافات بعيدة وهو ما يسمى المودم ، ويقوم المودم بالوصل بين الحاسوب الشخصي والمخدم الموجود لدى مزود خدمة الإنترنت ، وللتعامل مع خدمات الإنترنت كالبريد الإلكتروني وغوفر و WWW وغيرها يجب أن تتوفر لدينا البرمجيات اللازمة للتخاطب والاستعراض ، وهي متوفرة بكثرة هذه الأيام وبأسعار زهيدة أو مجانية إذ أن بعض هذه الأنواع توفره شركة Microsoft مجاناً مع نظام التشغيل وهو ما يسمى مستكشف الإنترنت Internet Explorer . كما تنتج شركة Netscape العديد من هذه البرمجيات أيضاً .



٤-١١- خدمات الإنترنت

٤-١١-١ شبكة الأخبار

Network New Transfer Protocol

وهي خدمة تقدم عبر الإنترنت من خلال بروتوكول نقل الإخباريات NNTP ويغطي لوحة الإعلانات هذه مجموعة متنوعة من المواضيع منها : موضوعات الحاسوب ، والسياسة ، والعلوم ، وتصنيف المنتجات ، والعروض التلفزيونية ، والموسيقى وغير ذلك ، وهذه المواضيع قد تكون مفهومة بالنسبة للشخص العادي أو تحتاج إلى تخصص عالي ، ولذلك فإن هذه الخدمة قد تنحصر ببضعة حواسيب أو قد توزع على مواقع حول العالم .

إذا هذه الخدمة تسمح للمرء بالانضمام إلى مجموعات متعددة من الحوار ، وأيضاً تمكنه بما يلي :

- ١- متابعة حوار مفتوح دون أن يشترك به وهذا ما يدعى بالترصد .
- ٢- اختيار مجموعة حوار واحدة أو أكثر تشترك معه بالاهتمام نفسه .
- ٣- إرسال ملاحظة إلى مجموعة الحوار ليقراها الآخرون .
- ٤- إرسال ملاحظة كاستجابة لموضوع كتبه شخص آخر .
- ٥- تسمح بالاستماع إلى محادثة وإلقاء الأسئلة وإبداء بعض التعليقات أحياناً ، لذلك فهي تُعد مصادر معلومات ممتازة وتمثل منبعاً للحوارات الحية وفرصة لاجتماع أشخاص مختلفين لهم اهتمامات مشتركة .

الدخول إلى الخدمة

تعطي كل مجموعة إخبارية اسماً واحداً يدخل المستخدم هذا الاسم عندما يودون الانضمام إلى المجموعة الإخبارية والمساهمة في النقاش الدائر ، وتعطي صيغة المجموعة الإخبارية والمساهمة في النقاش الدائر في ثلاثة أجزاء هي نوع المجموعة والموضوع الأساسي والموضوع الفرعي ، وحتى يتمكن المستخدم من

الاستفادة من هذه الخدمة يجب أن يملك برنامجاً خاصاً يمكنه من تخزين المعلومات على حاسوبه ومن تبادل المعلومات ، ويمكن التواصل مع هذه الخدمة برسالة تشبه البريد الإلكتروني ولكن بـ Form معين يفرضه المخدم وأسوأ ما في هذه الخدمة هو أنها لا تعلن عن وجود معلومات جديدة بل يتعين على المستخدم البحث عنها قبل أن يتلفها المخدم حيث يوجد موعد أقصى لإتلافها ، وهناك ترابط بين E-Mail و NNTP فكثيراً ما تعمم معلومات على بريد المشتركين وبالعكس.

٤-١١-٢ - خدمة الاتصال بوساطة الصوت والصورة

تقدم هذه الخدمة اتصالاً مباشراً لشخص أو لعدة أشخاص عبر الصوت والصورة بدون الحاجة إلى تسجيل المعلومات على ملفات وإرسالها فهي تبث المعلومات الصوتية والمرئية عبر خطوط الاتصال في الإنترنت ، وهذا يتطلب طبعاً أجهزة معينة في طرف المستخدم هي الميكروفون لتوجيه الكلام وجهاز لاستقبال الصوت وكاميرا لتصوير لتسجيل الصور .

٤-١١-٣ - خدمة البريد الإلكتروني E-Mail

يمكن بوساطة الربط مع شبكة الإنترنت إرسال البريد إلى أي شخص في العالم ، وأكثر استخدامات شبكة الإنترنت شيوعاً هو الاتصال مع مستخدمي الشبكة من خلال البريد الإلكتروني E-Mail أو مراسلتهم ، إلا أن أهم عيوب البريد الإلكتروني هو عدم توفر الحماية للمعطيات المرسلة ، أما الحسنات فهي إمكانية استخدام البريد الإلكتروني لإرسال ملفات النصوص والرسوم والصوت ، وأيضاً يمكن إرسال البريد إلى مجموعة أشخاص في الوقت نفسه حتى وإن كان الأشخاص غير موجودين بخلاف التراسل التقليدي بالفاكس ، حيث لا يمكن الاتصال في حال كان جهاز الفاكس لا يعمل أو مقفل .

إذا لا يحتاج البريد الإلكتروني إلى الاتصال المباشر مثل خدمة FTP ، أما البروتوكولات المتحركة على عمل هذه الخدمة هي :
بروتوكول (SMTP) الذي يقوم بالإشراف على انتقال الرسالة إلى المستخدم واستقبال الرسالة لتصل إلى صندوق بريد المستخدم ، أما بروتوكول POP فيقوم بإرسال البريد عند الاتصال من صندوق البريد إلى المستخدم ، وبروتوكول IMAP يتيح إمكانيات متقدمة في التحكم بعملية البريد.
والشرط الوحيد حتى يتم إرسال البريد لشخص ما هو معرفة عنوان بريده الإلكتروني، ومن البرامج الجيدة للتعامل مع البريد الإلكتروني هو برنامج Out Look Express المدمج مع المتصفح IE.4 والإصدارات الأعلى .

□ إرسال رسالة بالبريد الإلكتروني

يتوجب على المستخدم تشغيل تطبيق البريد الإلكتروني على حاسبه المحلي ليتمكن من إرسال بريده الإلكتروني ، ويعمل هذا التطبيق بشكل مشابه لبرامج معالجة النصوص مثل Word ، فيسمح للمستخدم بتشكيل رسالته وتجهيزها وتحديد المرسل إليه بإعطاء عنوان الصندوق البريدي ، وما أن ينتهي المستخدم من إدخال رسالته حتى يرسل تطبيق البريد الإلكتروني الرسالة إلى صندوق بريد المستقبل .

□ الإشعار بوصول البريد الإلكتروني

تعلن برمجيات البريد الإلكتروني في جهاز المستقبل عن وصول البريد بوسائل شتى ، منها إظهار رسالة نصية ، أو إضاءة شكل رسومي على وحدة الإظهار أو إطلاق صوت أو إذاعة رسالة مسجلة ، أو قد تنتظر حتى انتهاء المستخدم من استخدام التطبيق الحالي قبل إصدار التنبيه ، كما تسمح معظم

برمجيات البريد الإلكتروني بإلغاء التتبيه نهائياً ، مما يحتم على المستخدم إجراء مراقبة دورية للصندوق للتحقق من وصول البريد الإلكتروني .

□ قراءة رسالة بريد إلكتروني

يستطيع المستخدم استخراج الرسائل من صندوقه البريدي بالاستعانة ببرنامج التطبيقات المناسب ، ويسمح هذا التطبيق للمستخدم بمشاهدة كل رسالة والإجابة الاختيارية على أي منها ، كما أن التطبيق يعلن عادة عندما يبدأ عن وجود رسائل تنتظر في الصندوق البريدي بشكل خلاصة ابتدائية تحتوي سطرًا واحداً لكل رسالة من الرسائل الواردة ، ويعطي هذا السطر أسم المرسل ، وزمن وصول الرسالة وطولها ، ويستطيع المستخدم الاختيار من هذه الخلاصة ، ويؤدي اختيار أي من هذه الرسائل إلى قيام نظام البريد الإلكتروني بإظهار محتوياتها ، ويتوجب على المستخدم اتخاذ فعل ما بعد قراءة الرسالة ، كإرسال جواب إلى المرسل أو إبقائها في الصندوق لمشاودتها مجدداً أو تخزين نسخة عنها في ملف أو إتلافها .

□ كيف يعمل البريد الإلكتروني

يتشارك برنامجان بالعمل لنقل رسالة البريد الإلكتروني من الحاسوب المرسل إلى صندوق بريد المستقبل (يتطلب النقل وجود برنامجين لأن البرنامج التطبيقي الذي يعمل على الحاسوب الأول لا يستطيع تخزين المعطيات مباشرة في صندوق بريد الحاسوب الآخر ، أي على القرص المغناطيسي للمستقبل) ، ويصبح البرنامج الموجود على حاسوب المرسل عميلاً Client عندما يبعث برسالة البريد الإلكتروني ، ويتصل هذا العميل ببرنامج المخدم Server لدى المستقبل ويسلمه نسخة من الرسالة ، حيث يقوم المخدم بتخزين الرسالة في صندوق بريد المستقبل ، ويعمل برنامج العميل تلقائياً فور انتهاء المستخدم من تشكيل رسالة البريد الإلكتروني ، ويستخدم العميل عنوان البريد الإلكتروني لتحديد المستقبل ويستعمل بروتوكول TCP لإرسال نسخة من رسالة البريد الإلكتروني عبر الإنترنت إلى

المخدم Server ، وما أن يتلقى المخدم الرسالة حتى يخزنها في صندوق المستقبل ويعلمه بوصول بريد إلكتروني ، وبعد التخابط بين العميل والمخدم معقداً لأنه قد يحصل خلل في أي لحظة في الحواسيب أو في الإنترنت التي تربطها (كأن يغلق شخص ما أحد الحاسوبين فجأة بدون قصد) ، ولضمان تسليم البريد الإلكتروني بشكل سليم يحتفظ العميل Client بنسخة عن الرسالة أثناء نقلها ، وبعد أن يعلم المخدم العميل بأنه استلم الرسالة وخزنها على القرص يمحو العميل نسخته الخاصة أو بعبارة أخرى يتخلى عنها .

٤-١١-٤ - خدمة الغوفر Gopher

يعد الغوفر Gopher أداة مستخدمة على نطاق واسع في الإنترنت نستطيع من خلالها القيام باستعراض المعلومات دون أن يكون لدينا المعرفة المسبقة عن مكان هذه المعلومات ، أما آلية عمل برمجيات الغوفر فهي كالتالي :

عندما يقوم مستثمر ما بتنفيذ برنامج الغوفر فإنه يتم إنشاء مخاطبة ما بين برنامج عميل على حاسوب المستثمر ومخدم الغوفر في مكان ما من شبكة الإنترنت ، ويقوم برنامج المخدم من خلالها بعرض قائمة خيارات لبعض المعلومات المتوفرة لديه عن العميل ، وتحدد هذه المعلومات عادة مجموعة من محددات عناوين مختصرة تدل على أماكن تواجد المزيد من المعلومات عن موضوع معين من المواضيع المحددة بالقائمة ، ويقوم المستثمر باختيار أحد المواضيع أما قائمة خيارات غوفر هي:

- ✓ جمل مفيدة باللغة الإنجليزية تشرح نفسها .
- ✓ كل عنصر في القائمة يشير إلى ملف أو قائمة أخرى .
- ✓ عند الاختيار يظهر المحدد .
- ✓ يمكن لعنصر في قائمة أن يشير إلى قائمة غوفر على حاسوب آخر .

٤-١١-٥ - خدمة نقل الملفات File Transfer Protocol

سرعان ما أدرك الباحثون بعد استخدامهم الـ ARPANET إمكانية استخدام الشبكة لنسخ ملف من حاسوب إلى آخر ، فاخترعوا برمجيات تنجز المهمة وأطلقوا عليها نقل ملف File Transfer ، وأعاد الباحثون بعد ذلك كتابة البرمجيات لتعمل عبر الإنترنت وصارت تعرف ببروتوكول نقل ملف File Transfer Protocol ويختصر لها بـ FTP .

يجب على المستخدم تشغيل برنامج FTP للتمكن من استعمال هذه الخدمة على حاسوبه الشخصي ، حيث يعطي برنامج FTP محثاً للمستخدم ويتقبل سلسلة من الأوامر التخاطبية ، ويستجيب الـ FTP خلال جلسة التخاطب إلى كل أمر يدخل المستخدم ثم يطبع المحث عندما يصبح جاهزاً لتلقي الأمر التالي . يستعمل المستخدم FTP بطريقة مشابهة لاستخدام الهاتف ، فبعد تشغيل برنامج FTP يعرف المستخدم الحاسوب البعيد ويطلب من FTP إنشاء وصلة ، ويستعمل FTP برمجيات TCP/IP للاتصال بالحاسوب وفي حال إنشاء الوصلة يتخاطب المستخدم مباشرة مع الحاسوب البعيد فمثلاً يستطيع الحصول على لائحة بالملفات الموجودة في الحاسوب البعيد ويستطيع نسخها وينبغي على المستخدم إنهاء الجلسة عند انتهائه من الأنصال ، وتتطلب كل خطوة من خطوات المخاطبة إدخال المستخدم لأمر من الأوامر عند محث FTP .

٤-١١-٦ - خدمة الدخول إلى حاسوب آخر عبر الشبكة

وهي عبارة عن برنامج يساعد المستخدم أن يتصل بحاسوب آخر واستخدامه كما لو كان المستخدم جالساً أمام ذلك الجهاز الذي قد يكون في اليابان أو استراليا فعلى سبيل المثال يمكن لموظف الاتصال بمكتبه والرد على رسائله والقيام بأعماله الاعتيادية جميعها من مكان إقامته كما لو كان في مكتبه .

يمكننا على سبيل المثال استخدام Telnet لمشاهدة قاعة البطاقات الإلكترونية في مكتبة الكونغرس كما يمكننا استخدامها لاستعراض قواعد معطيات حكومية ، والشرط الوحيد لاستخدام هذه الخدمة بنجاح هو أن نعرف كيف نستخدم الحاسوب الذي دخلنا عليه عن بعد كما يجب أن يسمح لنا هذا الحاسوب بالنفاذ إلى ملفاته . هناك عدد كبير من المؤسسات التي قامت بكتابة برمجيات خاصة ضمن أنظمتها لكي يتمكن مستخدم الإنترنت من النفاذ الفوري إلى المعلومات .

تستمر الإنترنت بالتطور رغم وجود عدد كبير من الخدمات فيها ، و يعمل الباحثون كل يوم على اختراع وتقديم خدمات جديدة فنجد أن الخدمة تنتشر بسرعة فإن لم تلق قبولا تنتشر وتعطي دوراً لغيرها .

والسؤال هو ما الذي يجعل الإنترنت قادرة على دعم عدد كبير من الخدمات المتنوعة بينما تعجز نظم الاتصالات الأخرى عن ذلك ؟

السبب هو أن التقنية الأساسية التي تستخدمها الإنترنت أكثر مرونة من التقنيات الأخرى حيث تستخدم كل الخدمات عبر الشبكة البروتوكول TCP/IP الذي يسمح للحواسيب بتبادل الصور والصوت بسهولة وتبادل رسائل البريد الإلكتروني .

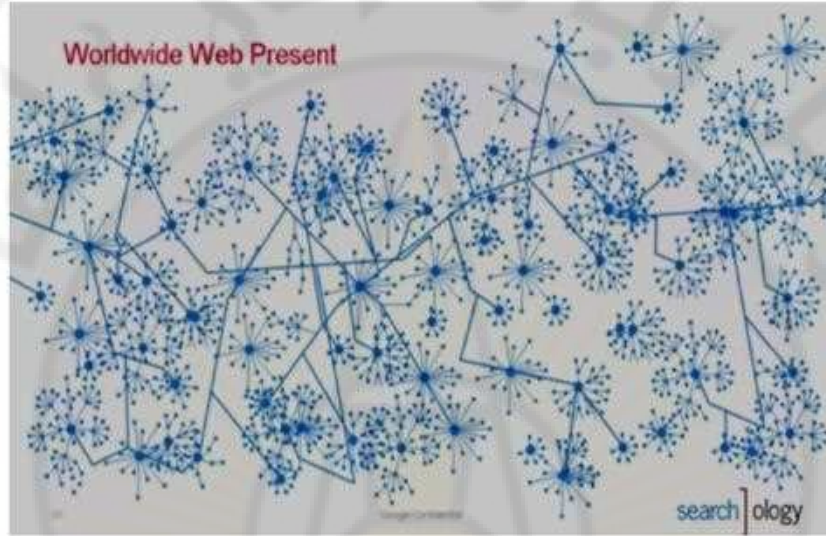
ومن الجدير بالذكر أن معظم الحواسيب والخدمات التي تستخدم بروتوكول TCP/IP تم اختراعها بعد البروتوكول TCP/IP إلا أن تصميم هذا البروتوكول حمل المرونة الكافية لاستيعاب التغيير .

٤-١١-٧- الشبكة العنكبوتية العالمية WWW

World Wide Web

هي من أبداع التقنيات التي تم معرفتها حتى الآن في تاريخ الإنترنت حيث يمكن عبر هذه التقنية حصول المستخدم على معلومات كتابية ومسموعة ومرئية عبر صفحات إلكترونية تمثل كتيباً إلكترونياً يتصفح فيه المستخدم عبر حاسوبه الشخصي ، فما على المستخدم إلا نقل المؤشر إلى الكتابة التي قد تكون موصولة

بصوت أو صورة أو موضوع معين ، وتتضمن WWW مناظر مرسومة وملفات صوتية وملفات سينمائية ، ويبين الشكل (٤-٢٠) الشبكة WWW .



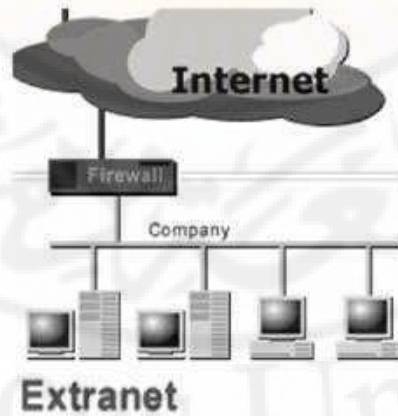
الشكل (٤-٢٠) - يبين الشبكة WWW

تستخدم الشبكة WWW بروتوكول نقل النص الفائق HTTP (Hypertext Transfer Protocol) وهو اللغة المعيارية التي تسمح لعملاء WWW ومقدميه بالاتصال فيما بينهم ، لذلك يمكن أن تسمى مخدمات web في بعض الأحيان باسم بروتوكول نقل النص الفائق فنقول أنها مخدمات HTTP ، وتستخدم جميع مواقع web نظام النص الفائق HTML حيث يتم ربط صفحات معينة بنقاط ارتباط ضمن النص الأصلي ، وعندما يتم التأشير بالفارة على نقطة ارتباط ، نتوجه فوراً إلى صفحات أخرى قد تكون ضمن نفس المخدم أو متوضعة في مخدم آخر عبر الشبكة ، إذ أن نظام WWW يربط جميع المعلومات المتواجدة عبر الإنترنت وكأنها كتاب واحد أو عدة كتب بغض النظر عن مواقعها .

٤-١٢ - شبكة الاكسترا نت Extranet

لنعرف شبكة الإنترنت Intranet هي شبكة حواسيب مقصورة على مؤسسة ما ولكنها تستخدم أسلوب عمل الإنترنت نفسه وغير متصلة بالانترنت. أما تعريف شبكة الاكسترا نت Extranet هي شبكة حواسيب مقصورة على مؤسسة ما ولكنها تستخدم أسلوب عمل الإنترنت نفسه ومتصلة بالانترنت مع وجود حماية تسمى الجدران النارية Firewalls تحمي الشبكة من القرصنة والمتسللين على الشبكات، ويبين الشكل (٤-٢١) شبكة الاكسترا نت Extranet .

إن شبكة الاكسترا نت هي الشبكة التي تربط شبكات الانترنت الخاصة بالشركات والعملاء ومراكز الأبحاث الذين تجمعهم أعمال مشتركة ، فإنها تؤمن لهم تبادل المعلومات والمشاركة فيها مع المحافظة على خصوصية الانترنت المحلية لكل شركة ، ويتم بناء تطبيقات الاكسترا نت في مجالات عديدة منها مثلاً شبكات البنوك والتعاملات المصرفية ، وأيضاً نجدها في نظم التعليم الواسعة النطاق مثل الجامعات ومراكز التدريب الأكاديمية وفي شبكات المواصلات مثل القطارات ، وتوفر شبكات الاكسترا نت الأمن المطلوب داخل الشبكة لتبادل المعلومات الخاصة بين المؤسسات مع بعضها .



الشكل (٤-٢١) - يبين شبكة الاكسترا نت Extranet

تصنف شبكات الاكسرنات إلى ثلاثة أنواع :

١. شبكات اكسترانت التزويد Supplier Exstranets

تربط هذه الشبكات المخازن والمستودعات الخاصة بالبضائع فهي تقوم بالتوفيق بين المخازن الفرعية والمستودعات الرئيسية للتحكم في مستوى المخزن بالنظام ، ولتفادي مشكلات العجز والمحافظة على كميات ثابتة من المخزون .

٢. شبكات اكسترانت التوزيع Distribution Extranet

يُعد هذا النوع من شبكات الاكسترانت أكثر الأنواع تواجداً فهي تقدم خدمة الطلب الالكتروني للعملاء من حجز واشتراكات وغيرها من الخدمات ، وفي ظل خدمات النشر الفوري للتعديلات والتغيرات التي قد تحدث على هذه الخدمات من تغيير أسعار ومواصفات .

٣. شبكات اكسترانت الند Peer Extranet

يكثر استخدام هذا النوع في مجالات الإنتاج الصناعية والغذائية ، فهي تسمح لتبادل الأسعار والمنتجات بين الشركات الصغيرة والشركات الكبيرة للمشاركين في مجال واحد للنهوض بحركة البيع والشراء مما يرفع من مستوى الخدمة في ذلك القطاع .

٤-١٣ - أمن المعلومات

تخزن أجهزة الحاسوب قدراً كبيراً من المعطيات التي تخص المؤسسات الحكومية والخاصة وكذلك الأشخاص فهناك المعطيات العسكرية التي تتمتع بمنتهى السرية ، والمعطيات التي تبين حسابات الشركات وأرباحها وخسائرها وكذلك معلومات شخصية تبين أرقام الهواتف وتوقيعهم وأرقام بطاقتهم المصرفية.

لذلك فإن المؤسسات والأشخاص يرغبون بأن تكون هذه المعلومات الشخصية سرية وعدم السماح للأشخاص غير المخولين بالإطلاع عليها، ويدعى هذا الحق بالخصوصية، ومن أمن المعلومات نذكر :

١ - سرية المعلومات وأمنها

إن سرية المعلومات وأمنها لا تعني فقط الأمان للمعدات وإنما للبرمجيات والمعطيات المخزنة في الحاسوب ، وتوفير الحماية للحواسيب والمعدات من الخراب قدر الإمكان ، وعمل نسخ احتياطية للمعطيات والبرمجيات واتخاذ الإجراءات اللازمة لحمايتها من الفيروسات ، واستخدام كلمات السر للمعدات والبرمجيات ويجب أن تتكون كلمات السر من أحرف وأرقام وأن تتغير من وقت إلى آخر ، وعمل النسخ الاحتياطية التي هي أساسية وضرورية.

٢ - الحواسيب والسرية

يشعر بعض الناس بالخوف من تهديد حقهم في سرية معلوماتهم الشخصية عن طريق سوء استخدام محتمل أو كشف غير مصرح به للمعلومات في قواعد معطيات الحاسوب ، وتحتوي قواعد المعطيات في معظم الأحيان على معلومات خاصة وشخصية، مثل السجلات الطبية والمصرفية والضرائبية ، وبعض قواعد المعطيات الأخرى، تحتفظ بخطط العمل التجاري، أو الاختراعات التي تحرص الشركة على إخفائها من الشركات المنافسة. كما تحتزن قواعد معطيات أخرى ومعلومات عسكرية في غاية السرية، أو معلومات أخرى ذات أهمية لأمن الدولة، وتتحكم القوانين اليوم في الكشف عن المعطيات.

٣ - الحواسيب والأمن

يتم تصميم نظم تشغيل الحواسيب بحيث تمنع الدخول إلى الحواسيب دون تصريح، ولكن على الرغم من ذلك، فإن جرائم الحاسوب قد تقع ، ويستخدم

جواسيس الصناعة واللصوص خطوط الهاتف للوصول إلى داخل الحاسوب، ويقوم بعض هؤلاء المجرمين بسرقة، أو تعديل المعلومات، في قاعدة المعطيات ، ويسرق البعض الآخر المال باستخدام إمكانية الحاسوب في تحويل الأموال كهربائياً من حساب إلى آخر ، ويمكن أن تحدث مشكلات رئيسية إذا تمكن أحدهم من الوصول غير القانوني لمعلومات سرية بقواعد معطيات تابعة للحكومة أو الشركات الكبيرة ، وفي بعض الأحيان، فإن العاملين أنفسهم بمنظمة ما يرتكبون جرائم الحاسوب .

□ عناصر أمن المعلومات

إن أغراض أبحاث واستراتيجيات ووسائل أمن المعلومات (سواء من الناحية التقنية أو الأدائية) وهدف التدابير التشريعية في هذا الحقل، هو ضمان توفر العناصر التالية لأية معلومات يراد توفير الحماية الكافية لها :

- ✓ السرية أو الموثوقية
- ✓ التكاملية وسلامة المحتوى
- ✓ استمرارية توفر المعلومات أو الخدمة
- ✓ عدم إنكار التصرف المرتبط بالمعلومات ممن قام به



٤-١٤ - الفيروسات Viruses

في نهاية الثمانينيات من القرن العشرين الميلادي تكتشف لخبراء الحاسوب خطورة نوع من البرامج أطلق عليه فيروس الحاسوب ، ويُصنّف فيروس الحاسوب ليلحق الأذى في بعض الأحيان بمسح أو تعديل المعلومات، وفي أحيان أخرى بإقحام رسالة ، وفي نهاية المطاف، يدخل فيروس الحاسوب إلى نظام تشغيل الحاسوب ، وينتشر بنسخ نفسه بسرعة فائقة، وهكذا ينشر العدوى في الحواسيب

الأخرى المتصلة بالشبكة ، ويمكن أن تؤدي هذه العملية بسرعة إلى زيادة الأحمال على شبكات الحاسوب الضخمة أو تعطيلها بالكامل.

إن انتشار الفيروسات ينجم غالباً عن عدم استخدام البرمجيات المضادة لها وكذلك الاستخدام غير الواعي لشبكة الانترنت والبريد الالكتروني .

تمثل فيروسات الحاسوب Computer Viruses نوعاً من الجرائم الحاسوبية ، وفيروس الحاسوب عبارة عن برنامج يدخل إلى الحاسوب ليهدم أو يشوه المعطيات والبرامج المخزنة داخله وينتقل فيروس الحاسوب إلى حواسيب أخرى عن طريق شبكات الحواسيب Computer Network واستخدام الأقراص الليزرية الملوثة بالفيروسات.

□ أنواع الفيروسات

أهم أنواع الفيروسات التي ظهرت هي :

▪ الفيروسات الدودية Worms

الفيروس الدودي عبارة عن برنامج يعطي أوامر خاطئة أو مضللة لجهاز الحاسوب ويحتل هذا النوع من الفيروسات الذاكرة الرئيسية وينتشر بسرعة فائقة جداً في الشبكات.

▪ القنابل الموقوتة Time Bombs

فيروس القنبلة الموقوتة عبارة عن برنامج يقوم بتفجير نفسه في وقت محدد أو بعد تنفيذ عدة مرات ويستخدم هذا النوع من قبل شركات الحاسوب التي تعطي نسخاً مجانية على أمل شراء النسخة الأصلية لاحقاً وإذا لم يتم المستخدم بشراء النسخة الأصلية يقوم البرنامج بتفجير نفسه.

■ أحصنة طروادة Trojan Horses

حصان طروادة عبارة عن برنامج يدخل الحاسوب بشكل شرعي يصعب كشفه وبعد تركيبه داخل الجهاز يقوم بتدمير وتشويه المعطيات والفهارس المهمة وكثير من حصون طروادة تنتقل عبر البريد الإلكتروني.

□ تعريف فيروسات

هي برمجيات صغيرة تختفي في موضع ما على الأقراص الخاصة (الأقراص المرنة أو القرص الصلب) ، وبعد فترة ينشط هذا الفيروس لينتشر بسرعة في معدات أو برمجيات الحاسوب ليقيم بتعطيلها . وإذا لم نستخدم برنامج مضاد للفيروسات، فسنعرف بوجود فيروس في الحاسوب عندما ينشط هذا الفيروس ، ويتم تنشيط الفيروسات المختلفة بطرق مختلفة.

والسؤال كيف يصيب الفيروس الحاسوب؟

تختفي الفيروسات على أي قرص وعندما تصل إلى القرص المناسب (قرص مرن أو قرص صلب آخر عبر شبكة)، يبدأ تنشيط برنامج الفيروس ويصيب الحاسوب الذي نعمل عليه.

إن أسوأ شيء في الفيروسات هو أنها يمكن أن تنتشر من حاسوب لآخر (من خلال استخدام قرص فيه فيروس) أو عبر شبكة من الحواسيب، بما فيها الإنترنت.

□ حماية الحاسوب من مخاطر الفيروسات

هناك عدد كبير من البرمجيات المضادة للفيروسات ، ولكنها ليست مجانية بالطبع . إن أهم شيء بالنسبة للبرنامج المضاد للفيروسات الذي سنستخدمه هو أن نقوم بتحديثه بشكل دائم ، وتقدم الكثير من الشركات أقراص تحديث بشكل منتظم أو تتيح لنا الحصول على نسخة محدثة من خلال لوحة إعلانات إلكترونية أو عبر الإنترنت.

الجزء الثاني
C++ البرمجة بلغة

الفصل الخامس
مبادئ الخوارزميات



مبادئ الخوارزميات

٥-١- مقدمة

رغم أن الحاسوب يتميز بقدرته على إنجاز العمليات الحسابية حسب الأوامر والتعليمات المعطاة له بسرعة فائقة و بدقة متناهية و كذلك بإمكانياته الكبيرة في حفظ المعلومات الواسعة والمختلفة التي يعجز الإنسان عن حفظها واستعادتها باستخدام ذاكرته العادية، فهو يعجز من أن يقوم بشكل ذاتي بحل أي مسألة مهما كانت بسيطة، أي أن عمله ينحصر في إنجاز الحلول للمسائل التي تبرمج له بشكل صحيح يتوافق مع الأسس العلمية الصحيحة التي تعتمد عليها هذه الحلول. لذا سوف نستعرض في هذا الفصل الخطوات الضرورية اللازمة لحل المسائل باستخدام الحاسوب وكذلك توضيحاً مفصلاً لمفهوم الخوارزميات و المخططات التدفقية التي توضح خطوات الحل والتي تشكل العنصر الأساسي لكيفية البرمجة.

الخوارزمية Algorithm مفهوم قديم يعود إلى مطلع القرن التاسع الميلادي (568- 668 هـ)، في أوج الدولة العباسية زمن الخليفة المأمون، ويعود الفضل في إيجاد هذا المفهوم الرياضي إلى العالم الجليل محمد بن موسى الخوارزمي الذي عاش في بغداد وهو من مدينة خوارزم ، وكلمة خوارزم (algorism) في الأصل كانت مقتصرة على القوانين الرياضية التي تستخدم الأرقام العربية و طُورت في اللاتينية من الخوارزمي (al-Khwarizmi) لتصبح (algorithm) في القرن الثامن عشر الميلادي لتشمل جميع إجراءات حل المشكلات و تنفيذ المهمات.

بقي الحساب العشري وجداول الضرب والقسمة تعرف باسم الخوارزميات والألواح الخوارزمية لعدة قرون في أوربا ، لكن هذا المصطلح تطور مع الزمن، ليرتبط مؤخراً ارتباطاً وثيقاً جداً ببرمجة الحواسيب الإلكترونية.

٥-٢- تعريف الخوارزمية

يمكن تعريف الخوارزمية بعدة أشكال نذكر منها:

١. الخوارزمية هي مجموعة من الخطوات المتسلسلة والمحددة والمنتهية التي تؤدي في حال تنفيذها إلى حل قضية معينة والوصول إلى نتائجها.
 ٢. الخوارزمية هي طريقة الوصول إلى هدف معين.
 ٣. الخوارزمية هي مجموعة متتالية من العمليات المعرفة والمنتهية اللازمة لإنجاز عمل ما، أو لحل مسألة ما والحصول على نتيجة صحيحة.
- تعالج الخوارزمية معطيات مُدخلة في معظم الحالات، والمعطيات المعالجة لا تقتصر على الأعداد والأرقام، بل تشمل الرموز والنصوص والرسوم والصور والأصوات كمدخلات ومُخرجات، ويتزايد الاهتمام بالخوارزميات بشدة مع ظهور الحواسيب لضرورة استخدامها في حل المسائل في جميع المجالات العلمية والتقنية والاجتماعية والاقتصادية والصناعية والتجارية بواسطة الحاسوب.

٥-٣- خطوات حل مسألة باستخدام الحاسوب

عند حل أي مسألة باستخدام الحاسوب لا بد من إتباع مجموعة من الخطوات سنقوم بإيجازها فيما يلي:

١. تعريف وتحليل المسألة: إن تعريف المسألة هو عبارة عن دقة التعبير في تطبيق المسألة بحيث يجعلها معروفة ومفهومة بصورة واضحة وبدون أي غموض لجميع الأشخاص العاملين ضمن مجال الاختصاص الذي تخضع له المسألة، أما تحليل المسألة ووضع طريقة الحل فهو أصعب المصاعب و

أشق الخطوات، و من أجل الوصول إلى حل صحيح فإن كثيراً من القوانين والطرق الرياضية المناسبة لحل المسألة يجب أن تستعمل و ربما تحتاج أيضاً إلى تطوير هذه القوانين والطرق لنجعلها تناسب الحل في كثير من الأحيان ففي هذه الخطوة يجب تحديد:

- ✓ طبيعة المخرجات (النتائج) و تنظيم كتابتها.
- ✓ المدخلات (البيانات أو المعلومات) و تحديد نوعها و تنظيم إدخالها إلى الحاسوب.
- ✓ طرق الحل المناسبة و تقييمها بما يتلاءم مع كيفية تنفيذها بالحاسوب و في ضوء ذلك يتم اختيار الحل الأفضل.

٢. **برمجة الحل خطياً:** بعد اختيار طريقة الحل المثالية و تحديد كل ما تشمله من علاقات رياضية، يتم التعبير عنها على شكل خطوات متسلسلة ومتراصة منطقياً، ودقيقة الوصف تؤدي إلى الوصول إلى حل المسألة ، و هذه الخطوات المتسلسلة تعرف بخوارزمية المسألة Algorithm of the Problem ، و يمكن تمثيل هذه الخوارزمية بعد إيضاح جميع التعليمات والأوامر المتسلسلة التي يراد تنفيذها في كل خطوة بمخطط وصفي تسلسلي يدعى بالمخطط التدفقي Flowchart وذلك باستخدام مجموعة من الأشكال الاصطلاحية الرمزية.

٣. **برمجة الحل باستخدام إحدى لغات البرمجة:** إن المخطط التدفقي هو عبارة عن تخطيط تصوري (بياني) مساعد وسهل الملاحظة بالنسبة للمبرمج و لكنه غير مفهوم عند الحاسوب، لذلك فإن طريقة الحل الممثلة بالمخطط التدفقي يجب أن تكتب بإحدى لغات البرمجة التي يفهمها الحاسوب و التي تتلاءم مع حل المسألة، و يلي ذلك كتابة البرنامج على الوسط الخارجي المناسب و إدخال البرنامج إلى الحاسوب ، و البرنامج الناتج من هذه الخطوة و المكتوب

بإحدى اللغات الرمزية أو الراقية يسمى بالبرنامج المصدري source program.

٤. **ترجمة البرنامج المصدري:** بعد الانتهاء من كتابة البرنامج المصدري يتعين إدخاله إلى الحاسوب للتأكد من صحة كتابته من جهة، ثم لترجمته إلى لغة الآلة بواسطة برنامج الترجمة الخاص في حالة عدم وجود أخطاء في البرنامج المصدري الذي يسمى بالترجم compiler ، و تمر عملية الترجمة في المراحل الآتية:

✓ **مرحلة التحليل المعجمي Lexical analysis:** في هذه المرحلة يتم مطابقة مفردات برنامج المصدر والعلاقات و الأسماء مع تلك المسموح بها في اللغة و اكتشاف أي أخطاء فيها.

✓ **مرحلة التحليل اللغوي والنحوي Syntax analysis:** في هذه المرحلة تجري عملية مطابقة تعليمات البرنامج المصدري مع القواعد اللغوية المستخدمة، و اكتشاف أي أخطاء فيها، بالإضافة إلى عملية تحويل البرنامج المصدري إلى تعليمات و أوامر رمزية بلغة التجميع.

✓ **مرحلة ترجمة البرنامج إلى لغة الآلة:** في هذه المرحلة نحصل على البرنامج الهدف object program و الذي بموجبه يمكن البدء في عملية التنفيذ.

٥. **تجربة البرنامج و تنفيذه:** بعد الحصول على البرنامج الهدف، تتم تجربته للتأكد من صحته منطقياً وذلك باستخدام عينة من المعطيات الاختبارية Test Data، فإذا ثبت صحة طريقة الحل بمطابقة النتائج الخارجة من الحاسوب مع النتائج التي تم الحصول عليها يدوياً على سبيل المثال، يمكن تنفيذ البرنامج على المعطيات الحقيقية.

٥-٤- أنواع الخوارزميات

لا بد من الإشارة إلى أننا نستخدم مفهوم الخوارزمية في حياتنا اليومية من حيث لا ندري، حيث إن أي عمل نقوم به وننفذه بالشكل الصحيح لا بد أن يخضع لخطة عمل محددة ، وبوجه عام يمكن تقسيم الخوارزميات إلى خوارزميات حسابية وخوارزميات غير حسابية.

٥-٤-١- الخوارزميات غير الحسابية

من خلال التسمية نلاحظ أن هذا النوع من الخوارزميات يكون بعيداً كل البعد عن العمليات الحسابية وربما كانت هذه الخوارزميات أكثر الخوارزميات استخداماً ونذكر منها:

- ✓ خوارزميات معالجة النصوص.
 - ✓ خوارزميات تخزين المعلومات واستعادتها.
 - ✓ خوارزميات إدارة قواعد البيانات والمساعدة في اتخاذ القرار في جميع نواحي الحياة.
 - ✓ خوارزمية التدقيق الإملائي.
- وفيما يلي سنطرح بعض الخوارزميات غير الحسابية ومن حياتنا اليومية:

مثال (5-1): خوارزمية طرح سؤال ضمن المحاضرة.

- خوارزمية طرح السؤال هي مجموعة الخطوات الواجب إتباعها للاستفسار عن قضية معينة ضمن المحاضرة، وهذه الخطوات يمكن سردها كما يلي:
١. الانتظار حتى يصل المحاضر إلى نهاية المقطع الكلامي.
 ٢. رفع اليد حتى يسمح للطالب بالكلام.
 ٣. خفض اليد والمباشرة في طرح السؤال.

٤. الاستماع للإجابة حتى النهاية.

تؤدي هذه الخطوات مجتمعة ومرتببة إلى طريقة سليمة لطرح السؤال.

مثال (5-2): خوارزمية عبور شارع ذي اتجاهين من قبل إنسان عادي.

المسألة هي عبور الشارع بسلامة ولتحقيق هذا الهدف لا بد من إتباع مجموعة من الخطوات التالية:

١. إذا كانت الإشارة خضراء، عندئذ لا تقطع الشارع.
٢. إذا كانت الإشارة حمراء أقطع الشارع إلى منتصفه، ناظراً إلى السيارات من اليسار.
٣. إذا كانت الإشارة خضراء للنصف الثاني من الشارع، عندئذ لا تقطع الشارع.
٤. إذا كانت الإشارة حمراء للنصف الثاني من الشارع، عندئذ أقطع الشارع ناظراً إلى السيارات من اليمين.

من الواضح أن هذه الخوارزمية صالحة لكي يقطع الشارع أي إنسان عادي، وإن تطبيقها على هذا النحو، يضمن حياة العابرين من خطر السيارات.

٥-٤-٢ - الخوارزميات الحسابية

أطلق اسم الخوارزميات الحسابية على تلك الخوارزميات التي تتعامل مع المقادير الرياضية، وفيما يلي سنطرح بعض الخوارزميات الحسابية:

مثال (3-5): خوارزمية حساب وطباعة قيمة الكسر المعرف بالشكل

$$Y = \frac{4x + 6}{6x - 8}$$

من الواضح أن الحل بسيط جداً، ويستطيع انجازه أي شخص لديه إلمام بسيط بالعمليات الحسابية ولا يحتاج إلا إلى معرفة قيمة المتغير x ، ولكن الغاية من هذا المثال هو كيف نصل إلى المطلوب من خلال إتباع مجموعة من الخطوات المتسلسلة والمحددة والتي يمكن صياغتها كما يلي:

١. إدخال (معرفة) قيمة للمتغير x .
٢. حساب قيمة البسط $4x+6$.
٣. حساب قيمة المقام $6x-8$.
٤. قسمة البسط $4x+6$ على المقام $6x-8$ وإسناد الناتج إلى Y .
٥. طباعة قيمة Y .

لكي يصبح أي تعبير رياضي خوارزمية لا بد أن يقترن بتسلسل تنفيذ عملياته، أي لا بد من إضافة بعض الشروط والقواعد مثل: البدء دوماً وفق أفضليات تحدد بحسب نوع المسألة المطروحة، وجعل هذه الأفضلية قواعد للتنفيذ تمكننا من الوصول إلى خوارزمية صالحة للتنفيذ.

٥-٥- أولوية تنفيذ العمليات الحسابية والمنطقية

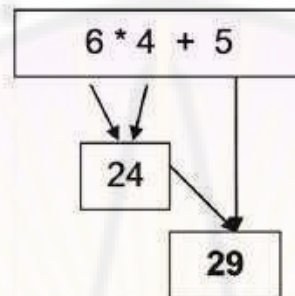
إن أولوية تنفيذ العمليات الحسابية والمنطقية كما يجريها الحاسوب موضحة في الجدول (٥-١).

العملية	رمز العملية	رقم الأولوية	جهة التنفيذ عند التساوي	ملاحظات
فك الأقواس	()	1	من اليسار	
الزيادة أو النقصان	++ أو --	2	من اليمين	رمز برمجي C++
العمليات الحسابية	*/%	3	من اليسار	
العمليات الحسابية	+ -	4	من اليسار	
عمليات المقارنة	< <= > >=	5	من اليسار	
عمليات المقارنة	== !=	6	من اليسار	رمز برمجي C++
العمليات المنطقية	&&	7	من اليسار	رمز برمجي C++
العمليات الشرطية	? :	8	من اليمين	رمز برمجي C++
عمليات الإسناد	= += -= *= /= %=	9	من اليمين	

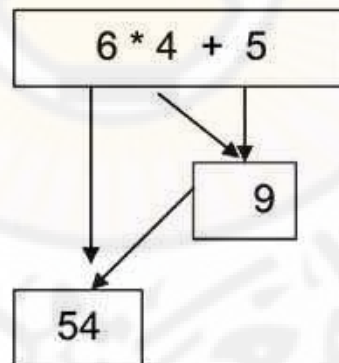
الجدول (١-٥) - يبين جدول أولوية تنفيذ العمليات الحسابية والمنطقية

مثال (5-1)

يبين هذا المثال أولوية تنفيذ بعض العمليات الحسابية على مقدار رياضي كما يلي:



ويعد هذا الجواب صحيح تماماً لأننا طبقنا أولوية التنفيذ ، وإذا لم يكن هناك أولوية في التنفيذ فقد نحصل على الجواب التالي:



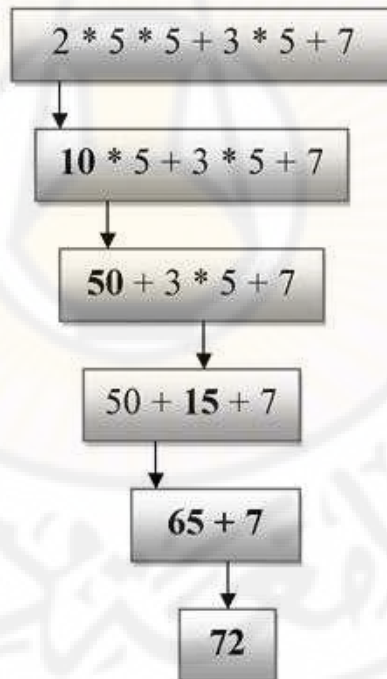
ويعد هذا الجواب خطأ في التنفيذ والقيمة.

مثال (٥-٥)

ما هو الخرج الناتج عند حساب قيمة التعبير الرياضي التالي:

$$2 * 5 * 5 + 3 * 5 + 7$$

فيما يلي سنوضح كيفية حساب قيمة التعبير المعطى أعلاه حسب أولوية تنفيذ العمليات الواردة في الجدول (٥-١).



٥-٦- طرق صياغة الخوارزميات

- الخوارزمية هي مجموعة قواعد وقوانين مكتوبة ومحددة مسبقاً ، تستعمل لوصف الطرق الرياضية والخطوات المنطقية اللازمة لتنفيذ حل المسائل والمشكلات بواسطة الحاسوب، ومن أهم خواص الخوارزميات ما يلي :
١. يجب أن تكون الخوارزمية واضحة ومحددة ويمكن تقسيمها إلى خطوات معينة ومتتالية تؤدي إلى نفس النتيجة .
 ٢. أن تكون الخوارزمية واحدة مهما اختلفت أساليب المعالجة.
 ٣. أن تكون الخوارزمية صالحة لحل جميع المسائل من نفس النوع أو الطراز .
 ٤. إن تصميم الخوارزمية يتطلب مهارة علمية وعملية عالية .
- ويمكن صياغة الخوارزميات بطرق عديدة تتفاوت فيما بينها من حيث دقة التعبير وسهولة الفهم ومن أهمها :

٥-٦-١- الطريقة اللغوية Language Method

تستخدم الطريقة اللغوية لصياغة الخوارزميات ، وفيها يتم استخدام اللغة المحكية (العربية، الإنكليزية،....) من أجل التعبير عن خطوات الحل ويجري تنفيذ تعليمات الخوارزمية بالتسلسل وفق ورودها في نص الخوارزمية في خطوات متسلسلة محددة ومعرفة تحدد سياق هذا التنفيذ، وهذه الطريقة تستخدم على نطاق واسع في صياغة بعض أنواع الخوارزميات الموجهة للجمهور كتعليمات تشغيل الأجهزة واستخدامها ، وفي وصفات تحضير أطباق الطعام وغيرها.

مثال (5-6): خوارزمية حساب وطباعة مساحة المستطيل.

١. البداية (بداية الخوارزمية).
٢. إدخال الطول والعرض.
٣. حساب المساحة وفق الدستور: المساحة = الطول $\times$ العرض
٤. طباعة مساحة المستطيل.
٥. النهاية (نهاية الخوارزمية) .

٥-٦-٢ - الطريقة الترميزية Pseudo code Method

تعتمد الطريقة الترميزية على قواعد محددة يمكن أن تكون مستنتجة من المفاهيم الرياضية، وفي هذه الطريقة يتم استخدام الرموز الرياضية بدلاً من استخدام الكلام المحكي للتعبير عن خطوات الحل ، وتعد هذه الطريقة الأقرب إلى لغات البرمجة وتسمى عند البعض " طريقة شبه البرنامج " ، ويوضح المثال (5-7) هذه الطريقة .

مثال (5-7): خوارزمية حساب وطباعة مساحة المستطيل.

- 1.Start
2. Input x ,y
3. Let s = x*y
- 4.Print s ,x,y
- 5.Stop

مثال (5-8): خوارزمية حساب وطباعة قيمة الكسر الوارد في المثال (5-3) .

- 1.Start
2. Input x
3. Let $A=4x+6$
4. Let $B=6x-8$
5. Let $Y=A/B$
6. Print Y
7. Stop

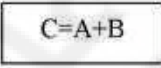
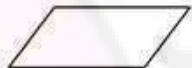
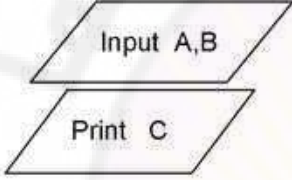
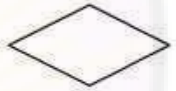
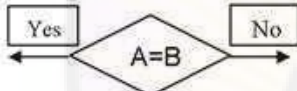
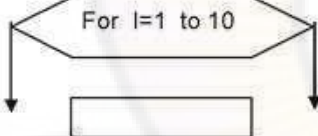
٥-٦-٣- الطريقة البيانية (طريقة المخططات التدفقية)

Flowchart Method

تُستخدم الطريقة البيانية في صياغة الخوارزميات وفيها يتم استخدام أشكال هندسية خاصة وأسهم تصل بينها، للتعبير عن خطوات تنفيذ الخوارزمية ، إضافة إلى عبارات باللغة الطبيعية أو تعابير رياضية ومنطقية ، وتُعد هذه الطريقة أكثر الطرق انتشاراً واستخداماً في توصيف الخوارزميات.

من المعلوم أن الخوارزمية هي عبارة عن مجموعة من الخطوات المتسلسلة التي تصف بصورة مضبوطة وبدون أي غموض جميع الخطوات الرياضية والمنطقية اللازمة لحل مسألة ما، ولكن هذا الوصف في كثير من الأحيان يكون معقداً وصعب الملاحظة والتتبع لذلك فإن المخطط التدفقي الذي يمثل وصفاً تصويرياً لخطوات الخوارزمية يكون أكثر وضوحاً ، ويقوم المخطط التدفقي مقام الخوارزمية ويمكن بواسطته ملاحظة تتبع التسلسل المنطقي لحل المسألة بكل سهولة، وغالباً ما يكون استخراج الخوارزمية من المخطط التدفقي أسهل بكثير من كتابة الخوارزمية مباشرة ، ويبين الشكل (٥-١) أهم الرموز الاصطلاحية

التي يمكن اعتمادها في توصيف العمليات عند استخدام المخططات التدفقية .

الرمز	العملية التي يمثلها	مثال
	البداية والنهاية Start and End	
	عملية الإسناد (العمليات الحسابية) Process	
	إدخال البيانات وإخراج المعلومات Input/Output Data	
	اتخاذ القرار Decision	
	حلقة التكرار Loop	
	اتجاه سير العمليات Flow line	

الشكل (٥-١) - يبين بعض الأشكال الهندسية الخاصة

٥-٧- فوائد استخدام المخططات التدفقية

جاءت المخططات التدفقية كضرورة لتسهيل عمل المبرمج عندما تصبح الخوارزمية أكثر تعقيداً، أي عندما تزداد خطواتها والمقارنات المنطقية فيها، والمخطط التدفقي هو تمثيل بياني (رسومي) للخوارزمية الذي يعطينا تمثيلاً جيداً

لها ويستخدم في هذه المخططات رموز وأشكال هندسية متنوعة لها دلالات محددة، وفيما يلي سنذكر بعض فوائد استخدام المخططات التدفقية:

١. المخططات التدفقية تمكن المبرمج من الإلمام الكامل بالمسألة المراد حلها والتحكم بكل أجزائها بحيث تساعد على اكتشاف الأخطاء المنطقية (Logic Error) و التي تُعد من أهم الأخطاء التي تجهد المبرمج.
٢. المخططات التدفقية تساعد بيسر و سهولة على تعديل البرامج الموضوعة بمجرد النظر إليها.
٣. يُعد الاحتفاظ بالمخططات التدفقية لحلول مسائل معينة أمراً مهماً إذ يكون مرجعاً عند إجراء تعديلات عليها أو استخدامها لحل مسائل أخرى مشابهة دون الحاجة للرجوع إلى المبرمج الأول باعتبار أن الحلول الأولى قد صيغت في خطوات واضحة بسيطة و مفهومة.
٤. المخططات التدفقية توفر وسيلة مناسبة ومساعدة في كتابة البرامج ذات التفرعات الكثيرة.

٥-٨- أنواع المخططات التدفقية

بعد أن تعرفنا على مفهوم المخططات التدفقية وفوائد استخدامها، يمكننا الآن تصنيفها إلى عدة أنواع هي التالية:

❖ المخططات التدفقية المتتابعة البسيطة (Simple sequential Flowchart) .

❖ المخططات التدفقية التفرعية (Branched Flowchart).

❖ المخططات التدفقية الحلقية (Loop Flowchart).

ويمكن للبرنامج الواحد أن يشتمل على أكثر من نوع واحد من هذه الأنواع .

٥-٨-١ - المخطط التدفقي التتابعي البسيط

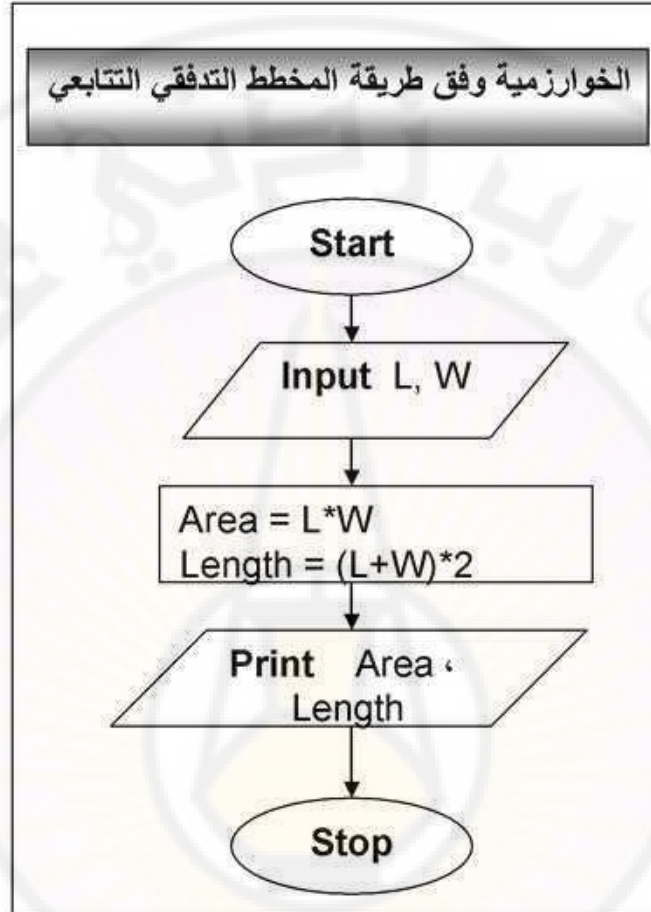
يستخدم هذا النوع من المخططات في المسائل التي يقتضي حلها تتالياً محدداً للخطوات التي تؤدي إلى النتيجة دون الحاجة إلى تغيير سياق التنفيذ ، وتنفذ هذه الخطوات المحددة والمعرفة خطوة خطوة حتى الوصول إلى النهاية، دون تجاهل أو تكرار لأي من الخطوات الموجودة، أي أن هذا النوع من المخططات يخلو من التفرعات والتكرارات أو اتخاذ القرارات، والأمثلة التالية توضح لنا كيفية استخدام المخططات التدفقية التتابعية.

مثال (5-9)

ارسم المخطط التدفقي اللازم لحساب وطباعة مساحة ومحيط مستطيل طوله L وعرضه W ، ثم اكتب الخوارزمية بطريقة Pseudo code .
يبين الشكل (٥-٢) المخطط التدفقي اللازم لحساب وطباعة مساحة ومحيط المستطيل ، أما الخوارزمية وفق طريقة Pseudo code كما يلي :

- 1.Start
2. Input L, W
3. $Area=L*W$
4. $Length=(L+W)*2$
- 5.Print Area, Length
- 6.Stop

الخوارزمية وفق طريقة المخطط التدفقي التتابعي

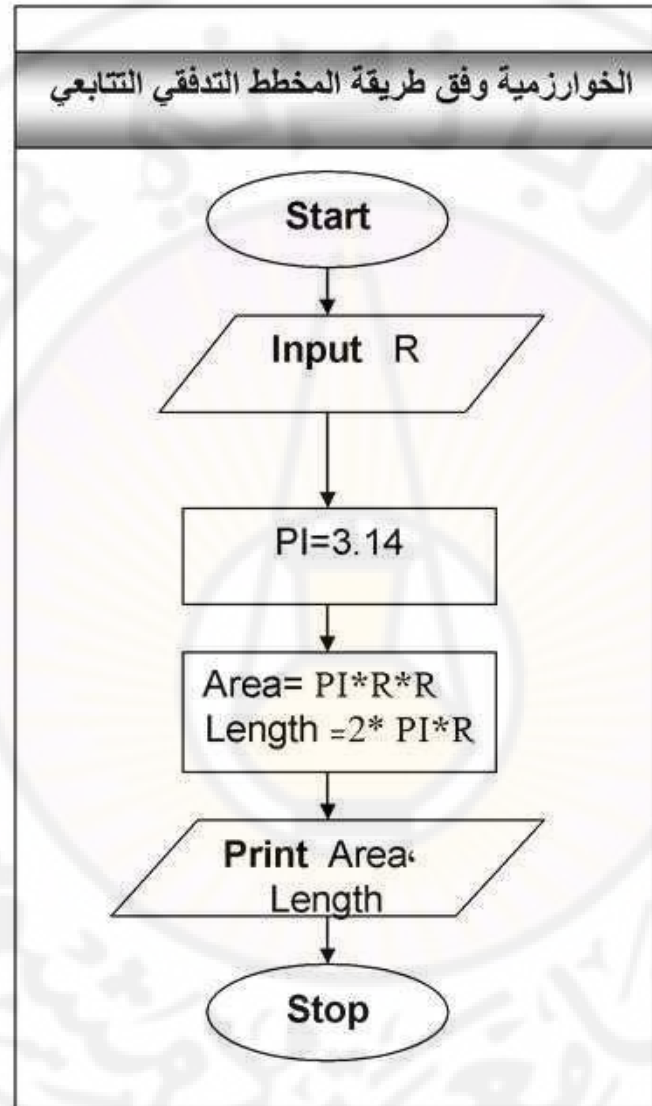


الشكل (٢-٥) - يبين المخطط التدفقي التتابعي للمثال (٥-٩)

مثال (٥-١٠)

ارسم المخطط التدفقي اللازم لحساب وطباعة مساحة ومحيط دائرة نصف قطرها R، ثم اكتب الخوارزمية بطريقة Pseudo code .

يبين الشكل (٣-٥) الخوارزمية وفق المخطط التدفقي .



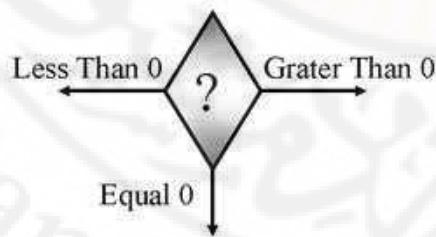
الشكل (٣-٥) - يبين الخوارزمية وفق المخطط التدفقي للمثال (5- 10)

الخوارزمية وفق طريقة Pseudo code

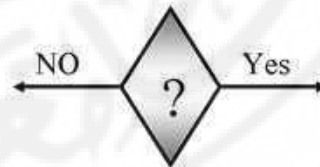
- 1.Start
2. Input R
- 3.PI=3.14
4. Area=PI*R*R
5. Length=2* PI*R
- 6.Print Area, Length
- 7.Stop

٥-٨-٢ - المخطط التدفقي التفرعي

يستخدم هذا النوع من المخططات في المسائل التي تخضع لشروط تحدد التالي المناسب للخطوات المطلوب تنفيذها، حيث يفرض تحقق الشرط أو عدمه ، الاختيار بين طريقتين أو عدة طرق ، وبمعنى آخر فإن هذا النوع من المخططات يتضمن اتخاذ القرارات أو المفاضلة بين خيارين أو أكثر وهناك أسلوبان في تنفيذ القرار هما: قرار ذو تفرعين و قرار ذو ثلاثة تفرعات ، ويبين الشكل (٥-٤) أساليب تنفيذ القرار .



قرار ذو ثلاثة تفرعات



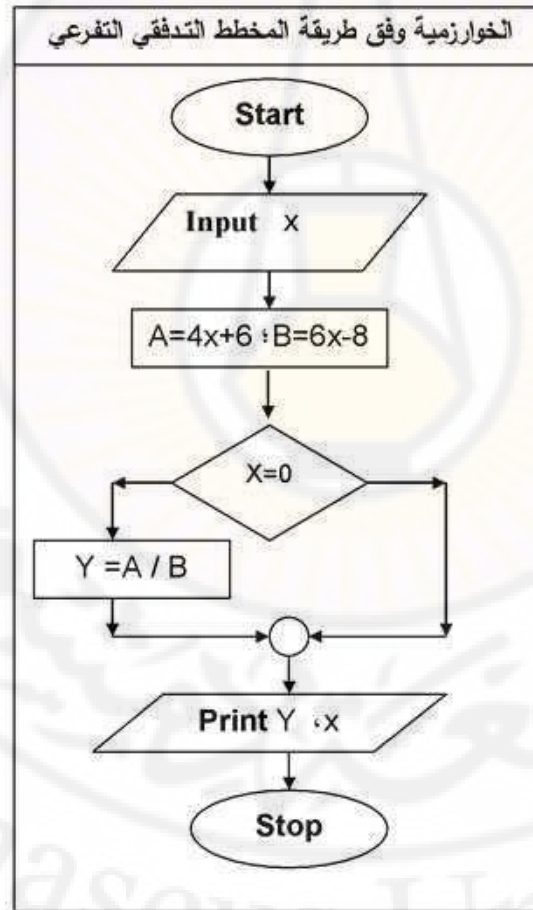
قرار ذو تفرعين

الشكل (٥-٤) - يبين أساليب تنفيذ القرار

ويقدم المثال (5 - 11) إحدى الخوارزميات التي يمكن توصيفها بالمخططات التدفقية التفرعية .

مثال (5-11)

ارسم المخطط التدفقي اللازم لحساب وطباعة قيمة الدالة الرياضية المعرفة كما يلي : $Y = \frac{4x + 6}{6x - 8}$ ، وذلك من أجل قيمة وحيدة للمتغير x ، ثم اكتب الخوارزمية بطريقة Pseudo code .

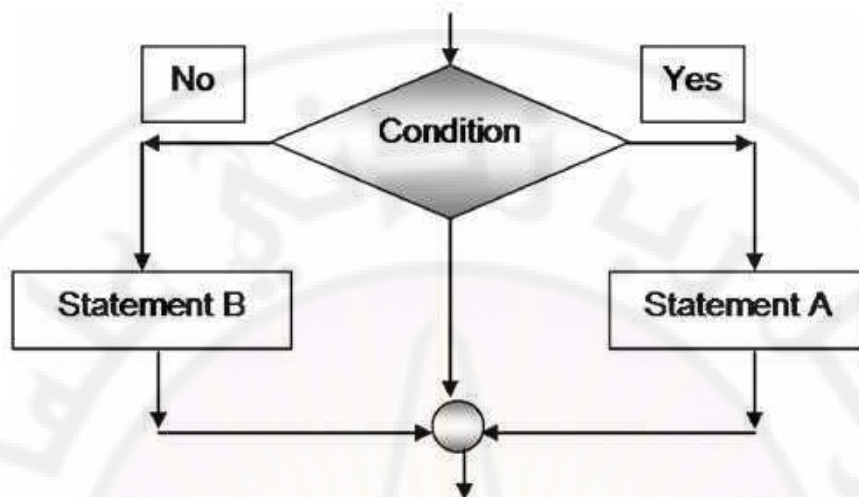


الشكل (5-5) - يبين المخطط التدفقي التفرعي للمثال (5-11)

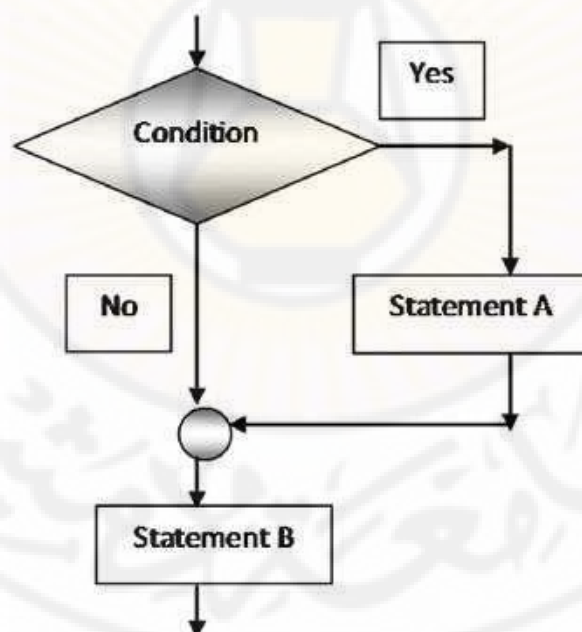
الخوارزمية وفق طريقة Pseudo code

- 1.Start
2. Input x
3. $A=4x+6$
4. $B=6x-8$
5. if $B=0$ then go to 2
6. $Y=A/B$
7. Print Y, x
8. Stop

وبشكل عام فإن المخطط التفرعي يمكن أن يأخذ إحدى الصورتين الآتيتين كما هو موضح في الشكل (٥-٦) . يمكننا ملاحظة أن الشكل (٥ - ٦ - أ) يبين أنه إذا كان جواب الشرط (Condition) YSE فإن التعليمة التالية في التنفيذ تكون Statement A ، أما إذا كان الجواب NO فإن التعليمة التالية تكون Statement B ، كما يمكننا أن نلاحظ في الشكل (٥ - ٦ - ب) أنه إذا كان جواب الشرط YSE فإن التعليمة التالية في التنفيذ تكون Statement A ثم يتبعها Statement B ، أما إذا كان جواب الشرط NO فإن التعليمة التالية تكون Statement B مباشرة.



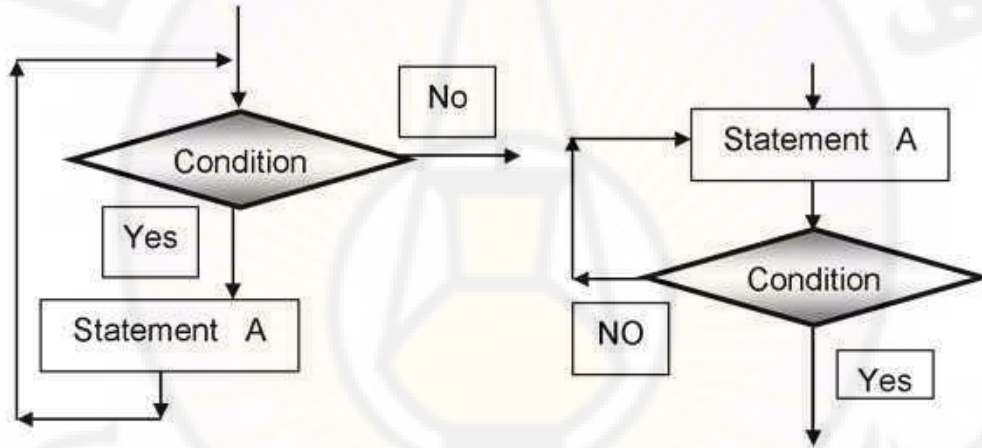
الشكل (٥-٦-أ) - يبين أساليب التفرع



الشكل (٥-٦-ب) - يبين أساليب التفرع

٥-٨-٣- المخطط التدفقي الحلقي

يستخدم هذا النوع من المخططات في المسائل التي يتضمن حلها تكرار تعليمة واحدة أو مجموعة تعليمات، أكثر من مرة، حيث يحدد الشرط الذي يقرر عدد المرات التي يجب تكرارها بواسطة صناديق الاختيار المرتبطة بزيادة أو إنقاص متحول يسعى إلى تحقيق الشرط المحدد في صندوق الاختيار، وبشكل عام فإن المخطط الحلقي يمكن أن يأخذ إحدى الشكلين الآتيين كما هو موضح في الشكل (٧-٥) .

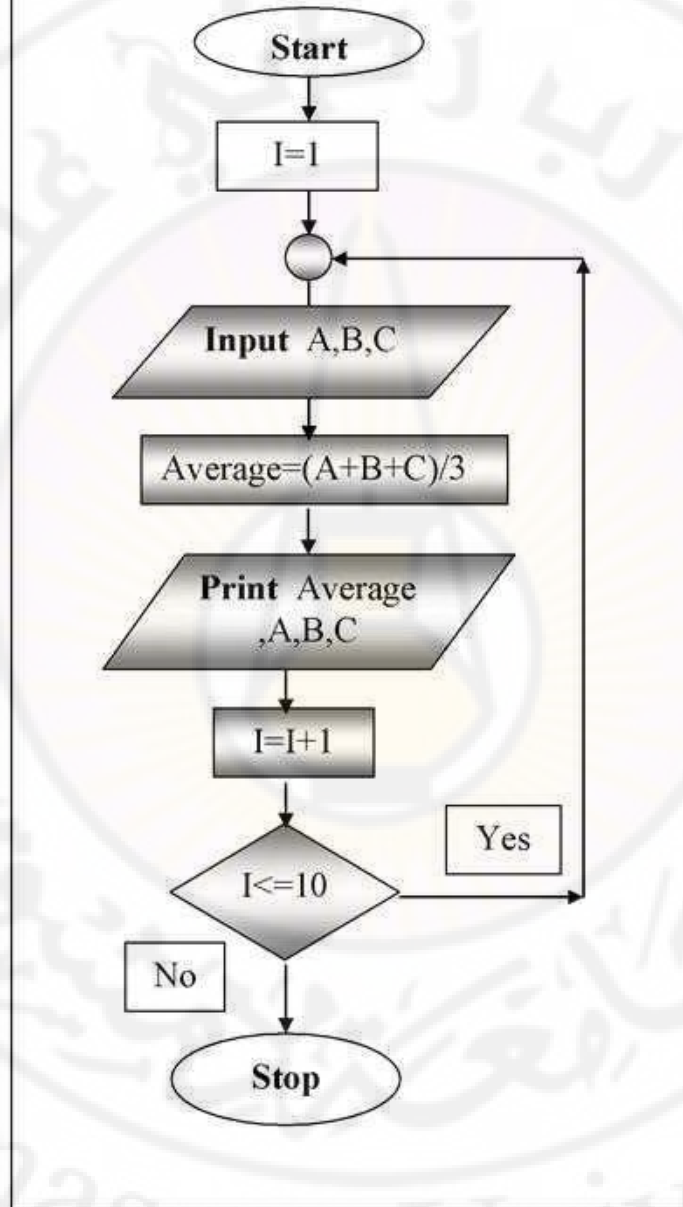


الشكل (٧-٥) - أساليب التكرار

مثال (5-12)

- ليكن لدينا عشر طلاب ونرغب في إدخال درجات هؤلاء الطلاب في ثلاثة مقررات
ثم حساب معدل كل طالب في المقررات الثلاثة، والمطلوب:
- ارسم المخطط التدفقي (الحلقي) الذي يقوم بما يلي:
- ❖ إدخال درجات الطلاب العشر في المقررات الثلاثة.

الخوارزمية وفق طريقة المخطط التدفقي الحلقي



الشكل (٨-٥) - يبين المخطط التدفقي الحلقي للمثال (٥-١٢)

- ❖ حساب معدل كل طالب.
- ❖ طباعة المعدل مع الدرجات لكل طالب.
- ثم اكتب الخوارزمية بطريقة Pseudo code .
- يبين الشكل (٥-٨) المخطط التدفقي الحلقي .

الخوارزمية وفق طريقة Pseudo code

1. **Start**
2. $I=1$
3. **Input** A,B,C
4. $Average = (A+B+C)/3$
5. **Print** Average , A , B , C
6. $I=I+1$
7. **if** $I \leq 10$ then go to 3
8. **Stop**

٥-٩- العداد Counter

في كثير من الأحيان نحتاج في برامجنا الحاسوبية إلى مفهوم العد Counting، فقد نريد مثلاً أن نعد عدد مرات تحقق شرطاً ما عند حل مسألة معينة، وقد تكون هذه العملية سهلة للإنسان لأنها أصبحت ضمن قدراته العقلية التي يكتسبها من الطفولة، إلا أن الحاسوب يحتاج إلى تصميم خوارزمية للعد Counting Algorithm تتضمن خطوات معينة إذا اتبعت يستطيع عندئذ القيام بمهمة العد.

ويمكن تحديد الخطوات التي يتبناها الحاسوب حتى يتمكن من العد كما يلي:

١. نجعل العداد مساوياً للصفر .

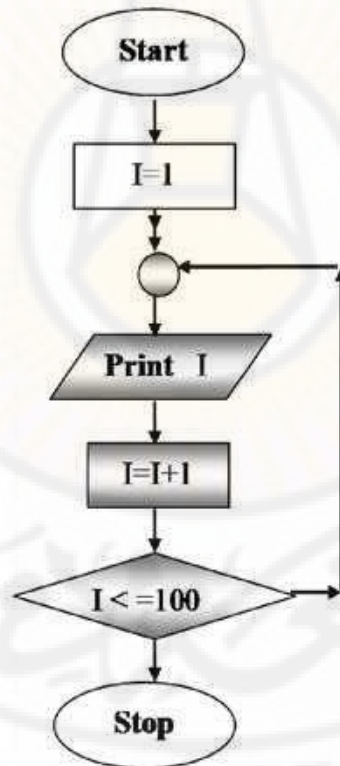
٢. نجعل القيمة الجديدة للعداد مساوية للقيمة القديمة له زائد واحد، أي أن:

قيمة العداد (الجديدة) = قيمة العداد (القديمة) + ١

٣. كرر الخطوات ابتداء من الخطوة ٢.

مثال (5-13)

ارسم المخطط التدفقي اللازم لطباعة الأعداد المحصورة بين العددين 1 و 100.
يبين الشكل (٥-٩) الخوارزمية وفق طريقة المخطط التدفقي الحلقي



الشكل (٥-٩) - يبين المخطط التدفقي الحلقي للمثال (5-13)

الخوارزمية وفق طريقة Pseudo code

- 1.Start
- 2.I=1
3. Print I
- 4.I=I+1
- 5.if $I \leq 100$ then go to 3
- 6.Stop

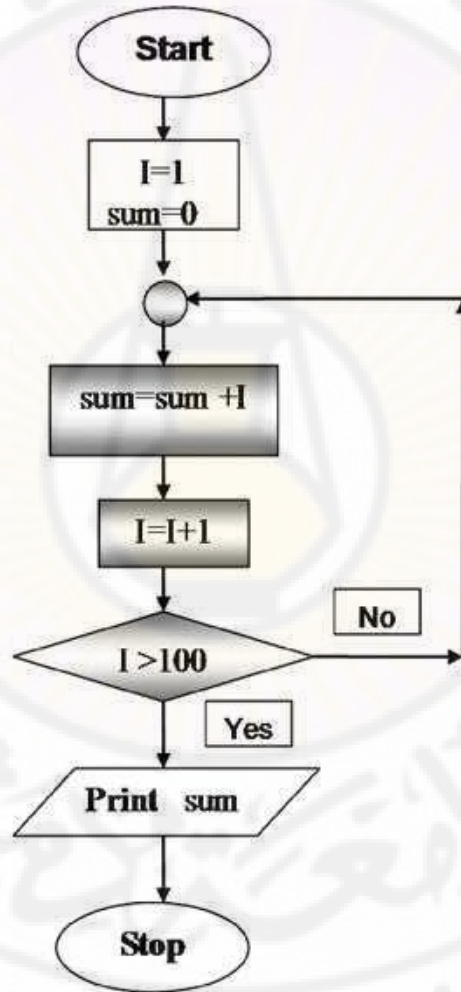
٥-١٠- المجاميع الإجمالية

في حالات كثيرة تحتاج بعض البرامج الحاسوبية إلى جمع مجموعة كبيرة من الأعداد التي تمثل معطيات ظاهرة معينة، فمثلاً قد نرغب في إيجاد المتوسط الحسابي لأعمار طلاب الجامعة، ولتحقيق هذا أولاً يجب أن نحسب مجموع أعمار الطلاب، وطبعاً ليس عملياً إعطاء رمز أبجدي لكل عمر طالب فقد تحتاج لأكثر من عشرة آلاف رمز، في مثل هذه الحالات نصمم خوارزمية معينة للتجميع تسمى خوارزمية التجميع Summers Algorithm تتضمن خطوات محددة إذا اتبعتها الحاسوب استطاع أن يجمع أي كمية من المعطيات باستخدام متغيرين اثنين إحداهما هو المتغير الذي نجمعه والآخر هو الجمع الإجمالي (المجمع)، ويمكن تحديد الخطوات التي يجب أن يتبناها الحاسوب لتحقيق ذلك في أربع خطوات هي:

١. اجعل المجمع مساوياً للصفر.
٢. ادخل قيمة واحدة للمتغير.
٣. اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير، أي أن:
قيمة المجمع الجديدة = قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.
٤. كرر ابتداءً من الخطوة الثانية.

مثال (5-14)

ارسم المخطط التدفقي اللازم لحساب وطباعة مجموع الأعداد المحصورة بين العددين 1 و 100، ثم اكتب الخوارزمية بطريقة Pseudo code . وبيّن الشكل (5-10) الخوارزمية وفق طريقة المخطط التدفقي الحلقي



الشكل (5-10) يبين طريقة المخطط التدفقي الحلقي للمثال (5-14)

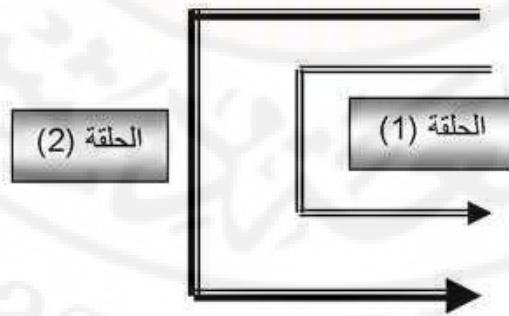
الخوارزمية وفق طريقة Pseudo code

- 1.Start
- 2.I=1
- 3.sum=0
- 4.sum=sum +I
- 5.I=I+1
- 6.if I > 100 then go to 8
7. go to 4
8. Print sum
- 9.Stop

١١-٥ - المخططات التدفقية الحلقية المتداخلة

Nested Loop Flowchart

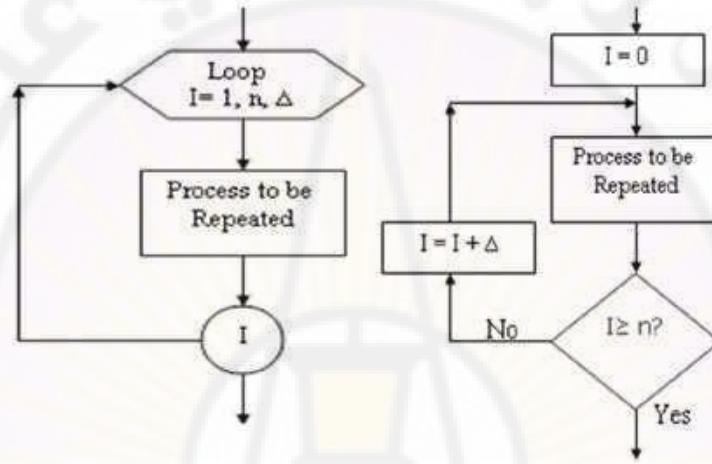
في هذه الحالة تكون الحلقات متداخلة فيما بينها بحيث لا تتقاطع فإذا كان لدينا مثلاً حلقتي (١١-٥) كما هو مبين في الشكل (١١-٥) فتسمى الحلقة (1) بالحلقة الداخلية (Inner Loop) بينما الحلقة (2) الحلقة الخارجية (Outer Loop) ويتم التناسق في عمليهما بحيث تكون أولوية التنفيذ للحلقة الداخلية.



الشكل (١١-٥) - يبين الحلقات المتداخلة

١٢-٥ - صيغة التكرار باستخدام الشكل الاصطلاحي

لقد عرفنا في الفقرتين السابقتين مفهوم التكرار البسيط والحلقات الضمنية ويمكننا الآن استخدام الشكل الاصطلاحي للتكرار كما هو موضح في الشكل (١٢-٥).



الشكل (١٢-٥) - يبين الشكل الاصطلاحي للتكرار

□ نلاحظ في الشكل (١٢-٥) أننا نحتاج إلى العناصر الآتية:

✓ القيمة الأولية للعداد I (هنا $I=1$).

✓ القيمة النهائية للعداد I (هنا n).

✓ قيمة الزيادة عند نهاية كل دورة (هنا Δ).

□ نلاحظ في الشكل (١٢-٥) أن إجراءات التكرار كانت تنتم طبقاً للخطوات

الآتية والمفصلة من قبل المبرمج:

١. أعط I قيمة أولية.

٢. أتم الإجراءات المطلوب إعادتها.

٣. (تقرير) إذا كانت قيمة العداد 1 وصلت إلى القيمة النهائية n اخرج إلى الخطوة التالية في البرنامج وإلا فإذهب إلى الخطوة (٤).

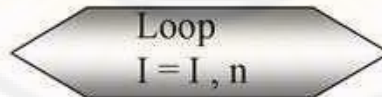
٤. زد I بمقدار الزيادة Δ .

٥. عد إلى الخطوة (٢).

يمكننا استبدال الخطوات المفصلة (1,2,3,4,5) في الشكل (٥-١٢) بخطوة مجملة واحدة مبينة في الشكل (٥-١٣) الذي يبين الشكل الاصطلاحي للتكرار حيث تنفذ هذه الخطوات بصورة أوتوماتيكية من قبل الحاسوب، وهذا من شأنه تسهيل عملية البرمجة واختصار عدد التعليمات في البرنامج وتجنب بعض الأخطاء.

□ ملاحظة (٥-٢)

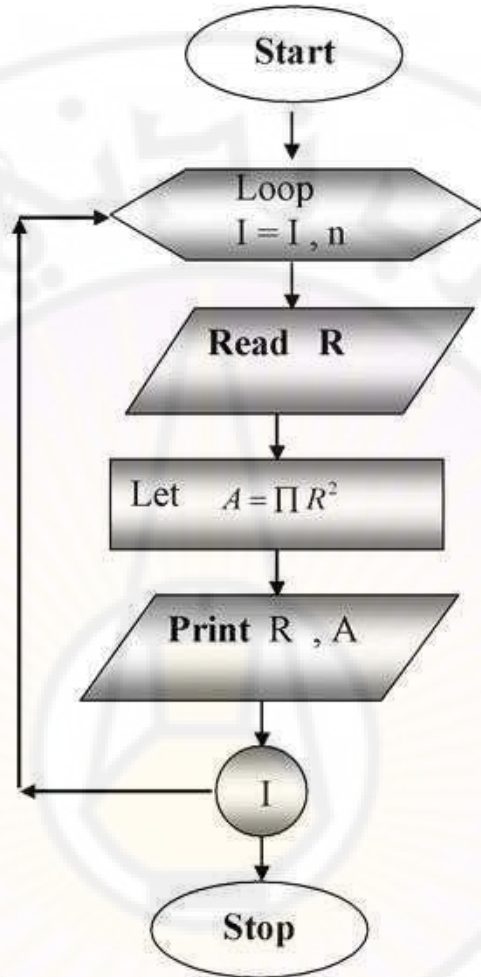
تُعد قيمة Δ تساوي 1 دائماً إذا لم تعط قيمة أخرى بخلاف ذلك، وفي حالة عدم ذكر قيمة Δ يصبح الشكل الاصطلاحي الوارد في الشكل (٥-١٣) كما يلي حيث تكون قيمة Δ تساوي 1 وبصورة أوتوماتيكية.



الشكل (٥-١٣) - يبين الشكل الاصطلاحي للتكرار

مثال (5-15)

أعد حل المثال الموضح في الشكل (٥-١٢) لإيجاد مساحة n من الدوائر باستخدام الشكل الاصطلاحي للدوران.
خطوات الحل مبينة في الشكل (٥-١٤) الذي يوضح المخطط التدفقي الدوراني.



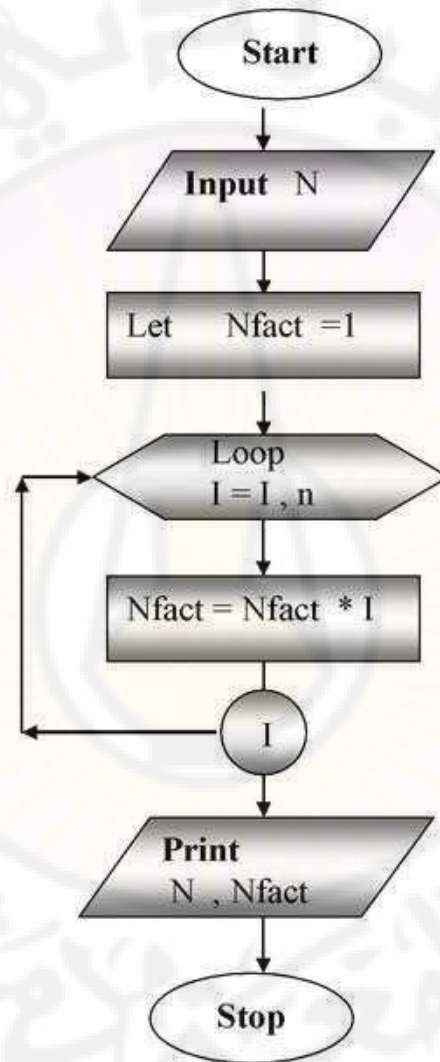
الشكل (١٤-٥) - يوضح المخطط التدفقي الحلقي للمثال (5-15)

مثال (5-16)

ارسم المخطط التدفقي لإيجاد قيمة العامل للعدد N ($N!$).

الحل:

. $N! = N(N-1)(N-2) \dots 3 \cdot 2 \cdot 1$ ، خطوات الحل مبينة في الشكل (١٥-٥) .



الشكل (١٥-٥) - يوضح المخطط التدفقي لحساب $N!$ للمثال (5-16)

٥-١٣ - تطبيقات على الخوارزميات

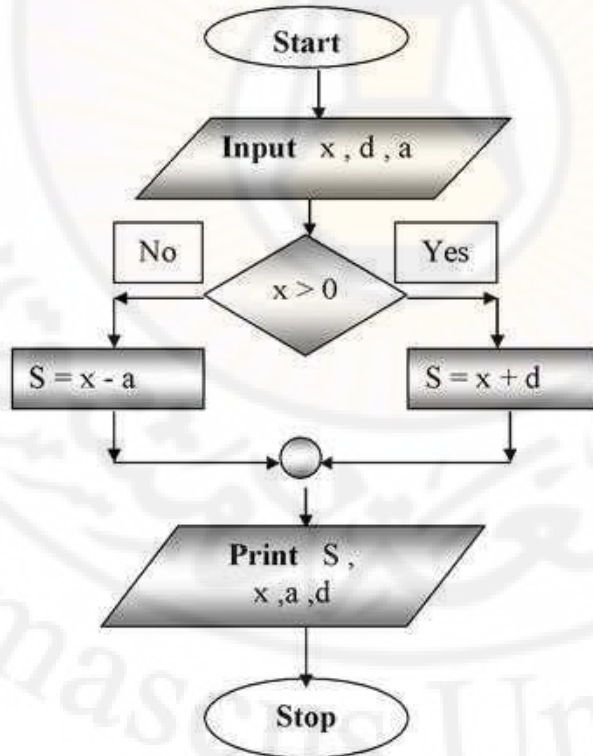
فيما يلي نستعرض مجموعة من التمارين التي بدورها تساعد الطالب على فهم أنواع المخططات التدفقية بشكل جيد، حيث إننا عند طرح التمرين لا نأتي على ذكر المخطط التدفقي اللازم استخدامه، والذي يحدد ذلك هو طبيعة المسألة.

مثال (٥- 17)

ارسم المخطط التدفقي اللازم لحساب وطباعة قيمة الدالة الرياضية التالية:

$$s = \begin{cases} x+d & \text{if } x > 0 \\ x-a & \text{if } x \leq 0 \end{cases}$$

الحل: يبين الشكل (٥- ١٦) المخطط التدفقي .

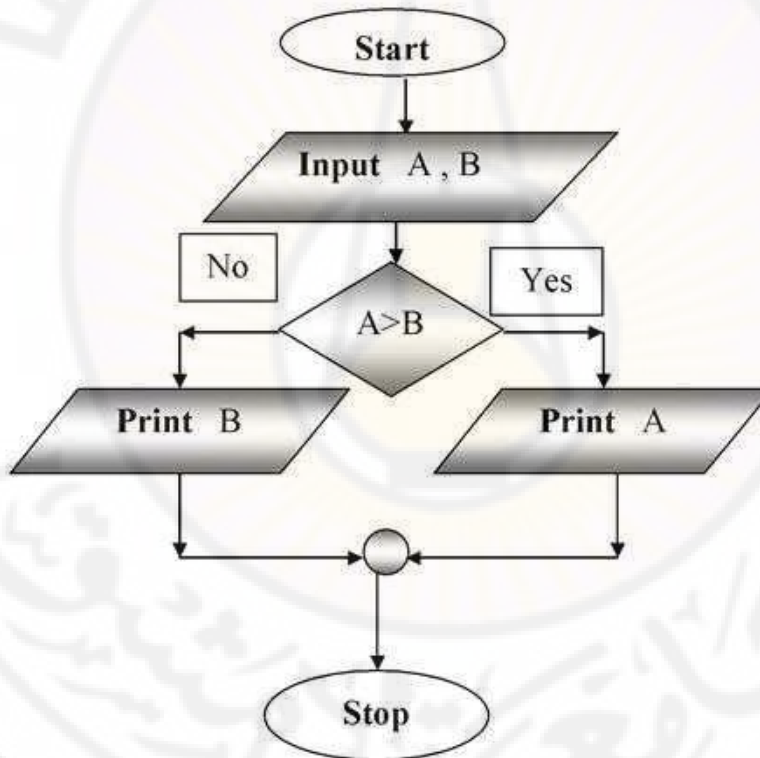


الشكل (٥- ١٦) - يبين المخطط التدفقي للمثال (٥- 17)

مثال (5-18)

ارسم المخطط التدفقي اللازم لإيجاد وطباعة العدد الأكبر والأصغر من بين عددين مُدخلين A و B .

الحل: يبين الشكل (5-17) المخطط التدفقي .



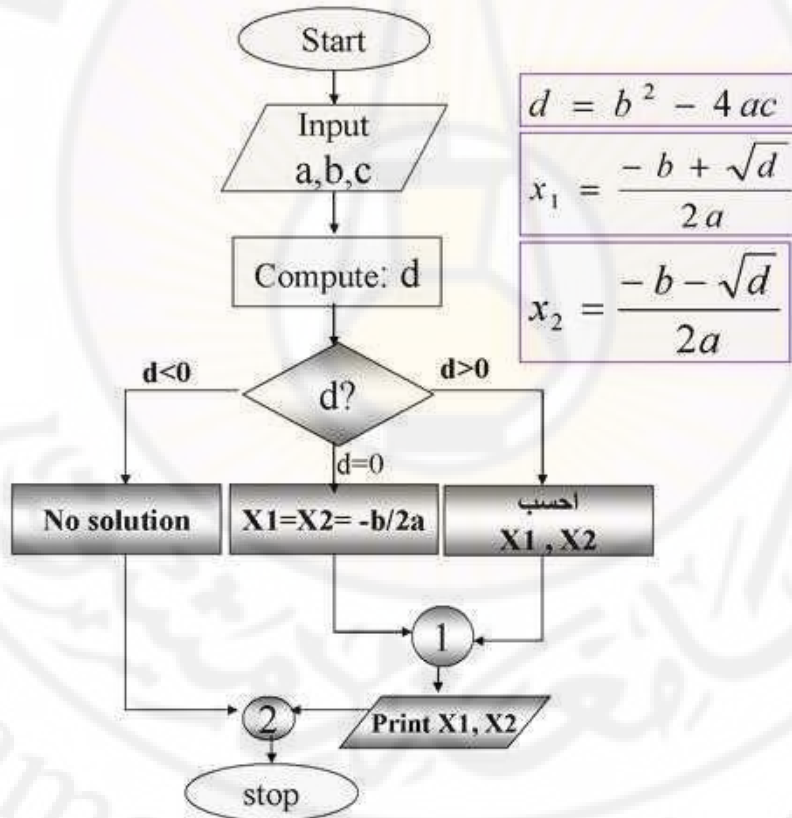
الشكل (5-17) - يبين المخطط التدفقي للمثال (5-18)

مثال (5-19)

ضع الخوارزمية والمخطط التدفقي اللازمان لحساب وطباعة جذور معادلة من الدرجة الثانية والتي تملك الشكل التالي:

$$ax^2 + bx + c = 0$$

الحل: يبين الشكل (5-18) المخطط التدفقي .

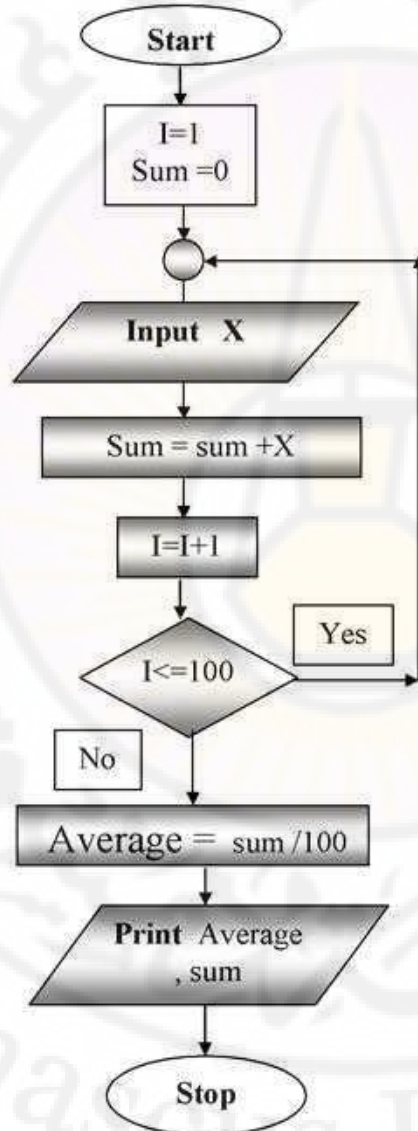


الشكل (5-18) - يبين المخطط التدفقي للمثال (5-19) .

مثال (5-20)

ارسم المخطط التدفقي اللازم لحساب وطباعة المتوسط الحسابي لـ 100 عدداً صحيحاً يتم إدخالها من قبل المستخدم (ليست الأعداد المدخلة متتالية).

الحل:

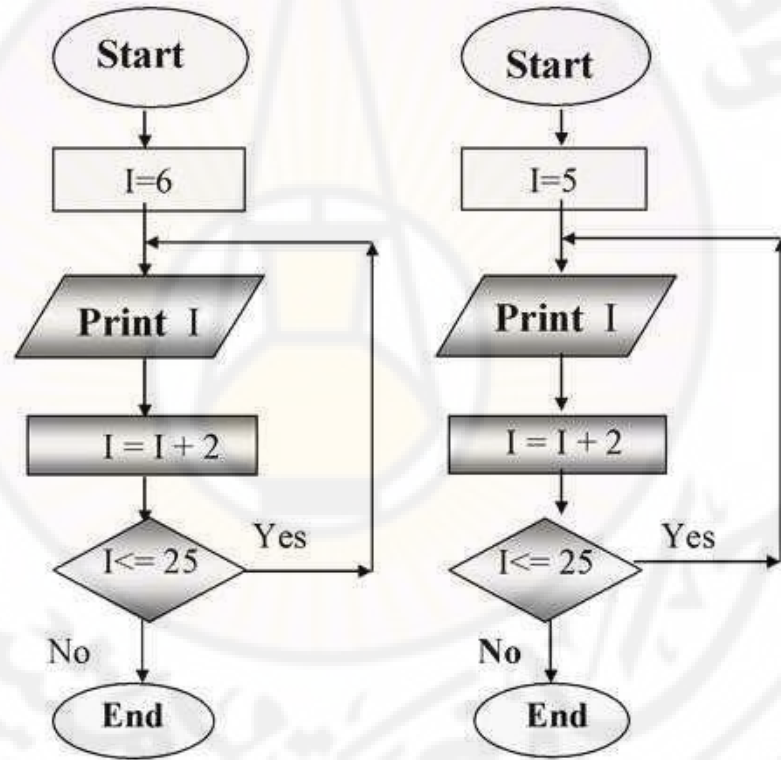


الشكل (5-19) - يبين المخطط التدفقي للمثال (5-20)

مثال (5- 21)

ارسم المخطط التدفقي اللازم لطباعة الأعداد الفردية من 5 حتى 25 .
أعد السؤال من أجل طباعة الأعداد الزوجية من 5 حتى 25 .

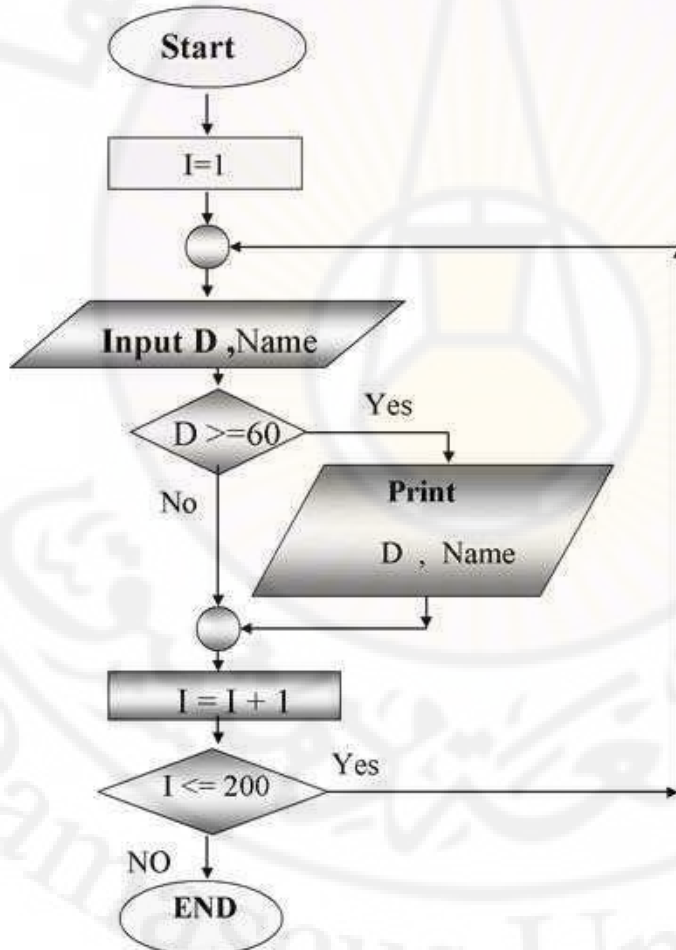
الحل: يبين الشكل (٥ - ٢٠) المخطط التدفقي .



الشكل (٥ - ٢٠) - يبين المخطط التدفقي للمثال (5- 21)

مثال (5-22)

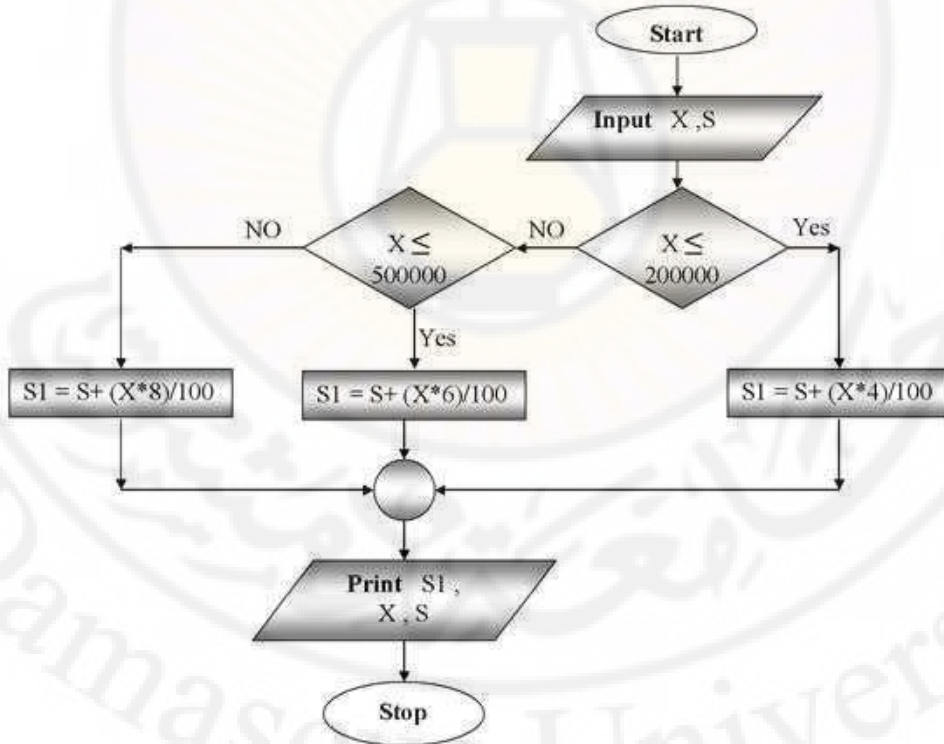
يبلغ عدد طلاب السنة الأولى في قسم هندسة الحاسوب والأتمتة 200 طالباً في
مقرر مدخل على الحاسوب والبرمجة ، والمطلوب :
اكتب خوارزمية (المخطط التدفقي) لإدخال أسماء ودرجات جميع الطلاب
ويطبع فقط أسماء ودرجات الطلاب الناجحين .
الحل: يبين الشكل (5-21) المخطط التدفقي .



الشكل (5-21) - يبين المخطط التدفقي للمثال (5-22)

مثال (5-23)

رجل مبيعات يتقاضى راتباً شهرياً مقداره $S = 10000$ ليرة سورية ويتقاضى عمولة مقدارها 4% إذا كانت المبيعات الشهرية لا تتجاوز 200000 ليرة سورية ويتقاضى عمولة مقدارها 6% إذا كانت المبيعات الشهرية لا تتجاوز 500000 وإذا زادت المبيعات عن 500000 فإن نسبة العمولة تصبح 8% والمطلوب :
اكتب خوارزمية (المخطط التدفقي) لحساب وطباعة الدخل الشهري الكامل للموظف، علماً أن الراتب الشهري الكلي يساوي :
الراتب الشهري الكامل = الراتب الأساسي + العمولة ($S1 = S + W$)
الحل: يبين الشكل (5-22) المخطط التدفقي .



الشكل (5-22) - يبين المخطط التدفقي للمثال (5-23)

مسائل عامة

- ١- ارسم المخطط التدفقي اللازم لحساب وطباعة المتوسط الحسابي لـ 500 عدداً حقيقياً يتم إدخالها من قبل المستخدم (ليس من الضروري أن تكون الأعداد المدخلة متتالية).
- ٢- ارسم المخطط التدفقي اللازم لحساب مجموع الأعداد الفردية الصحيحة المحصورة بين العددين n و m (حيث $n < m$ وأن n فردي).
- ٣- شركة تجارية يبلغ عدد موظفيها 300 موظفاً ويتقاضون رواتب شهرية مختلفة والمطلوب:
- ارسم المخطط التدفقي اللازم لقراءة أسماء ورواتب كل الموظفين ومن ثم طباعة أسماء ورواتب الموظفين الذين تزيد رواتبهم الشهرية عن 10000 ليرة سورية.
- ٤- ارسم المخطط التدفقي اللازم لحساب وطباعة قيمة المقدار Y والمعرف كما يلي:

$$Y = \sum_{i=1}^n \frac{i}{i+1}$$
$$\dots = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$$

- ٥- ارسم المخطط التدفقي اللازم لإدخال عدد حقيقي R من لوحة المفاتيح ويطبع كلمة Positive إذا كان العدد موجباً ويطبع كلمة Negative إذا كان عدد سالباً .

٦- يرغب قسم هندسة الحواسيب والأتمتة بمنح مكافأة مالية قدرها 10000 ليرة سورية لكل طالب في السنة الأولى يحصل على درجة 90 أو أكثر في مقرر مدخل إلى الحاسوب والبرمجة والمطلوب :
ارسم المخطط التدفقي اللازم لإدخال أسماء ودرجات الطلاب ويطبع فقط أسماء ودرجات الطلاب الذين يستحقون المكافئة .
٧- ارسم المخطط التدفقي اللازم لحساب وطباعة مجموع مربعات الأعداد من 1 إلى 100 .

٨- في شركة 50 مندوب مبيعات يتقاضى كل مندوب راتباً شهرياً مقداره $S = 10000$ ليرة سورية ويتقاضى عمولة مقدارها 4% إذا كانت المبيعات الشهرية لا تتجاوز 200000 ليرة سورية ويتقاضى عمولة مقدارها 6% إذا كانت المبيعات الشهرية لا تتجاوز 500000 ، وإذا زادت المبيعات عن 500000 فان نسبة العمولة تصبح 8 % والمطلوب :
اكتب خوارزمية (المخطط التدفقي) لحساب وطباعة الدخل الشهري الكامل للموظف، علماً أن الراتب الشهري الكلي يساوي :
الراتب الشهري الكامل = الراتب الأساسي + العمولة ($S1 = S + W$)



الفصل السادس

أساسيات لغة C++

Basics of C++ language



أساسيات لغة C++

٦-١- مقدمة

يقوم المبرمجون بكتابة برامجهم الحاسوبية بلغات برمجية مختلفة، بعضها يكون مفهوماً مباشرة من قبل الحاسوب وبعضها الآخر يحتاج إلى خطوات وبرامج ترجمة ، ويتوفر حالياً عشرات من لغات البرمجة المستخدمة والتي يمكن تقسيمها إلى نوعين رئيسيين:

١- لغات البرمجة منخفضة المستوى (LLL) ، مثل :

- لغة الآلة Machine Language .

- لغة التجميع Assembly Language .

٢- لغات البرمجة عالية المستوى (HLL) .

يستطيع الحاسوب أن يفهم وبشكل مباشر لغة الآلة الخاصة به والتي هي اللغة الطبيعية بالنسبة له، وتتألف لغة الآلة بشكل عام من سلاسل من الأصفار (0) والواحدات (1) والتي تعطي الأوامر للحاسوب من أجل تنفيذ تعليماته الأولية، كل تعليمة على حدة ، ولا بد من الإشارة إلى أن لغة الآلة ترتبط ارتباطاً وثيقاً بالآلة (الحاسوب) ، أي بمعنى آخر أن لكل نوع من أنواع الحواسيب لغة آلة خاصة به، وتعتبر لغة الآلة من اللغات الصعبة والمرهقة للمبرمجين والتي تستغرق وقتاً طويلاً من أجل كتابة أي برنامج، ولا يستطيع كتابة البرامج بلغة الآلة إلا الشخص الذي لديه إلمام كامل ببنية الحاسوب.

بدأ المبرمجون باستخدام مصطلحات قريبة من اللغة الإنكليزية للتعبير عن العمليات (التعليمات) الأولية للحاسوب ، وهذه المصطلحات أصبحت تمثل الأساس الذي انطلقت من لغة التجميع ، وكان لا بد من توفر برنامجاً يقوم بترجمة البرامج

المكتوبة بلغة التجميع إلى لغة الآلة ليتمكن الحاسوب من فهمها وتنفيذها ، فتم تطوير ما يسمى بالمجمعات Assemblers .
ومع تزايد استخدام الحواسيب ومن أجل تسريع عملية البرمجة وتبسيطها تم تطوير لغات برمجية جديدة سُميت لغات البرمجية عالية المستوى HLL (High Level Languages) ، وسُميت البرامج التي تقوم بتحويل البرامج المكتوبة باللغات البرمجية العالية المستوى إلى لغة الآلة باسم المترجمات compilers وهناك بعض اللغات التي تستخدم نوعاً آخر من البرمجيات تسمى المفسرات interpreters وهي تقوم بتنفيذ البرامج المكتوبة بلغات برمجية عالية المستوى تعليمية تعليمية دون الحاجة لترجمة كامل البرنامج كما تفعل المترجمات.
تساعد لغات البرمجة عالية المستوى المبرمجين على كتابة تعليمات قريبة من لغة الإنسان ، وقد أصبحت اللغات البرمجية العالية المستوى أكثر تقبلاً من قبل المبرمجين ، وتعد اللغتان C و C++ من أكثر اللغات البرمجية العالية المستوى قوة وانتشاراً.

٦-٢- تاريخ تطور لغة C++

في عام 1960 صُممت لغة CPL (Combined Programming Language) ، وتم ذلك في جامعة لندن وقد استخدمت فيها بعض تعليمات لغة ALGOL-60، وفي عام 1967 انبثقت لغة BCPL من لغة CPL ، وفي عام 1972 قامت مخابر شركة AT&T الأمريكية باستنباط لغة جديدة من لغة B أخذ منها أحسن تعليماتها وأضاف إليها أوامراً جديدة وأنواعاً جديدة للمعطيات ، وكثيراً من التوابع التي تفيد المبرمج وسُميت هذه اللغة بلغة C .
ومنذ ذلك التاريخ أخذت لغة C شهرة واسعة لأنها أصبحت تنتمي إلى اللغات البرمجية العالية المستوى High Level Languages من حيث سهولة

الاستخدام من ناحية ومن ناحية ثانية تنتمي إلى اللغات البرمجية المتدنية المستوى من حيث قدرة اللغة على مخاطبة مكونات الحاسوب المادية Hardware. ومع نجاح وانتشار لغة C انتشرت لهجات كثيرة لمستخدمي لغة C ، لولا أن قام معهد القياسات الأمريكية بعملية توحيد لهذه اللهجات المختلفة فأصدر في عام 1983 اللغة القياسية، وإن معظم مترجمات لغة C تتوافق مع ANSI C مثل : Borland C, Microsoft C, Lattice C . لقد أدى الاستخدام الواسع للغة C على العديد من الأنواع المختلفة للحواسيب إلى بعض الاختلافات بين نسخ لغة C التي يمكن أن تكون متشابهة ولكنها غالباً غير متوافقة فيما بينها، لذلك ظهرت الحاجة إلى تعريف نسخة معيارية من لغة C . وقد تم تشكيل لجنة عام 1983 للإشراف على تحديد المعايير والمواصفات للغة C المعيارية وقد تم عام 1989 المصادقة على النسخة المعيارية التي تم إصدارها، بعدها قامت منظمة ANSI بالتعاون مع المنظمة العالمية للمعايرة ISO بتعميم معايير لغة C على مختلف أنحاء العالم وقد سميت النسخة المعيارية من اللغة باسم ANSI C .

بعد ذلك تم تطوير لغة البرمجة C++ التي تُعد امتداداً للغة C وقد قدمت لغة C++ العديد من الإمكانيات التي تسمو بلغة C وتعلو بها، ولكن الأهم من ذلك كله قدرات " البرمجة غرضية التوجه (Object Oriented Programming) .

٦-٣- مزاي لغة C/C++

- تتميز لغة C بمجموعة من المزايا مثل:
 - لغة عامة: هذا يعني أنها تصلح لتنفيذ برامج قواعد المعطيات والرسومات والحسابات ونظم التشغيل وغيرها.

- **C لغة تركيبية Structured Language** : هذا يعني أن البرنامج المكتوب بلغة C هو عبارة عن تابع رئيسي يمكن أن يستدعي مجموعة من التوابع الأخرى وكل تابع هو عبارة عن مجموعة من التعليمات.
- **C لغة تتعامل على مستوى " البت " Bit Manipulation**: حيث نستطيع أن نقرأ ونكتب ونقوم بعمليات على مستوى الـ Bit وهذه الميزة جعلت من لغة C لغة متخصصة في بعض مجالات التحكم الآلي والروبوت وبرامج معالجة الصور وضغط الملفات واكتشاف الأعطال.
- **C لغة متنقلة Portable**: هذا يعني أنه يمكن للبرنامج المكتوب بلغة C أن يعمل مع أكثر من جهاز مثل IBM أو Apel أو الأجهزة المتوسطة والكبيرة مع بعض التعديلات الطفيفة.
- **C لغة سريعة** : لأن أدوات اللغة تتعامل بشكل مباشر مع الآلة (الحاسوب) مما يختصر وقت التنفيذ.
- **C لغة قياسية**: هذا يعني أن معظم مترجمات اللغة تتوافق مع اللغة القياسية ANSI C .
- **C لغة نظام التشغيل UNIX**: هذا يعني أن نظام التشغيل الشهير UNIX مكتوب بهذا اللغة ، وهذا يدل على ثراء هذه اللغة ومرونتها.
- تحتوي لغة C/C++ على مجموعة ضخمة من التوابع (الدوال) جعلت منها لغة كبيرة نسبياً ولحل هذه المشكلة تم توزيع هذه التوابع على مجموعة من المكتبات المتخصصة ، كل مكتبة لها اسم معين وتحتوي على توابع معينة.
- أصبح من الممكن في لغة C/C++ الإعلان الحر عن المتغيرات في أي موقع من البرنامج مما يتيح ربط المتغير بالوظيفة التي من أجلها تم الإعلان عنه مما يزيد من سهولة ومتابعة وفهم البرنامج.

- في لغة C يتم الإعلان عن الثوابت باستخدام الكلمة المحجوزة `#define` وفي هذه الطريقة لا يتم الإعلان عن نوع الثابت المعلن عنه، أما في لغة C++ يتم استخدام الكلمة المحجوزة `const` للإعلان عن الثوابت وفيها يتم تحديد نوع الثابت ومن مزايا هذه الطريقة أنها تساعد المترجم على فحص الأنواع (أنواع المعطيات) وحجز ذاكرة تتناسب مع نوع الثابت.

٦-٤- بيئة لغة C++

تمر ببرامج لغة C++ بعدة مراحل رئيسية قبل أن يتم تنفيذها ، وهي :

المرحلة الأولى: مرحلة كتابة البرنامج

تهدف هذه المرحلة إلى كتابة نص البرنامج بواسطة محرر النصوص والقيام بمراجعة البرنامج وإجراء التصحيحات في حال الضرورة. بعدها يتم تخزين البرنامج على وحدة التخزين مثل القرص الصلب أو القرص المرن ونختار عندها الاسم المناسب للبرنامج ، وتنتهي أسماء البرامج بالامتداد C أو CPP .

المرحلة الثانية : مرحلة الترجمة

يقوم المبرمج بعدها بطلب أمر الترجمة للبرنامج، فيقوم المترجم بترجمة البرنامج إلى لغة الآلة، وقبل البدء بمرحلة الترجمة يتم تنفيذ مرحلة ما قبل الترجمة التي تهدف إلى التعامل مع أوامر خاصة تسمى بموجهات ما قبل الترجمة `Preprocessor directives` ، وتحدد هذه الموجهات جملة من المعالجات والإجراءات الواجب تنفيذها على نص البرنامج قبل الترجمة الحقيقية.

المرحلة الثالثة: مرحلة الربط

تسمى المرحلة اللاحقة لمرحلة الترجمة بمرحلة الربط (الوصل) `linking` ، فبرامج لغة C++ تتضمن استدعاء لتوابع المعرفة في مكان آخر مثل المكتبات المعيارية أو المكتبات الخاصة .

المرحلة الرابعة: مرحلة التحميل

في مرحلة التحميل loading يتم وضع البرنامج في الذاكرة الرئيسية قبل البدء بتنفيذ البرنامج وتتم هذه المرحلة باستخدام المحمل loader .

المرحلة الخامسة: مرحلة التنفيذ

يبدأ الحاسوب في هذه المرحلة بتنفيذ البرنامج تعليمة فتعليمة تحت إشراف وحدة المعالجة المركزية CPU .

٦-٥- الشكل العام لبنية البرنامج بلغة C++

يتكون أي برنامج مكتوب بلغة C++ من أربعة عناصر أساسية هي التالية:

- ١- استدعاء المكتبات المستخدمة في البرنامج.
- ٢- تعريف المعطيات المستخدمة في البرنامج.
- ٣- تعليمات (عبارات) البرنامج.
- ٤- توابع (دوال) المستخدم إن وجدت فهي تكتب في الجزء الأخير من البرنامج.

الشكل العام لبنية برنامج بلغة C++:

```
#include<iostream.h>
void main ( )
{
```

```
    body {          Body of main( )
```

```
    } // end main
```

تحتوي لغة C++ على مجموعة من المكتبات، كل مكتبة تحتوي على مجموعة من التوابع وكل تابع له وظيفة معينة يؤديها داخل البرنامج. الخطوة الأولى في البرنامج تتمثل في استدعاء المكتبات التي يتطلبها البرنامج وتسمى هذه العملية بالترويسة Header لذلك نجد المكتبات تنتهي أسماؤها باللاحقة (h)، ومن أشهر المكتبات في لغة C++ المكتبة المتضمنة دوال الإدخال والإخراج وتسمى iostream .

٦-٦- قواعد بناء البرنامج بلغة C++

إن أي برنامج سيكتب بلغة C++ يجب أن يخضع إلى القواعد التالية:

- يبدأ البرنامج بالعبارة التالية:

```
# include <iostream.h>
```

- يتكون البرنامج من دالة رئيسية (main) وتبدأ بالقوس " { " (قوس الفتح) وتنتهي بالقوس " } " (قوس الإغلاق).
- جميع كلمات ودوال اللغة تكتب بالأحرف الصغيرة.
- كل عبارة برمجية يجب أن تنتهي بفاصلة منقوطة (;) مع وجود بعض الاستثناءات والتي سوف نتعرف عليها فيما بعد.
- يجوز كتابة أي ملاحظة أو تعليق خاصين بالبرنامج، وذلك بوضعهما بين علامتين /* */ أو بعد علامتين //.

لنكتب أول برنامج بلغة C++ والذي يقوم بطباعة العبارة :

Welcome to C++ Programming !

1. `//This program will display a message on the screen.`
2. `#include<iostream.h>`
3. `Int main ( )`
4. `{`
5. `cout <<"welcome to C++ Programming ! \n " ;`
6. `return 0`
7. `}//End main`

شرح البرنامج:

السطر 1 : `comment`

الذي يبدأ بالإشارة (الشرطة المائلة) // على أن النص المكتوب خلف هذه الإشارة هو عبارة عن تعليق ، حيث يحق للمبرمجين إدراج تعليقاتهم ضمن نص البرنامج وتحسين القدرة على قراءته فيما بعد وخصوصاً من قبل أشخاص آخرين ، وتساعد التعليقات على فهم البرنامج، وهي لا تتطلب من الحاسوب القيام بأي فعل (تعليلة غير تنفيذية) عند تنفيذ البرنامج ، فالمترجم يقوم بتجاهل جميع أسطر التعليقات أثناء عملية الترجمة. تستخدم الشرطة المزدوجة (//) إذا كان التعليق يمتد لسطر واحد فقط `single-line comment` ، وهناك نوع آخر من التعليقات يتيح لنا كتابة تعليقات تمتد إلى عدة أسطر `multi-line comments` ، ونستطيع كتابة التعليق السابق كما يلي:

/*

This program will display a message on the screen

*/

يبدأ الرمز " /* " ثم التعليق وينتهي بالرمز " */ " ، ونجد أن نهاية السطر لا تعني انتهاء التعليق لذا يمكننا كتابة ما نشاء من أسطر التعليقات قبل الانتهاء بالرمز " /* " .

السطر 2 : #include<iostream.h>

يسمى هذا بموجه المهيئ Preprocessor directive ، وهو عبارة عن تعليمة للمترجم أن يدرج كل النص الموجود في الملف `iostream.h` في البرنامج، وهو ملف يجب تضمينه مع أي برنامج يحتوى على عبارات تطبع معطيات على الشاشة أو تستقبل معطيات من لوحة المفاتيح ، ويسمى `iostream` الملف الرأسي (header file)، وهناك الكثير من الملفات الرأسية الأخرى، فمثلاً إذا كنا نستعمل في برنامجنا توابع رياضية مثل: `sin()` و `cos()` نحتاج إلى شمل ملف رأسي يدعى `math.h`، وإذا كنا نتعامل مع سلاسل المحارف سنحتاج للملف `string.h` ، وعموماً هنالك عدد كبير من الملفات الرأسية التي يجب تضمينها وذلك حسب طبيعة البرنامج ، وتعد الملفات الرأسية جزء مهم من برامج لغة C++ وسنحتاج إلى شمل الملف `iostream.h` لتشغيل أي برنامج يقوم بعمليات الإدخال والإخراج.

السطر 3 : التابع (main)

هو بداية البرنامج الرئيسي (التابع `main`) ، حيث يمكن أن يحتوي البرنامج بلغة C++ على تابع واحد أو أكثر (`functions`) ولكن من الضروري أن يكون أحدهما التابع `main` ، ويجب أن يتبع التابع القوسين () لتشير دائماً إلى التابع ، بينما تشير الكلمة `int` إلى أن التابع `main` هو تابع صحيح (`integer`) ، أي أنه يعيد قيمة صحيحة كخرج للتابع.

السطر 4 : قوس الفتح " { "

يدل على بداية جسم التابع `main` .

السطر 5 : تعليمة الخرج على الشاشة

`cout <<"welcome to C++ Programming !"`

هذه التعليمة (statement) تجبر الحاسوب أن يظهر على الشاشة النص المحصور بين علامتي الاقتباس " " ، ويسمى هذا النص بسلسلة محرفية ، ويجب أن تنتهي كل تعليمة في برنامج C++ بفاصلة منقوطة " ; " (semi colon). الكلمة المحجوزة أو العبارة `cout` الذي تلفظ كـ `C out` يمثل غرض في C++ مقترن مع الشاشة والعامل `<<` الذي يسمى بعامل الوضع `Put to operator` يجبر على إرسال الأشياء التي على يمينه إلى أي شيء يظهر على يساره.

العبارة `\n` : تشير إلى الانتقال إلى بداية سطر جديد بعد عملية الطباعة ، والكلمة المحجوزة `endl` تشير إلى الانتقال إلى بداية سطر جديد .

السطر 6 : العبارة `return`

يبين خرج التابع الرئيسي `main()` فيعيد القيمة 0 أي لا يعيد شيئاً، أي لا ليس له خرج ، لذلك تُستخدم القيمة 0 مع العبارة `return` ، وتكتب العبارة `return 0;` في نهاية التابع `main()` إذا لم يسبقه العبارة `void` ، والقيمة 0 تشير إلى أن البرنامج انتهى نهاية صحيحة وسيبدو لنا سبب تضمين هذه العبارة واضحاً عندما نتعرف على التوابع في C++ بالتفصيل.

يمكن استخدام الكلمة المحجوزة `void` التي تدل على أن التابع لا يعيد شيئاً و عندها يجب حذف السطر 6 ، لأن أي تابع من نوع `void` لا يحتوي على العبارة `return 0` .

السطر 7 : قوس الإغلاق " }

يدل على نهاية جسم التابع `main` .

وعند تنفيذ البرنامج السابق سنحصل على الخرج التالي:

RESULT

Welcome with C++ language

Press any key to continue

مثال (1-6)

أكتب برنامجاً بلغة C++ يقوم بطباعة ما يلي :

C++ Language is a High Level Language

C++ Language

is

a Structured Language.

البرنامج:

//The second program in c++

include <iostream.h>

void main()

{

cout<<" C++ Language is a High Level Language"<<endl;

cout<<" C++ Language "<<endl;

cout<<" is "<<endl;

cout<<" a Structured Language "<<endl;

}

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

C++ Language is a High Level Language
C++ Language
is
a Structured Language
Press any key to continue

مثال (2-6)

البرنامج التالي يوضح كيفية استخدام أمر الطباعة من أجل طباعة بعض القيم دون التصريح عنها .

```
#include <iostream.h>
void main ( )
{
    cout << 7 << " is an integer.\n";
    cout << 'a' << "is a character.\n";
}
```

RESULT

7 is an integer.
a is a character
Press any key to continue

من خرج البرنامج يتضح لنا ما يلي:

- يتم حصر النص المطلوب ظهوره على الشاشة بين علامتي اقتباس مزدوجة " is an integer " .
- تتم كتابة الثوابت الرقمية بدون علامتي اقتباس `cout<<7` .
- يتم حصر حرف واحد مطلوب ظهوره على الشاشة بعلامة اقتباس مفردة `cout << 'a'` .

لا تنشئ العبارة `cout` أسطراً جديدة تلقائياً، والمخرجات في المثال (6 - 3) توضح ذلك.

مثال (3-6)

البرنامج التالي يوضح كيفية استخدام أمر الإخراج `cout` في طباعة بعض القيم دون ترك فراغات بين المخرجات .

```
#include<iostream.h>
void main ( )
{
    cout<<10;
    cout<<20<<30;
}
```

RESULT

102030

Press any key to continue

حيث يلتصق الخرج ببعضه بعضاً ، لذا من الأفضل أن يكون لدينا طرق في C++ للتحكم بطريقة تنسيق الخرج والتي منها متواليات الهروب (Escape Sequences) .

٦-٦-١ - متواليات الهروب (Escape Sequences)

نلاحظ في المثال (6 - 2) أنه لم تتم طباعة \n على الشاشة ، وأن " \ " تسمى الشرطة الخلفية (Back slash) أو حرف هروب (Escape character) وتسمى الشرطة \ والحرف الذي يليها متواليات الهروب. متواليات الهروب \n يعني الانتقال إلى سطر جديد حيث يجبر المؤشر على الانتقال إلى بداية السطر التالي ، ويبين الجدول (٦-١) بعض متواليات الهروب الشائعة .

متوالية الهروب	الوصف
\n	الانتقال إلى بداية سطر جديد
\t	وضع مسافة أفقية بين قيم الخرج بمقدار 8bits
\b	حرف التراجع back space أرجع المؤشر خطوة إلى الخلف
\\	طباعة شرطة خلفية
\r	يجبر المؤشر على الانتقال إلى بداية هذا السطر
\v	طباعة مسافة رأسية " "
\?	طباعة علامة استفهام ؟
\a	إصدار صوت alert(deep)
\"	طباعة علامة اقتباس مزدوجة

الجدول (٦-١) - يبين بعض متواليات الهروب الشائعة

٦-٦-٢- التعليلة endl

```
cout<<"sum="<<sum<<endl
```

تطبع النص المحرفي sum= متبوعاً بقيمة sum ، ونلاحظ أننا استخدمنا endl وهو وسيلة أخرى في C++ للانتقال إلى سطر جديد، ويسمى مُناوِر manipulator و endl اختصاراً لـ end line، وهو يعمل تماماً كما تعمل متوالية الهروب \n .

٦-٦-٣- رموز لغة C++

يبني البرنامج في لغة C++ من مجموعة من العناصر التالية :

- الحروف الانكليزية الصغيرة a, b, c, d, ... x, y, z
- الحروف الانكليزية الكبيرة A, B, C, D, ... X, Y, Z
- الأرقام العربية 0, 1, 2, 3, ... 8, 9
- رموز خاصة مثل :

<	<=	>	>=	+	-	*	/	=	#	%	\$
?	;	:	,	.	<<	>>	!	'	"		
				{	}	[	]		&	(	)

٦-٦-٤- الكلمات المحجوزة في لغة C++

Key Words

تستخدم لغة C++ مجموعة الكلمات المحجوزة نفسها التي تستخدمها لغة C ، وتضيف إليها عدداً من الكلمات الأخرى التي تجسد المبادئ الجديدة في لغة C++ ، ونبين في الجدول (٦-٢) مجموعة الكلمات المحجوزة لكل من لغتي C و C++ وسوف نشير إلى تلك الكلمات التي أضافتها لغة C++ .

	asm	inline	(c++)	union
	auto		int	unsigned
	break		long	virtual (c++)
	case	new	(c++)	void
catch	(c++)	operator	(c++)	volatile
	char	private	(c++)	while
class	(c++)	protected	(c++)	
	const	public	(c++)	
	continue		register	
	default		return	
delete	(c++)		shorts	
	do		signed	
	double		sizeof	
	else		static	
	enum		struct	
	extern		switch	
	float	template	(c++)	
	For	this	(c++)	
friend	(c++)	throw	(c++)	
	goto	try	(c++)	
	if		typedef	

الجدول (٦-٢) - الكلمات المحجوزة في لغة C++

٦-٧ - المتحولات والثوابت variables and constants

من المعروف أن البرمجة هي ليست فقط في طباعة بعض الأسطر على الشاشة، بل هي أبعد من ذلك بكثير، لذلك من أجل كتابة برامج تقوم بتنفيذ بعض المهام وحل مسائل متنوعة، فإننا نحتاج إلى التعرف إلى مفهوم المتحولات والثوابت ومعرفة أنواع المعطيات .

٦-٧-١ - أنواع المعطيات الأساسية في لغة C++

تحتوي لغة C++ على مجموعة من أنواع المعطيات كغيرها من لغات البرمجة ، حيث توجد أنواع أساسية وأنواع مشتقة من الأنواع الأساسية ويمكن تلخيصها في الجدول (٦-٣) . هنالك ثمانية أنواع من المعطيات الأساسية في C++ ، واحد منها يمثل المحارف وأخرى تمثل قيمة منطقية وثلاثة تمثل أعداداً صحيحة ، وثلاثة أخرى تمثل أعداداً حقيقية.

النوع	الوصف	الحجم
char	محرّف (أحرف)	1 byte
short	أعداد صحيحة قصيرة	2 bytes
int	أعداد صحيحة عادية	4 bytes
long	أعداد صحيحة طويلة	8 bytes
float	أعداد حقيقية عادية	4 bytes
double	أعداد حقيقية مضاعفة	8 bytes
long double	أعداد حقيقية ضخمة	10 bytes
bool	قيمة منطقية true/false	1 bytes

الجدول (٦-٣) - يبين أنواع المعطيات الأساسية

٦-٧-٢- المتحولات (المتغيرات) variables

المتحول هو موضع من الذاكرة مخصص لتخزين قيمة محددة، مثلاً لنتعرف على المتغيرات التالية:

$a = 5 ;$

$b = 2 ;$

$a = a + 1 ;$

$r = a - b ;$

نلاحظ أن المتحول a يأخذ القيمة الصحيحة 5 ، بينما المتحول b يأخذ القيمة الصحيحة 2 ، بعد ذلك يتم تغيير قيمة المتحول a من القيمة القديمة (5) إلى القيمة الجديدة (6) ، وفي النهاية يتم إسناد حاصل طرح قيمة المتحول b من قيمة المتحول a ، إلى متحول جديد r .

٦-٧-٢-١- تسمية المتحولات Naming of variables

كل متحول يحتاج إلى اسم يميزه عن متحول آخر وعند تسمية أي متحول يجب أن نراعي بعض الشروط (القواعد) تجنباً للوقوع في أخطاء برمجية عند تنفيذ البرامج بلغة C++ ، ويجب إتباع القواعد التالية عند تسمية المتحولات :

١. يجب أن يبدأ اسم المتحول بحرف من حروف اللغة الانكليزية الصغيرة أو

الكبيرة ، ثم تتوالى بعد ذلك الحروف أو الأرقام ويجب ألا يحتوي على أي

رمز خاص سوى الشرطة التحتية (_).

٢. من الممكن أن يكون طول اسم المتحول حتى 32 حرفاً وإذا زاد عن ذلك

لا يأخذ بعين الاعتبار من قبل مترجم اللغة (Compiler).

٣. لغة C++ تفرق بين الأحرف الكبيرة والصغيرة لذلك يجب عدم استخدام الأحرف الكبيرة والصغيرة لاسم المتحول نفسه، لأن المترجم يفرق بينها، مثلاً، المتحول a يختلف عن المتحول A .
٤. يجب أن يكون اسم المتحول فريداً في الكتلة التي عُرف بها .
٥. يجب ألا يكون كلمة محجوزة مثل for ، أو أي قيمة منطقية true, false ، أو الكلمة null أو أي كلمة من الكلمات المحجوزة في لغة C++ .
٦. يفضل أن يبدأ اسم المتحول بحرف صغير ، وفي حال كان الاسم مؤلفاً من أكثر من كلمة يفضل جعل الحرف الأول من كل كلمة (ما عدا أول كلمة) حرفاً كبيراً مثل : isGoodColor .

٦-٧-٢-٢ - مجال المتحولات

إن مجال المتحول هو المنطقة التي يمكن فيها استخدام المتحول ، ويحدد مجال المتحول متى يقوم النظام بإنشاء المتحول ومتى يقوم بتحريره من الذاكرة . وإن مكان توضع المتحول البرمجي داخل النص البرمجي هو الذي يحدد مجاله ، وذلك حسب إحدى التصنيفات التالية :

١. **المتحولات الأعضاء Members Variable** : هي معطيات أعضاء من صف ، ويتم التصريح عنها داخل الصف ولكن خارج أي تابع أو بان ، ومجال هذا النوع من المتحولات هو كامل الصف (انظر إلى مقرر الرمجة (1)) .
٢. **المتحولات المحلية Local Variables** : يتم التصريح عنها داخل الكتلة ، ومجالها يمتد من نقطة التصريح عنها ولغاية نهاية الكتلة التي صُرحَت فيها .

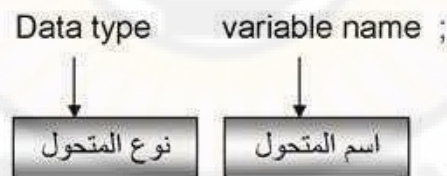
٣. الوسطاء لتابع (Parameters function) : هي متحولات يتم استخدامها في التتابع أو البواني من أجل تمرير القيم من/إلى التابع أو الباني و أن مجال المتحولات هو كامل التابع .

يمكن تهيئة المتحولات الأعضاء والمتحولات المحلية مباشرة أثناء التصريح عنها :
int x=200 ; long y=300 ;
وإن الوسطاء لا يمكن إسناد قيم لها مباشرة ، بل يجب تمرير قيمها من خلال النص البرمجي .

٦-٧-٢-٣- التصريح عن المتحولات

Declaration of variables

كل متحول يُستخدم في برنامج بلغة C++ لا بد من التصريح عنه ويتم ذلك في أي مكان من البرنامج ولكن بشرط أن يكون التصريح قبل العبارات البرمجية التي ستستخدم هذا المتحول يبين الشكل (٦-١) الشكل العام للتصريح عن المتحولات .



الشكل (٦-١) - يبين الشكل العام للتصريح عن المتحولات

في لغة C++ يوجد العديد من المتحولات التي يمكن استخدامها والتصريح عنها وفيما يلي سنذكر أهم هذه المتحولات وكيفية التصريح عنها .

١ - التصريح عن المتحولات المحرفية char

للتصريح عن متحول حرفي يُستخدم الكلمة المحجوزة char ، ويتم تخزين المحارف في متحولات من النوع char ، والعبارة التالية توضح ذلك :

```
char ch ;
```

تنشئ مساحة في الذاكرة لمحرف وتسميه ch وحجمه 1Byte ، ولتخزين محرف ما في هذا المتحول نكتب :

```
ch = 'z'
```

ودائماً تكون المحارف الثابتة كـ 'a' و 'b' محصورة بعلامتي اقتباس مفردة.

مثال:

```
char a ; //a is a character variable
```

```
char b ; //b is a character variable
```

Or

```
char a , b ;
```

يمكن استعمال المتحولات من النوع char لتخزين أرقام كاملة بدلاً من محارف فمثلاً يمكننا كتابة:

```
ch = 2 ;
```

لكن نطاق القيم الرقمية التي يمكن تخزينها في النوع char يتراوح بين -128 إلى 127 لذا فإن هذه الطريقة تعمل مع الأرقام الصغيرة فقط.

ملاحظة

عندما نريد إدخال مجموعة محارف إلى برنامج ما بلغة C++ ، يجب أن نصرح عن المتحول على شكل مصفوفة من المحارف وذلك كما يلي:

char name[20]; char address[50];

٢- التصريح عن المتحولات الصحيحة

تمثل المتحولات الصحيحة أرقاماً كاملة أي قيماً يمكن تعدادها ، كعدد الأشخاص أو الأيام أو عدد الصفحات مثلاً ، ولا يمكن أن تكون الأعداد الصحيحة أرقاماً ذات فاصلة عشرية ولكنها يمكن أن تكون سالبة، وهنالك ثلاثة أنواع من الأعداد الصحيحة في C++ ، هي :

- short : عدد صحيح قصير
- int : عدد صحيح عادي
- long : عدد صحيح طويل

وهي تحتل مساحات مختلفة في الذاكرة ، والجدول (٤-٦) يبين هذه الأنواع والمساحة التي تأخذها في الذاكرة ومجال الأرقام التي يمكن أن تأخذها.

اسم النوع	الحجم	المجال
char	1 byte	128- إلى 127
Short	2 byte	إلى -32,768 32,767
int	4 byte	إلى -32768 +32798
Long	8 byte	إلى -2,147,483,648 2,147,483,647

الجدول (٤-٦) - يبين أنواع المتحولات الصحيحة

- للتصريح عن متحول صحيح من النوع **int** .

مثال:

```
int a ; //a is an integer variable
```

```
int b ; //b is an integer variable
```

Or

```
int a , b ;
```

- للتصريح عن متحول صحيح من النوع **long** .

مثال:

```
long a ; // a is a long integer variable
```

```
long b ; // b is a long integer variable
```

Or

```
long a , b ;
```

٣-التصريح عن متحول حقيقي

يتم استعمال المتحولات الحقيقية لتمثيل قيم يمكن قياسها كالأطوال أو الأوزان، ويتم تمثيل الأعداد الحقيقية عادة برقم كامل على اليسار مع فاصلة عشرية وكسر على اليمين، وهناك ثلاثة أنواع من الأعداد الحقيقية في أنظمة التشغيل الشائعة الاستعمال ، وأشهر أنواع الأعداد الحقيقية هي من النوع **double** والذي يتم استعماله لمعظم توابع **C++** الرياضية ، ويتطلب النوع **float** ذاكرة أقل من النوع **double** ، و يوضح الجدول (٦-٥) هذه الأنواع والحجم الذي تأخذه في الذاكرة.

اسم النوع	الحجم
float	4byte
double	8byte
long double	10byte

الجدول (٥-٦) - يبين أنواع المتحولات الحقيقية

- للتصريح عن متحول حقيقي من النوع **float** :

مثال:

float a ; // a is a real variable

float b ; // b is a real variable

Or

float a , b ;

- للتصريح عن متحول حقيقي من النوع **double** :

مثال:

double a ; // a is a big real variable

double b ; // b is a big real variable

Or

double a , b ;

ملاحظة

يجب أن نميز بين المتحولات العامة Global والمتحولات المحلية Local من خلال وضع عملية التعريف داخل البرنامج حيث:

١- المتحولات التي يتم تعريفها قبل بداية البرنامج (الدالة main) تمثل متحولات عامة.

٢- المتحولات التي يتم تعريفها داخل الدالة main أو داخل أي تابع تعد متحولات محلية تستخدم فقط داخل الجزء المعرفة فيه.

مثال (4-6)

المثال التالي يوضح لنا كيفية استخدام التصريح عن بعض المتحولات ضمن البرنامج .

البرنامج:

```
# include <iostream.h>
int main() {
    //Declaring variables
    int a , b, sum ;
    //Process
    a = 5 ;
    b = 2 ;
    a = a + 1 ;
    sum = a + b ;
    cout<<"Sum = "<< sum << endl ;
    return 0 ;
}
```


وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT  
Sum = 8  
Press any key to continue
```

ملاحظة

عند إعطاء قيمة أولية لمتغير ما a ، على سبيل المثال (القيمة 0) ، فإننا نصرح عن ذلك كما يلي:

```
int a=0;
```

٦-٧-٣ - الثوابت constants

الثابت هو مقدار لا يتغير أثناء تنفيذ البرنامج ويأخذ قيمة ثابتة ، وفي لغة C++ نميز عدة أنواع من الثوابت ، منها :

١- الثوابت الصحيحة

الثوابت الصحيحة هي الثوابت التي تأخذ قيمة صحيحة محددة.

مثال:

1776 , 707 , -273

٢- الثوابت الحقيقية

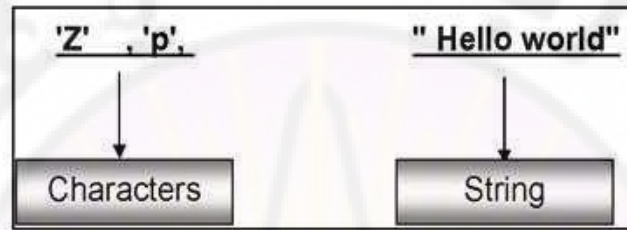
الثوابت الحقيقية هي الثوابت التي تأخذ قيمة حقيقية محددة.

مثال:

3.14159 , 6.02E23 , 1.6E-19 , 3.0

٣- الثوابت المحرفية وثوابت السلاسل المحرفية

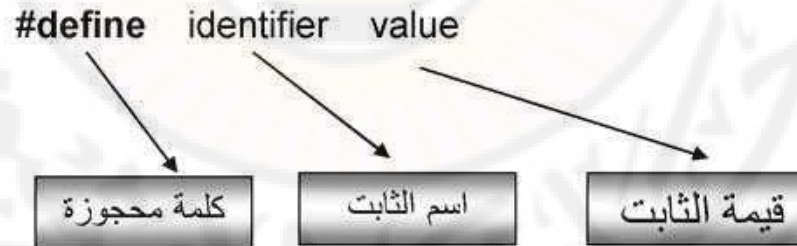
الثوابت المحرفية هي الثوابت التي تأخذ قيمة محرفية (حرف أو سلسلة محارف) محددة، ويبين الشكل (٢-٦) ذلك .
مثال:



الشكل (٢-٦) - يبين الثوابت والسلاسل المحرفية

٦-٨-٣-١- تعريف الثوابت برمجياً

يبين الشكل (٣-٦) الشكل العام لتعريف الثوابت برمجياً .



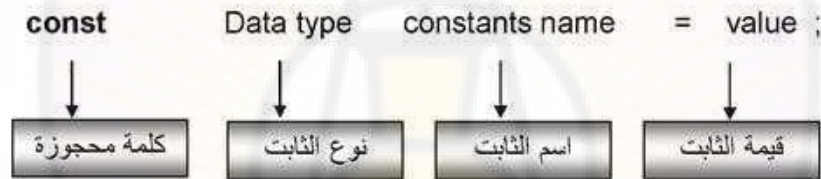
الشكل (٣-٦) - يبين الشكل العام لتعريف الثوابت برمجياً

أمثلة:

```
#define PI 3.14159  ثابت حقيقي
#define newline '\n'  ثابت محرفي
#define w 100  ثابت صحيح
```

٦-٨-٣-٢ - التصريح عن الثوابت

يتم التصريح عن الثوابت باستخدام الكلمة **const** ويبين الشكل (٦-٦) الشكل العام للتصريح عن الثوابت برمجياً .



الشكل (٦-٦) - يبين الشكل العام للتصريح عن الثوابت برمجياً

أمثلة:

```
const float PI = 3.14159 ;
const int w = 100 ;
const char newline = '\n' ;
const z = 12440 ;
```

ملاحظة

في الحالة التي لا يتم فيها ذكر نوع الثابت فإن المترجم يُعد هذا الثابت من النوع **.int**

٦-٨- تعليمات الإدخال والإخراج الأساسية

Basic Input /output statements

تعليمات الإدخال والإخراج هي عبارة عن التعليمات المتضمنة في المكتبة `. iostream`

١- تعليمة الإخراج `cout`

تُستخدم التعليمة `cout` مع العملية `<<` لطباعة المخرجات على الشاشة ، والصيغة العامة لتعليمة الإخراج `cout` تأخذ ثلاث أشكال ، وهي :
- لطباعة سلسلة على الشاشة.

- `cout <<"output sentence";`

مثال:

```
cout <<" Welcome " ;
```

- لطباعة قيمة متحول على الشاشة.

- `cout <<var_name;`

مثال:

```
int x=10;  
cout <<x;
```

- لطباعة نص و قيمة متحول معاً على الشاشة.

- `cout <<"message"<<value;`

مثال:

```
int age=20;  
cout <<" Hello, I am "<<age<<" years old ";
```


٢- تعليمة الإدخال cin

من أجل إدخال قيم للمتحولات عن طريق لوحة المفاتيح يُستخدم في لغة C++ التعليمة cin مع العملية >> وهذه التعليمة لها الشكل العام التالي:

```
cin>>v1>>v2>>v3>>...>>vn;
```

حيث: v1,v2,v3,...,vn ، أسماء المتحولات التي نريد إدخال قيماً لها.

فالعبرة:

```
cin>>integer1
```

هذه العبارة تخزن الرقم الذي يكتبه المستخدم من لوحة المفاتيح في متحول يدعي integer1 ، تمثل العبارة cin (والذي تلفظ كـ C in) الإدخال من لوحة المفاتيح، ويأخذ عامل الحصول (>>) get from الأشياء الموضوعه على يساره ويضعها في المتحول الموجود على اليمين ، وعند تنفيذ هذه العبارة ينتظر البرنامج أن يكتب المستخدم رقماً من النوع integer ويضغط على مفتاح Enter، ويتم إسناد القيمة التي أدخلها المستخدم إلى المتغير integer1 .

أمثلة:

١- إدخال محرف من لوحة المفاتيح:

```
char a;  
cin>>a;
```

٢- إدخال قيمة صحيحة من لوحة المفاتيح:

```
int age;  
cin>>age;
```

٣- إدخال قيمة صحيحة وقيمة حقيقية من لوحة المفاتيح:

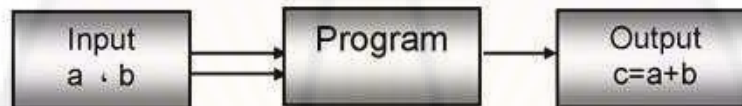
```
int x;  
float y;  
cin>>x>>y;
```

مثال (5-6)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة حاصل جمع عددين صحيحين a و b .

الحل:

قبل البدء بكتابة البرنامج لنحاول فهم المسألة (معرفة الدخل والخرج) ويتم ذلك كما هو مبين في الشكل (٦-٦) .



الشكل (٦-٦) - يبين مخطط سير البرنامج

البرنامج:

```
//This program computes the sum of numbers
```

```
# include <iostream.h>
```

```
int main( )
```

```
{
```

```
int a, b,sum;
```

```
cout<<"Enter the numbers a and b ";
```

```
cin>>a>>b;
```

```
sum=a+b;
```

```
cout<<"sum="<<sum<<endl;
```

```
return 0;
```

```
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

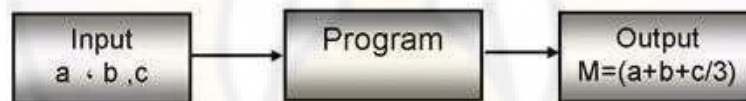
Enter the numbers a and b 10 20

sum=30

Press any key to continue

مثال (6-6)

اكتب برنامجاً بلغة الـ C++ يقوم بحساب وطباعة المتوسط الحسابي لثلاثة أعداد صحيحة مدخلة a , b , c من لوحة المفاتيح، يبين الشكل (٦-٧) مخطط سير البرنامج .



الشكل (٦-٧) - يبين مخطط سير البرنامج

البرنامج:

```
//This program to calculate the average of three numbers
```

```
# include <iostream.h>
```

```
int main( )
```

```
{
```

```
int a , b , c ;
```

```
float sum , m ;
```

```

cout<<"Enter the numbers " ;
cin >> a >> b >> c ;
sum =(float)(a + b + c) ;
m = sum / 3 ;
cout<<" m = "<< m << endl ;
return 0;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT
Enter the numbers a and b 10 20
sum=30
Press any key to continue

ملاحظة

لا بد من التحويل إلى أنه يمكن التحويل بين أنواع المعطيات العددية، فعلى سبيل

المثال ، استخدمنا الكلمة المحجوزة **float** في العبارة البرمجية:

```
sum =(float)(a + b + c) ;
```

من أجل تحويل نوع المجموع (a + b + c) من النوع **int** إلى النوع **float**

وذلك لأن نوع المتغير **sum** هو **float** ، وفي حالة الكسر ، يمكن تحويل نوع

البسط أو نوع المقام إلى النوع المطلوب.

مثال (7-6)

اكتب برنامجاً بلغة الـ C++ يقوم بحساب وطباعة مساحة دائرة نصف قطرها r .
البرنامج:

```
//This program computes the square of a circle
# include <iostream.h>
#define PI 3.14159
int main( )
{
    double r , s ;
    cout<<"Enter the value of r " ;
    cin >> r ;
    s = PI*r*r;
    cout<< " s = "<< s << endl ;
    return 0;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```
Enter the value of r 5.5
s = 95.0331
Press any key to continue
```

ملاحظة

يمكن إعادة كتابة البرنامج السابق باستخدام مفهوم التصريح عن الثابت باستخدام الكلمة المحجوزة `const`، حيث أن الشكل الأول استخدمنا فيه تعريف الثابت وليس التصريح عن الثابت.

مثال (6-8)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة مساحة دائرة نصف قطرها `r`، وذلك باستخدام التصريح عن الثابت `PI` وليس تعريف الثابت.

البرنامج:

```
//This program computes the square of a circle
# include <iostream.h>
int main( )
{
    const double PI= 3.14159;
    double r , s ;
    cout<<"Enter the value of r " ;
    cin >> r ;
    s = PI*r*r;
    cout<< " s = "<< s << endl ;
    return 0;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of r 5.5

s = 95.0331

Press any key to continue

مثال (9-6)

إذا كان لدينا القيم التالية:

n='a' , x=7 , g='b'

والمطلوب : اكتب برنامجاً بلغة C++ من أجل الحصول على المخرجات على

الشكل التالي :

Name	Mark	Grade
a	7	b
c++ language		

البرنامج:

```
//This program uses Escape Characters
```

```
# include <iostream.h>
```

```
int main( )
```

```
{
```

```
    int x;
```

```
    char n,g;
```

```
    n='a';
```

```
    x=7;
```

```
    g='b';
```

```

cout<< "Name\t\tMark\t\tGrade \n";
cout<<n<<"\t\t"<< x<<"\t\t"<< g<< "\n";
cout<< "\t\tC++ language \n";
return 0;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT		
Name	Mark	Grade
a	7	b
C++ language		
Press any key to continue		

٦-٩- العوامل الحسابية والمنطقية والعلائقية

العامل Operator هو عبارة عن رمز رياضي يقوم بإجراء عملية معينة على معاملاته ويمكن أن تكون عملية حسابية أو منطقية أو عملية مقارنة ، وفي هذه الفقرة سنتعرف على أنواع العوامل التي توفرها لغة C++ .

٦-٩-١- العوامل الحسابية Arithmetic Operators

أكثر البرامج في لغة C++ تستخدم العمليات الحسابية ، مثل : الجمع والطرح وغيرها، ويبين الجدول (٦-٧) العمليات الحسابية ورموزها المقابلة في لغة C++

العملية	الرمز الرياضي للعملية	رمز العملية بلغة C++	مثال
الجمع	+	+	a + b
الطرح	-	-	a - b
الضرب	× or .	*	a * b
القسمة	÷ ، ____	/	a / b
باقي القسمة الصحيح	a mod b or %	%	a % b

الجدول (٦-٧) يبين العمليات الحسابية ورموزها المقابلة في لغة C++

العوامل الأربعة الأولى تتجزأ أعمالاً مألوفة لدينا، أما عامل باقي القسمة % المسمى أيضاً المعامل modulus، يتم استعماله لحساب باقي قسمة عدد صحيح على عدد آخر، لذلك فالتعبير 3%20 يساوي 2 . تسمى هذه العوامل الحسابية بالعوامل الثنائية لأنها تعمل على قيمتين ، ويمكن استعمال أكثر من عامل في تعبير رياضي واحد، فمثلاً التعبير:

$$C=(f-32)*5/9;$$

يحول درجة الحرارة من مئوية إلى فهرنهايت. (استعملت الأقواس لكي يتم تنفيذ الطرح أولاً بالرغم من أولويته المتدنية، ويشير المصطلح أولوية Precedence إلى ترتيب تنفيذ العوامل، فالعاملان * و / لهما أولوية أعلى من العاملين + و - ، وهذا ما سنراه لاحقاً بعد أن نتعرف على بقية عوامل C++ .

التعبير الحسابي هو عبارة عن مجموعة من العمليات الحسابية التي تربط الأعداد والثوابت والمتغيرات والتوابع، ويبين الجدول (٦-٨) بعض التعبيرات الحسابية.

الشكل الرياضي للتعبير الحسابي	الشكل البرمجي للتعبير الحسابي
$\frac{x+y}{x-y} + c$	$(x + y) / (x - y) + c$
$2 \cdot r - 6.5$	$2 * r - 6.5$
$D = b^2 - 4ac$	$D = b * b - 4 * a * c$
$y = 2x^2 + 5x - \frac{7}{3} + 2x^3$	$y = 2 * x * x + 5 * x - \frac{7}{3} + 2 * x * x * x$

والجدول (٦-٨) - يبين بعض التعبيرات الحسابية

٦-٩-٢ - العوامل العلائقية (المقارنة)

Relational Operators

تقارن العوامل العلائقية قيمتين، وتكون نتيجة التنفيذ إما صحيحة True أو خاطئة False حسب نتيجة المقارنة مع القيمتين ، ويوضح الجدول (٦-٨) العوامل العلائقية (المقارنة) وما يكافئها في لغة C++ .
عند استخدام العمليات ذات الإشارات المزدوجة (=, <=, >=) نتجنب ترك أي فراغ بين علامة المساواة والعلامة الأخرى المرافقة لأن ذلك قد يتسبب ببعض الأخطاء القواعدية البرمجية.

مثال: لنفرض أنه لدينا المتغيرين :

`int i= 5, j= -3;`

والجدول (٩-٦) يوضح لنا مجموعة من تعابير المقارنة والنتائج المناظرة لها.

العملية	الرمز الرياضي للعملية	رمز العملية بلغة C++	مثال
المساواة	=	==	a == b ?
أكبر من	>	>	a > b ?
أكبر أو تساوي	$\geq$	>=	a >= b ?
أصغر من	<	<	a < b ?
أصغر أو تساوي	$\leq$	<=	a <= b ?
عدم المساواة	!=	!=	a != b ?

الجدول (٨-٠٦) - يوضح عوامل المقارنة وما يكافئها في لغة C++

النتيجة	تعابير المقارنة
TRUE	i != j
FALSE	j * 2 == i * 2
FALSE	i - j < i + j
TRUE	i - 3 == j + 5
TRUE	j < i > i + j

الجدول (٩-٦) يوضح مجموعة من تعابير المقارنة نتائجها

٦-٩-٣-العوامل المنطقية

Logical Operators

تُستخدم العوامل المنطقية لربط أكثر من تعبير منطقي بسيط مع بعضها البعض، ويوضح الجدول (٦-١٠) العوامل المنطقية وما يكافئها في لغة C++.

العملية	الرمز الرياضي للعملية	رمز العملية بلغة الـ C++	مثال
و	AND	&&	(a>b) && (a>c)
أو	OR		(a>b) (a>c)
النفي	NOT	!	!(a>b)

الجدول (٦-١٠) - يوضح العوامل المنطقية وما يكافئها في لغة C++

يوضح الجدول (٦-١١) النتائج المقابلة لبعض التعبيرات المنطقية .

قيمة الشرط المركب	قيم الشروط البسيطة	تعبير المقارنة
False	True && False	((5 == 5) && (3 > 6))
True	True False	((5 == 5) (3 > 6))

الجدول (٦-١١) - يوضح النتائج المقابلة لبعض التعبيرات المنطقية .

٦-٩-٤ - المعامل الشرطي الثلاثي الحدود

Conditional Operator (?)

يوجد في لغة جافا معامل خاص يسمح باختبار قيمة الشرط المنطقي واستخدام واحد من تعبيرين حسب قيمة هذا الشرط وهو المعامل (?) وله الصيغة التالية :

(Condition) ? expression – if – true : expression – if - else

يُستخدم المعامل الشرطي (?) من أجل اختبار شرط معين ويُعيد قيمة إذا تحقق الشرط أو يُعيد قيمة أخرى إذا لم يتحقق الشرط ، ويمكن كتابته بصيغتين :

1 : (condition) ? result1 : result2

2 : (condition ? result1 : result2)

حيث :

condition - الشرط المنطقي.

result1 - القيمة المعادة إذا تحقق الشرط.

result2 - القيمة المعادة إذا لم يتحقق الشرط.

مثال :

y = (x < 5) ? 44 : 55 ;
y = (x < 5 ? 44 : 55) ;

مثال (6-10)

أكتب برنامجاً بلغة ++C يقوم بإيجاد وطباعة القيمة العظمى لثلاثة أعداد صحيحة يتم إدخالها من لوحة المفاتيح وذلك باستخدام المعامل الشرطي (?).

البرنامج:

//This program uses Conditional Operator (?)

```

#include <iostream.h>

int main ( )
{
    int a,b,c,d,e;
    cout<<"Enter the values of a,b and c: ";
    cin>>a>>b>>c;
    d=(a > b ) ? a : b ;
    e=(c > d ? c : d );
    cout<<"max = "<< e <<endl;
    return 0;
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the values of a,b and c: 7 5 4

max = 7

Press any key to continue

٦-٩-٥ - عوامل الإسناد المركبة

Compound Assignment Operators

يتوفر في لغة C++ عدداً من عمليات الإسناد، فبالإضافة لعملية الإسناد العادية التي تستخدم فيها الإشارة (=) ، يوجد طرق مختصرة أخرى تسمى عمليات الإسناد المركبة ، ويمكن توضيحها من خلال الجدول (٦-١٢).

عملية الإسناد المختصرة	عملية الإسناد
$a+=b$	$a=a+b$
$a-=b$	$a=a-b$
$a*=b$	$a=a*b$
$a/=b$	$a=a/b$
$a\%=b$	$a=a\%b$

(٦-١٢) - يوضح عمليات الإسناد المركبة

٦-٩-٦ - عمليات الزيادة والنقصان

Increment and Decrement Operators

إن عامل الزيادة هو (++) ويعني زيادة المتحول بمقدار واحد، بينما عامل النقصان (--) يعني إنقاص قيمة المتحول بمقدار واحد. والزيادة والنقصان يمكن أن يكون من اليسار (الأمام) أو اليمين (الخلف)، أي يمكن وضع ++ أو -- قبل المتحول (الزيادة أو النقصان اليسار) أو يمكن وضع ++ أو -- بعد المتحول (الزيادة أو النقصان من اليمين). من حيث المبدأ لا يوجد فرق بين الزيادة (النقصان) من اليسار أو الزيادة (النقصان) من اليمين كقيمة وإنما

الاختلاف في الاستخدام ، فعلى سبيل المثال التعبير ++I يعني إضافة القيمة واحد إلى قيمة المتحول I قبل استخدام المتحول I ، في حين أن التعبير I++ يعني استخدام القيمة الحالية للمتحول I ثم إضافة واحد إلى قيمة المتحول I .
□ الزيادة بمقدار واحد (++)

مثال

```
int A ;  
A++ ; // A = A + 1 ;  
cout << A++ ;
```

• النقصان بمقدار واحد (--)

مثال

```
int B ;  
B-- ; // B = B - 1 ;  
cout << B-- ;
```

مثال (11-6)

اكتب برنامجاً بلغة C++ يقوم بتعريف ثلاثة متحولات من نوع int كما يلي :

```
int k = 4 , i , j ;
```

- أجرِ عملية الزيادة من اليسار للمتحول k ، ثم أجرِ الزيادة من اليمين للمتحول k ، ثم اطبع قيمة k .
- اجعل $i = k++$ ثم اطبع قيمة i و k .
- اجعل $i = ++k$ ثم اطبع قيمة i و k .
- ماذا تلاحظ ؟

البرنامج


```

#include<iostream.h>
void main ( ) {

int k = 4 , i , j ;

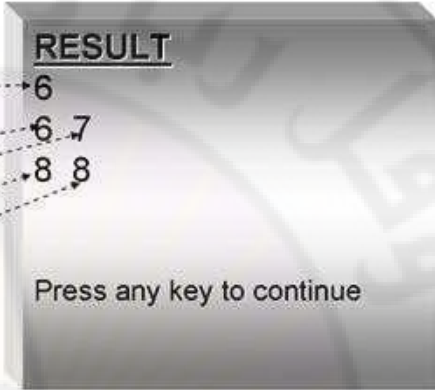
++k ;
k++ ;
cout << k << endl;

i = k++;
cout << i << " " << k << endl;

j = ++k;
cout << j << " " << k << endl;

} //End main

```



RESULT

6

6 7

8 8

Press any key to continue

مثال (11-6)

ما هو الخرج الناتج عند تنفيذ البرنامج التالي:

البرنامج:

//This program uses increment (++) and decrement (--) Operators

```
#include <iostream.h>
```

```
int main ( )
```

```
{
```

```
int i=2,j=4,k=6;
```

```
i+=5+i;
```

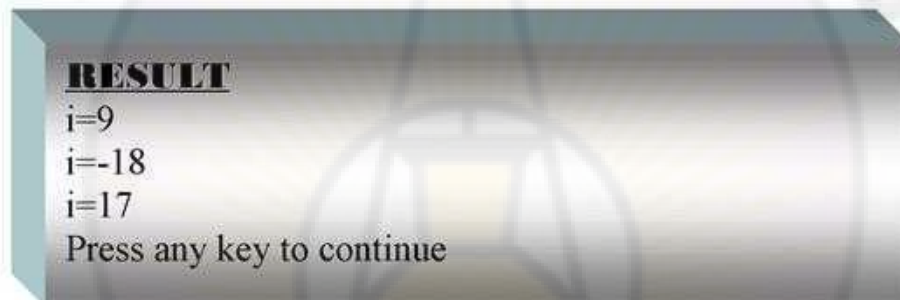
```
cout<<"i"<<i<<endl;
```

```

i*=j++ -k;
cout<<"i="<<i<<endl;
i=-j++ *++k;
cout<<"i="<<i<<endl;
return 0;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



٦-١٠- أولوية تنفيذ العمليات الحسابية والمنطقية والعلائقية

إن تنفيذ العمليات الحسابية والمنطقية تتم في مترجم C++ حسب قواعد محددة وأن أولويات تنفيذ العمليات الحسابية والمنطقية والمقارنة موضحة في الجدول (٦-١٣) .

إذا تساوت الأولويات في تعبير حسابي ما، فيتم التنفيذ من اليسار إلى اليمين أو من اليمين إلى اليسار (حسب العمليات).

العملية	رمز العملية	رقم الأولوية	جهة التنفيذ عند التساوي
فك الأقواس	() أو []	1	من اليسار
التوابع	Sin(),.... sqrt(),....	2	من اليسار
الزيادة أو النقصان	++ أو --	3	من اليمين
العمليات الحسابية	* / %	4	من اليسار
العمليات الحسابية	+ -	5	من اليسار
عمليات المقارنة	> >= < <=	6	من اليسار
عمليات المقارنة	= = !=	7	من اليسار
العمليات المنطقية	&&	8	من اليسار
العمليات الشرطية	?:	9	من اليمين
عمليات الإسناد	= += -= *= /= %=	10	من اليمين

الجدول (٦-١٣) - يوضح أولوية تنفيذ العمليات الحسابية والمنطقية والمقارنة

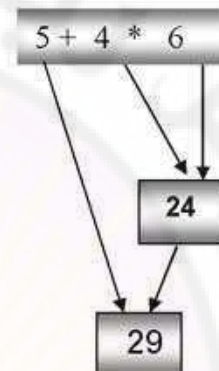
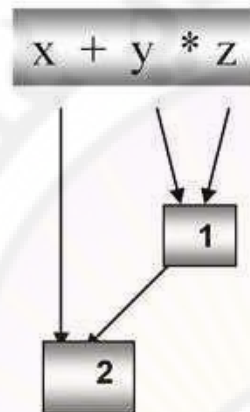
=====

=====

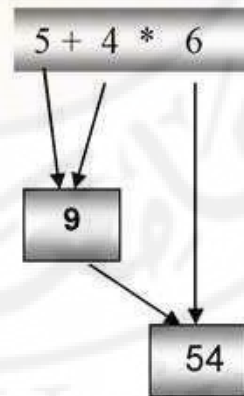
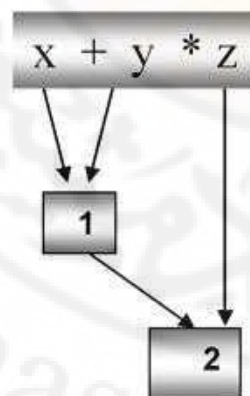
=====

مسائل عامة محلولة

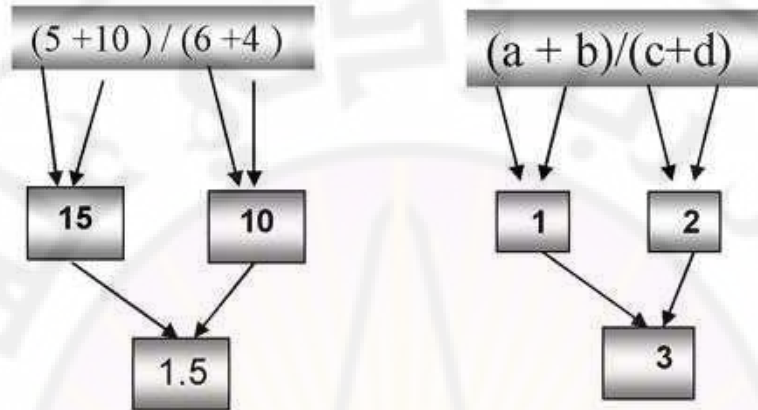
١- اكتب تسلسل تنفيذ التعابير التالية :



لو لم يكن هناك أولوية في التنفيذ فقد نحصل على الجواب الخاطئ التالي:



٢- اكتب تسلسل تنفيذ التعابير التالية :



٣- ما هو الخرج الناتج عند تنفيذ العبارة البرمجية التالية:

$$\begin{aligned}
 & (5 + 3) * 2 + 4 \% (3 - 1) \\
 & \quad \downarrow \qquad \qquad \qquad \downarrow \\
 & \underline{8} * 2 + 4 \% (\underline{3} - \underline{1}) \\
 & \quad \downarrow \qquad \qquad \qquad \downarrow \\
 & \underline{8} * \underline{2} + 4 \% \underline{2} \\
 & \quad \downarrow \qquad \qquad \qquad \downarrow \\
 & \underline{16} + 4 \% \underline{2} \\
 & \quad \downarrow \qquad \qquad \qquad \downarrow \\
 & \underline{16} + \underline{0} \\
 & \quad \downarrow \\
 & \underline{16}
 \end{aligned}$$

الخرج النهائي للعبارة البرمجية السابقة هو : 16

٤- ما هو الخرج الناتج عند تنفيذ العبارة البرمجية التالية:

$$(2 + 5 > 6 - 7 = 2 * 3)$$

الحل :

$$(2 + 5 > 6 - 7 = 2 * 3)$$

$$(2 + 5 > 6 - 7 = \underline{6})$$

$$(\underline{7} > 6 - 7 = 6)$$

$$(7 > \underline{-1} = 6)$$

$$(\underline{1} = 6)$$

$$\underline{0}$$

الخرج النهائي للعبارة البرمجية السابقة هو : 0

=====

=====

=====

مسائل عامة غير محلولة

- ١- حدد ما إذا كانت العبارات الآتية صحيحة أم خطأ:
 - التعليقات تجبر الحاسوب على طباعة النص الذي يلي // على الشاشة عند تنفيذ البرنامج.
 - تتابع الهروب \n يجبر المؤشر على الانتقال إلى سطر جديد.
 - برنامج C++ والذي يقوم بطباعة ثلاثة أسطر على الشاشة يجب أن يحتوى على ثلاث عبارات تستعمل cout.

٢- ما هو الخرج من العبارة الآتية:
`cout << "\n **\n ***\n ****\n";`

- ٣- اكتب عبارة C++ صحيحة تقوم بالآتي:
 - تعريف المتغيرات x، y، z و result لتكون من النوع int.
 - الطلب من المستخدم إدخال ثلاثة أرقام صحيحة.

- ٤- حدد ما إذا كانت العبارات الآتية صحيحة أم خطأ:
 - يجب الإعلان عن المتغيرات قبل استعمالها في البرنامج.
 - يجب تحديد نوع المتغيرات عند الإعلان عنها.
 - لا تفرق C++ بين المتغيرات Number و number.

٥- ما هو الخرج الناتج عند تنفيذ العبارات البرمجية التالية المكتوبة بلغة C++:

- 1) $2 * 5 * 5 + 3 * 5 + 7$
- 2) $(3 * 9 * (3 + (9 * 3 / (3))))$

3)

$$2 \% 2 + 2 * 2 - 2 / 2$$

٦- اكتب برنامجاً بلغة الـ C++ يقوم بطباعة مايلي:

Welcome with C++.

This program

Is the third in C++.

٧- اكتب برنامجاً بلغة الـ C++ يقوم بحساب وطباعة مساحة مستطيل بعده L

و W .

٨ - اكتب برنامجاً بلغة الـ C++ يقوم بحساب وطباعة مربع ومكعب أي عدد

صحيح a.





The background of the page features a large, faint watermark of the Damascus University logo. The logo is circular, with the university's name in Arabic 'جامعة دمشق' at the top and 'Damascus University' at the bottom. In the center is a stylized sun or star symbol with rays.

الفصل السابع

بنى التحكم و التكرار

Control and Looping Structures



بنى التحكم و التكرار

٧-١- مقدمة

في التطبيقات العملية نحتاج لاتخاذ بعض القرارات تبعاً لشروط معينة، ومن هنا ظهرت الحاجة لوجود طرق لجعل البرنامج قادراً على تغيير تسلسل تنفيذ التعليمات تبعاً للشروط المطلوبة وهو ما يعرف بالجمال الشرطية ، وكثيراً ما نحتاج في البرامج إلى تكرار أمر موجه للحاسوب عدداً من المرات، وتوفر لغة C++ عدة وسائل تمكن المبرمج من أداء هذا التكرار، وعادة ما تسمى هذه الوسائل " الحلقات التكرارية " ، ويوجد عدد من الحلقات التكرارية في لغة C++ ، وتعد أساليب الشرط والتكرار بمثابة القلب في جسم لغات البرمجة، وبدونها لا يمكن تنظيم أي برنامج. وتوفر لغة C++ للمبرمج عدداً من الأساليب والتوابع الفعالة، المتعلقة بهذا الشأن، وتمتاز هذه الأساليب بأنها أساليب بنيوية Structured أي يمكن تنظيم عمليات التحكم والتكرار فيها، بأسلوب ذاتي من بداية العمليات وحتى نهايتها دون تدخل من المبرمج أثناء هذه العمليات، للأشراف على التوجيه والتخطيط لكل خطوة من خطوات البرنامج ، وحيث أن جواب الشرط إما أن يكون صحيحاً true أو خاطئاً false ، فإن لغة C++ توفر مرونة كبيرة في استخدام عدد كبير من التوابع ، وفي توجيه البرنامج بطريقة فعالة وفائقة.

٧-٢- بنى التحكم

Control Structures

إن بنى التحكم هي عبارة عن بنى تتحكم في تنفيذ خطوات البرنامج من خلال شرط أو مجموعة شروط ، وإن عدم استخدام بنى التحكم يؤدي إلى جعل مترجم لغة C++ يقوم بتنفيذ تعليمات البرنامج بنفس الترتيب الذي ظهر في ملف

المصدر ، ويمكن استخدام بنى التحكم لتغيير مسار تنفيذ البرنامج إما بوضع تعليمات شرطية (تنفذ عند تحقيق شرط معين فقط) أو حلقات تكرارية (تكرار مجموعة من التعليمات عدداً من المرات) ، ونزودنا لغة ++C بعدد من تعليمات التحكم .

٧-٢-١ - التعليمات if

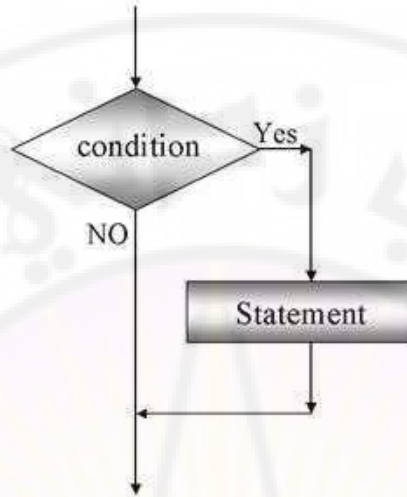
تستخدم التعليمات if في اختبار شرط ما، فإذا تحقق الشرط يتم تنفيذ تعليمة أو مجموعة من التعليمات والشكل التالي يوضح الصيغة العامة لهذه البنية في حالة تنفيذ تعليمة واحدة

```
if ( <condition> )
{
    <statement>;
}
```

حيث condition هو الشرط الواجب تحققه حتى يتم تنفيذ التعليمة statement ، ويمكن أن يكون هذا الشرط على شكل مقارنة بين قيمتين أو متحولين ، وهذه المقارنة يمكن توضيحها من خلال الجدول (٧-١)، ويبين الشكل (٧-١) المخطط التدفقي لبنية التعليمات if .

الشرط	معنى الشرط
if(a == b)	هل قيمة المتغير a تساوي قيمة المتغير b ؟
if (a<b)	هل قيمة المتغير a أصغر من قيمة المتغير b ؟
if(a<=b)	هل قيمة المتغير a أصغر أو تساوي قيمة المتغير b ؟
if(a>b)	هل قيمة المتغير a أكبر من قيمة المتغير b ؟
if(a>=b)	هل قيمة المتغير a أكبر أو تساوي قيمة المتغير b ؟
if(a!=b)	هل قيمة المتغير a لا تساوي قيمة المتغير b ؟

الجدول (٧-١) - يبين جدول المقارنة للتعليمات الشرطية



الشكل (١-٧) - يبين المخطط التدفقي للتعليلة if

مثال (1-7)

البرنامج الجزئي التالي يوضح كيفية تنفيذ تعليلة واحدة في حال تحقق الشرط .

```
if (grade >= 90)
```

```
    cout<<" Congratulations!\n";
```

```
    cout<<" Your grade is " << grade ;
```

وفي حالة تنفيذ مجموعة من التعليلات تكون الصيغة العامة لهذه البنية كما يلي :

```
if(condition)
```

```
{
```

```
    statement 1;
```

```
    statement 2;
```

```
    .....
```

```
    statement n;
```

```
}
```

حيث : condition هو الشرط الواجب تحققه حتى يتم تنفيذ التعليمات
statement 1, statement 2, ..., statement n

مثال (2-7)

البرنامج الجزئي التالي يوضح كيفية تنفيذ مجموعة من التعليمات في حال تحقق الشرط .

```
if( x== 100 )  
{  
    cout<<"x is ";  
    cout<<"100";  
}
```

مثال (3-7)

اكتب برنامجاً بلغة C++ يقوم بإدخال عدد حقيقي n ثم بطبع الجملة التالية:
" This number n is positive " إذا كان العدد المدخل n أكبر من الصفر.
البرنامج:

```
#include<iostream.h>  
  
void main() {  
    float n;  
    cout<<"Enter the number n";  
    cin>>n;  
    if (n>0)  
        cout<<"The number"<<n<<" is positive"<<endl;  
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```
Enter the number n: 8.5
The number 8.5 is positive
Press any key to continue
```

٧-٢ - ٢- if / else التعليمية

تستخدم التعليمية if / else في اختبار شرط ما ، فإذا تحقق الشرط يتم تنفيذ تعليمة أو مجموعة من التعليمات ، وإذا لم يتحقق الشرط فإنه يتم تنفيذ تعليمة أخرى أو مجموعة من التعليمات ، الصيغة العامة لهذه البنية كما يلي :

```
if(condition)
    statement1;
else
    statement2;
```

مثال (4-7)

البرنامج الجزئي التالي يوضح كيفية تنفيذ تعليمة واحدة في تحقق الشرط

```
if ( x == 100 )
    cout<<"x is 100";
else
    cout <<"x is not 100";
```

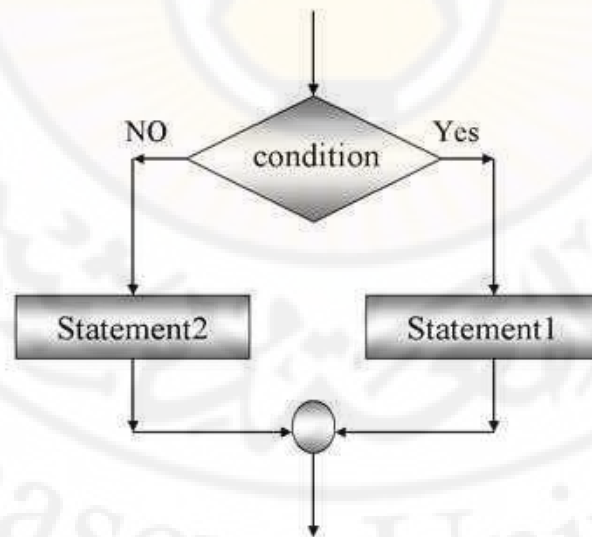
وعند تنفيذ أكثر من تعليمة في حال تحقق الشرط أو عدم تحقق الشرط ، فإن الصيغة العامة لهذه البنية كما يلي :


```

if(condition)
{
    statement 1;
    statement 2;
    .....
    statement n;
}
else
{
    statement 1;
    statement 2;
    .....
    statement m;
}

```

ويبين الشكل (٧-٢) المخطط التدفقي للتعليمية if/else .



الشكل (٧-٢) - المخطط التدفقي للتعليمية if/else

فإذا تحقق الشرط condition ، يتم تنفيذ التعليمة 1 statement وإذا لم يتحقق الشرط فيتم تنفيذ التعليمة 2 statement .

مثال (5-7)

أكتب برنامجاً بلغة C++ يقوم بإدخال عدد صحيح m ومن ثم يطبع الجملة التالية : "This number m is positive" إذا كان العدد m أكبر أو يساوي الصفر أو يطبع الجملة : "This number m is negative" إذا كان العدد m أصغر من الصفر .

البرنامج:

```
#include<iostream.h>

void main() {
    int m;
    cout<<"Enter the number m: ";
    cin>>m;
    if (m>0)
        cout<<"The number "<<m<<" is positive"<<endl;
    else
        cout<<"The number "<<m<<" is negative"<<endl;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```
Enter the number m: -3
The number -3 is negative
Press any key to continue
```

مثال (6-7)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة جذور معادلة من الدرجة الثانية:

$$ax^2+bx+c=0$$

البرنامج:

```
//This program solves the quadratic equation.
```

```
#include <iostream.h>
```

```
#include <math.h>
```

```
void main( )
```

```
{
```

```
    int a,b,c,delta;
```

```
    float x1,x2;
```

```
    cout <<"Enter the values of a, b and c : ";
```

```
    cin >>a>>b>>c;
```

```
    delta=((b *b )-(4*a *c ));
```

```
    if (delta>=0)
```

```
    {
```

```
        x1=(-b -delta)/(2*a );
```

```
        x2=(-b +delta)/(2*a );
```

```
        cout<<"x1="<<x1<<endl;
```

```
        cout<<"x2="<<x2<<endl;
```

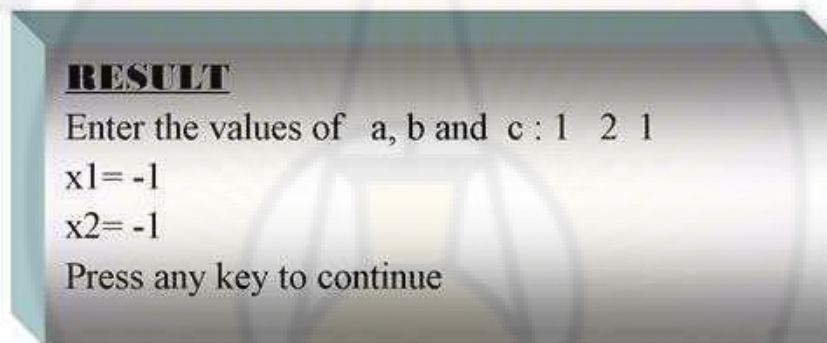
```
    }
```

```

else
{
cout<<"this equation does not have";
cout<<" any real solution";
}
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



٧-٢-٣ - التعليمة if- else-if

تستخدم التعليمة if- else-if في اختبار الشرط الأول ، فإذا تحقق هذا الشرط فان التعليمة الأولى (أو مجموعة من التعليمات) تنفذ وإذا لم يتحقق الشرط الأول فانه يتم اختبار الشرط الثاني ، فإذا تحقق الشرط الثاني يتم تنفيذ التعليمة الثانية (أو مجموعة من التعليمات) وإلا يتم تنفيذ التعليمة الثالثة (أو مجموعة من التعليمات)، والصيغة العامة لهذه البنية هو التالي:

```
if (condition1)
```

```
statement1;
```

```
else if(condition2)
```

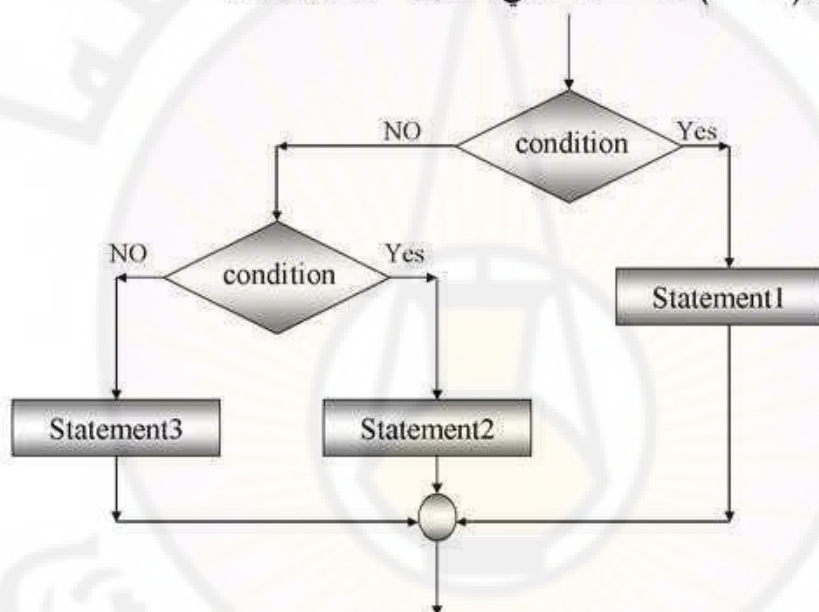


```

statement2;
else
statement3;

```

ويبين الشكل (٧ - ٣) المخطط التدفقي للتعليلة if-else-if .



الشكل (٧ - ٣) - يبين المخطط التدفقي لبنية التعليلة if-else-if

مثال (7-7)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة قيمة الدالة الرياضية المعرفة كما يلي:

$$f = \begin{cases} x^2 + 3 & \text{if } 5 \leq x \leq 10 \\ x^3 + \frac{3-x}{2} & \text{if } 10 < x \leq 100 \\ \frac{x-1}{25} & \text{other wise} \end{cases}$$

البرنامج:

```
#include<iostream.h>
void main() {
    float x,f;
    cout<<"Enter the value of x ";
    cin>>x;
    if ((x<=10)&&(x>=5))
        f=x*x+3;
    else if((x<=100)&&(x>10))
        f=x*x*x+(3-x)/2;
    else
        f=(x-1)/25;
    cout<<"The value of function f is "<<f<<endl;
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of x 7.5

The value of function f is 59.25

Press any key to continue

ملاحظة

عندما يزيد عدد التعليمات الواجب تنفيذها بعد if أو بعد else/ if أو بعد else ، عن تعليمة واحدة ، عندئذ يجب حصر هذه التعليمات بين قوسين { } .

مثال (7-8)

اكتب برنامجاً بلغة C++ يقوم بما يلي :

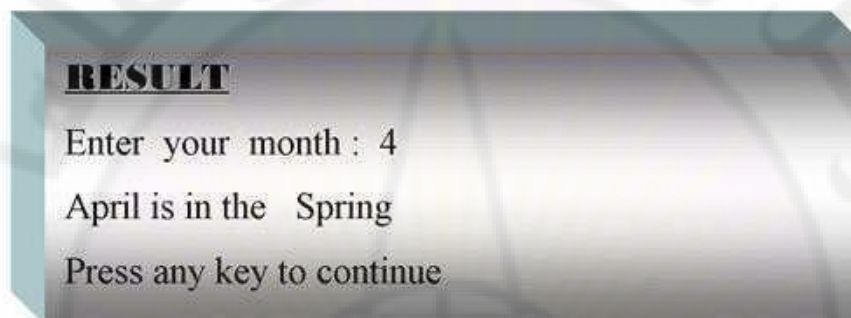
- إدخال رقم الشهر month (1,2,3,...,12) .
- طباعة الفصل الذي ينتمي إليه الشهر [Winter(12,1,2) (Spring(3,4,5), Summer(6,7,8), Autumn(9,10,11)] .

البرنامج :

```
// Demonstrate if-else-if statements.
#include<iostream.h>
void main ( ) {
    int month ;
    cout<< " Enter Your Month \n " ;
    cin>>month; // 4= April
    if(month == 12 || month == 1 || month == 2)
        cout<< "Winter \n";
    else if(month == 3 || month == 4 || month == 5)
        cout<< " Spring\n";
    else if (month == 6 || month == 7 || month == 8)
        cout<< " Summer\n";
    else if (month == 9 || month == 10 || month == 11)
```

```
cout<< " Autumn\n";
else
cout<< " Bogus Month\n";
} //End main
```

وعند تنفيذ البرنامج نحصل على الخرج التالي :



٧-٢-٤- if المركبة

يمكن استخدام تعليمات if المتداخلة ، بحيث تحتوي التعليمة if على تعليمة if واحدة أو أكثر ، الصيغة العامة كما يلي :

```
if (expression1) // if1
{
    statement1;
    Statement2;
    if (expression2) // if2
    {
        statement1;
        statement2;
    } //end if2
} //end if1
```


مثال (7-9)

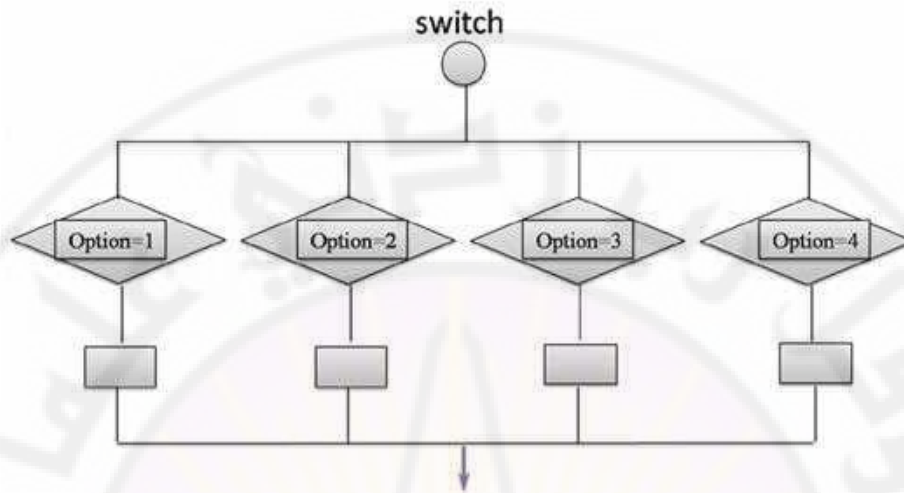
البرنامج الجزئي التالي يوضح كيفية استخدام بنية if المتداخلة .

```
if (u > v)
{
    a = 1;
    b = 2;
    if ( u > z)
    {
        x = 11;
        y = 12;
    }
}
```

٧-٢-٥ - بنية التعليم switch

تستخدم هذه البنية جملة الشرط المغلقة Close Condition أي الشرط الذي يأخذ قيمة محددة (رقمية أو حرفية) ولهذا يمكن القول بأن هذه البنية تتعامل مع متحول واحد يمكن أن يأخذ في كل مرة قيمة محددة من خلال مجموعة من القيم.

التعليم switch هي تعليمة تتحكم بتدفق البرنامج ، وتبدأ بتعبير expression وتنقل التحكم إلى إحدى عبارات case بناءً على قيمة ذلك المتحول وتعمل التعليم switch مع الأنماط الأولية: int و short و byte و char وأنماط التعداد enumeration ، وتستخدم عبارة break من أجل الخروج من العبارة switch ، وإذا لم تحتوي عبارة case على الكلمة break فإنه سيتم تنفيذ السطر الموجود بعد نهاية العبارة case ، ويستمر التنفيذ إلى أن يتم الوصول إلى عبارة break أو الوصول إلى نهاية عبارة switch ، ويُسمح بعلامة default واحدة فقط في عبارة switch . ويبين الشكل (٧-٤) المخطط التدفقي للتعليم switch .



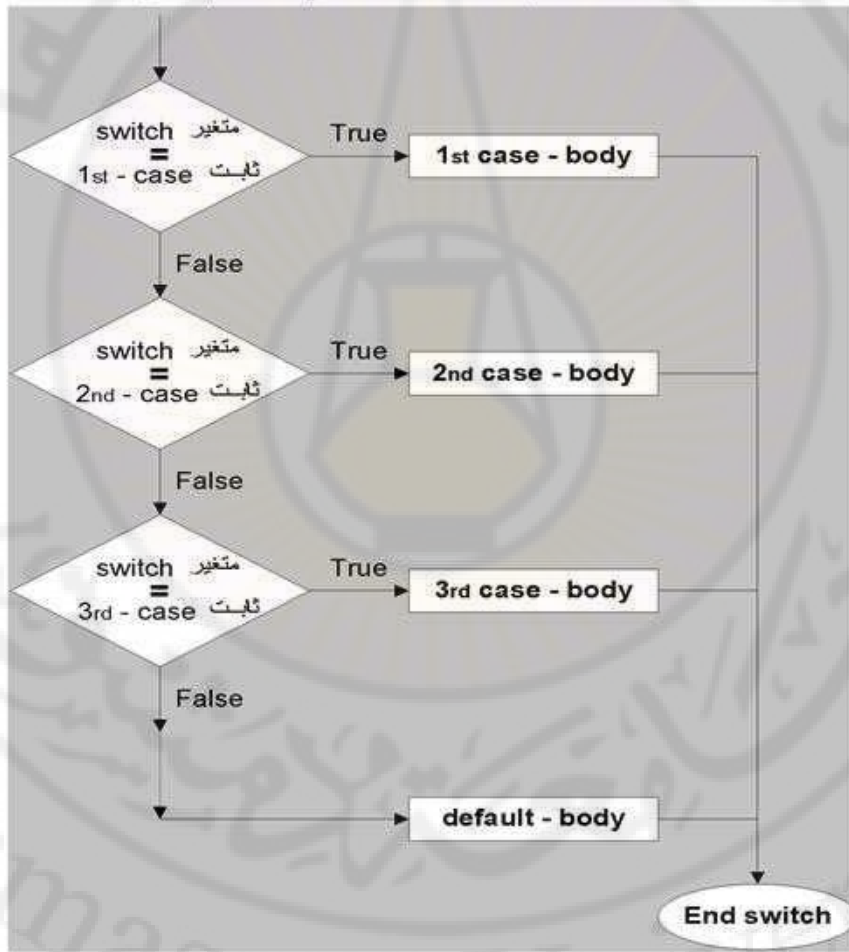
الشكل (٧-٤) - يبين المخطط التدفقي للتعليمية switch

والشكل العام لبنية التحكم **switch** هو التالي:

```

switch( expression) // int expression
{
    case value 1: statement(s); // int value
        break;
    case value 2: statement(s); // int value
        break;
    :
    case value n: statement(s); // int value
        break;
    default :statement(s);
} //end switch
  
```

هذا الشكل يمثل الاحتمالات التي يمكن أن يأخذها المتحول variable والذي يمكن أن يكون على شكل تعبير expression، كل احتمال يتم وضعه داخل جملة باستخدام العبارة case ثم بعد ذلك كتابة الحدث الذي يمكن أن يكون تعليمة أو مجموعة تعليمات ولا بد أن ينتهي الحدث بواسطة العبارة break التي تمثل نهاية الجملة، أما الاحتمال default يمثل الحدث الملازم للحالة خارج النطاق ، ويمكن رسم المخطط التدفقي لبنية التحكم switch كما في الشكل (٧-٥) .



الشكل (٧ - ٥) - يبين المخطط التدفقي لبنية التحكم switch

مثال (7-10)

اكتب برنامجاً بلغة C++ يقوم بإدخال عدد صحيح a ويطبع "zero" إذا كان $a=0$ أو يطبع "one" إذا كان $a=1$ أو يطبع "Two" إذا كان $a=2$ أو يطبع "No thing" في باقي الحالات.

البرنامج:

```
#include <iostream.h>

void main()
{
    int a;
    cout<<"Enter the value of a: ";
    cin>>a;
    switch (a)
    {
        case 0:cout<<" zero "<<endl;
        break;
        case 1:cout<<"one"<<endl;
        break;
        case 2:cout<<"Two"<<endl;
        break;
        default :cout<<" No thing"<<endl;
    }//end switch
}//end main
```


وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of a: 2

Two

Press any key to continue

مثال (11-7)

- اكتب برنامجاً بلغة C++ يقوم بما يلي (استخدم التعليمة **switch**) :
- إدخال رقم الشهر month (1,2,3,...,12).
 - طباعة الفصل الذي ينتمي إليه الشهر [Winter(12,1,2) , Spring(3,4,5), Summer(6,7,8), Autumn(9,10,11)].

البرنامج :

// An improved version of the season program.

```
#include<iostream.h>
```

```
void main ( ) {
```

```
int month ;
```

```
cout<< " Enter Your Month \n " ;
```

```
cin>>month; // 4= April
```

```
switch (month) {
```

```
case 12:
```

```
case 1:
```

```
case 2:
```

```
cout<< "Winter \n";  
break;  
case 3:  
case 4:  
case 5:  
cout<< "Spring \n";  
break;  
case 6:  
case 7:  
case 8:  
cout<< "Summer \n";  
break;  
case 9:  
case 10:  
case 11:  
cout<< "Autumn \n";  
break;  
default:  
cout<< "Bogus Month \n";  
} //End switch  
} //End main
```

وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Enter your month : 4
April is in the Spring
Press any key to continue

٧-٢ - بنية تعليمة التحكم (?:)

هذه البنية هي أداة مكافئة للتعليمة if-else ، وقد مر معنا كيفية استخدامها والشكل العام لبنية التحكم (?:) :

Variable=(condition)?result1:result2;

عندما يكون الشرط condition محققاً (true) فإن Variable =result1
وعندما يكون الشرط condition غير محققاً (false) فإن Variable =result2

مثال (7-12)

اكتب برنامجاً يقوم بما يلي :

- إدخال العددين a و b .
- إيجاد العدد الأكبر و طباعته .

البرنامج:

```
#include <iostream.h>
void main( )
{
```

```
int a,b,y;  
cout<<"Enter the value of a & b: ";  
cin>>a>>b;  
y=(a>b)?a:b;  
cout<<"y="<<y<<endl;  
} //end main
```

وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of a & b : 99 55
y=99
Press any key to continue

وعند التنفيذ الثاني للبرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of a & b : 10 88
y =88
Press any key to continue

٧-٣- بنى حلقات التكرار

Looping Structures

إن استخدام حلقات التكرار يُعد من المواضيع المهمة في البرمجة، حيث إن العديد من المسائل تحتاج إلى تكرار تعليمة أو مجموعة تعليمات أكثر من مرة، أو يتم التكرار لطالما أن شرط ما محقق، وفي لغة C++ هناك ثلاث بنى للتكرار وهي : حلقة for ، حلقة while ، وحلقة do-while .

٧-٣-١- الحلقة for

- تستخدم الحلقة for لتكرار مجموعة من التعليمات عدداً محدداً من المرات ، وتحتوي على ثلاثة أقسام هي : التهيئة والتعبير والتحديث .
- التهيئة : هو تعبير يقوم بتهيئة متحول الحلقة initialization ، لإعطاء قيمة ابتدائية لعداد (متحول) الحلقة .
 - التعبير termination : يمثل شرط التوقف ويتم اختياره في بداية كل تكرار ويحدث التكرار فقط في حال كانت قيمة التعبير هي true .
 - التحديث increment : يمثل مقدار الزيادة أو النقصان لعداد (متحول) الحلقة.

الصيغة العامة :

```
for(initial exp ; condition exp ; incre/decre exp)
```

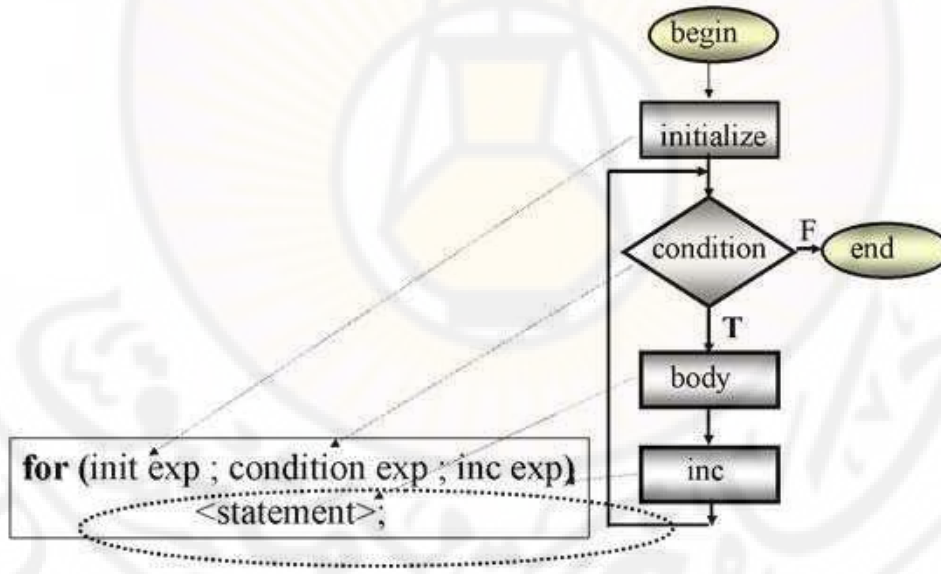
```
<statement>;
```

```
Or compound statement (block) { ... }
```

حيث:

initial- exp - القيمة الابتدائية لعداد حلقة التكرار for .

condition exp - شرط حلقة التكرار وغالباً ما يحوي قيمة نهائية.
 incre/decre exp - الزيادة المنتظمة لعدد الحلقة .
 Statement - تعليمة يتم تكرار تنفيذها لعدد من المرات .
 تقوم الحلقة for بمبتدئة بقيمة العدد الابتدائية بتنفيذ التعليمة Statement في المرة الأولى، وفي المرة التالية تزداد القيمة الابتدائية للعدد بمقدار الزيادة ثم تنفذ التعليمة Statement مرة ثانية، وهكذا حتى يصبح الشرط condition غير محققاً لإنهاء عملية التكرار والخروج من الحلقة for ، ويبين الشكل (٧ - ٦) الحلقة for وما يقابلها من المخطط .



الشكل (٧ - ٦) - يبين الحلقة for وما يقابلها من المخطط

مثال (7-13)

اكتب برنامجاً بلغة C++ يقوم بطباعة الأعداد الصحيحة من 1 حتى 10.
البرنامج:

```
#include<iostream.h>
void main() {
    for (int i=1; i<=10; i++)
        cout<<i<<" ";
    cout<<endl;
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

1 2 3 4 5 6 7 8 9 10
Press any key to continue

ضمن سياق حلقة التكرار for يمكن استخدام في حلقة for النقصان المنتظم في العداد وذلك بدلاً عن الزيادة المنتظمة في العداد والمثال (7-14) يوضح ذلك.

مثال (7-14)

اكتب برنامجاً بلغة C++ يقوم بطباعة الأعداد الصحيحة من 10 حتى 1.
البرنامج:

```
#include<iostream.h>
void main() {
```

```
for (int i=1; i<=10; i-- )
    cout<<i<<" ";
cout<<endl;
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

10 9 8 7 6 5 4 3 2 1

Press any key to continue

مثال (7-15)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة مجموع الأعداد الصحيحة الفردية من 1 حتى 20.

البرنامج:

```
#include<iostream.h>
void main( )
{
    int sum=0;
    for (int i=1; i<=20; i+=2)
        sum=sum + i;    // sum+= i ;
    cout<<"sum="<<sum<<endl;
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT  
sum=100  
Press any key to continue
```

عندما يحتوي جسم الحلقة على أكثر من تعليمة يجب وضع جسم الحلقة ضمن قوسين { } كما يلي :

```
for (initial exp ; condition exp ; incre exp )  
{  
    Statement 1;  
    .....  
    Statement n;  
}
```

مثال (7-16)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة المتوسط الحسابي لعشرة أعداد حقيقية يتم إدخالها من لوحة المفاتيح ، ثم ارسم المخطط التدفقي .
البرنامج:

```
#include<iostream.h>  
void main( )  
{  
    float a, sum=0.0, average ;
```

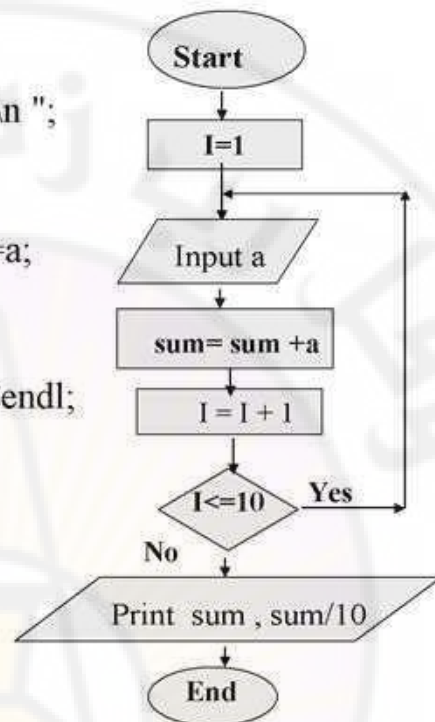
```

for (int i=1; i<=10; i++)
{
    cout<<"Enter your numbers:\n ";
    cin>>a;

    sum = sum + a ;    // sum+=a;

    average =sum/10;
    cout<<"average="<<average<<endl;
} // end main

```



وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter your numbers :

2.4 5.6 6.7 5.7 7.8 5.5 6.7 4.6 6.0 7.0

average=5.8

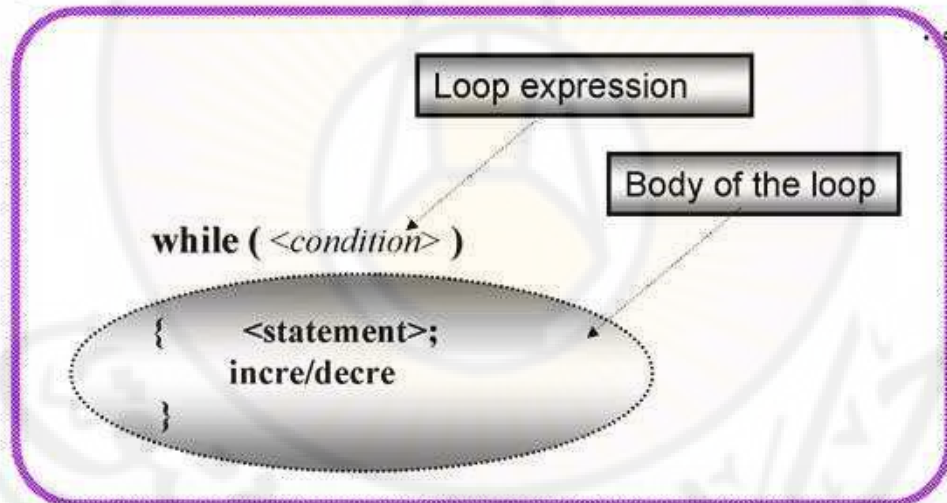
Press any key to continue

٧-٣-٢ - الحلقة while

تستخدم الحلقة while لتكرار تنفيذ تعليمة أو مجموعة تعليمات عدداً من المرات ويتوقف هذا العدد على شرط موجود في جسم الحلقة ، والشكل العام لبنية الحلقة while هو التالي:

```
while (condition)
```

```
{  
    Statement;  
    increment / decrement;  
}
```



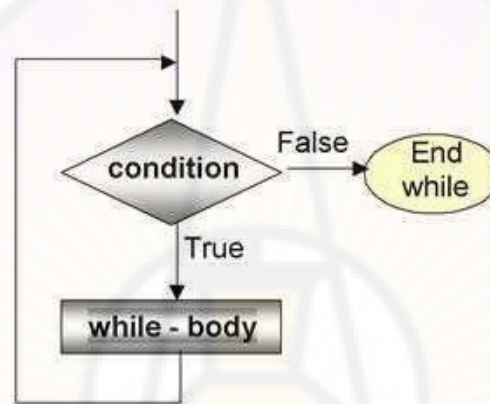
حيث:

condition - شرط حلقة التكرار.

Increment / decrement - الزيادة المنتظمة في العداد أو النقصان.

Statements - مجموعة من التعليمات يتم تكرار تنفيذها عدداً من المرات .

يتم في الحلقة while تكرار تنفيذ التعليمات statements ما دام الشرط condition محققاً أي true ، وعندما يصبح الشرط condition غير محقق false يتوقف تنفيذ التعليمات في الحلقة، وينتقل تنفيذ تسلسل تنفيذ البرنامج إلى التعليمة التي تلي الحلقة while ، ويبين الشكل (٧-٧) المخطط التدفقي لحلقة التكرار while .



الشكل (٧-٧) - يبين المخطط التدفقي لحلقة التكرار while

مثال (٧-١٧)

اكتب برنامجاً بلغة C++ يقوم بطباعة الأعداد الصحيحة من 1 حتى 10 باستخدام بنية التكرار while .

البرنامج:

```

#include<iostream.h>
void main() {
    int i=1;

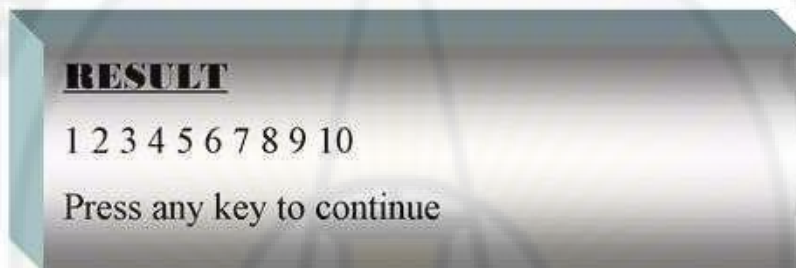
```

```

while (i<=10)
{
    cout<<i<<" ";
    i++;
}
cout<<endl;
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



مثال (7-18)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة المتوسط الحسابي لعشرة أعداد حقيقية يتم إدخالها من لوحة المفاتيح ، ثم ارسم المخطط التدفقي ، استخدم الحلقة while في البرنامج .

البرنامج:

```

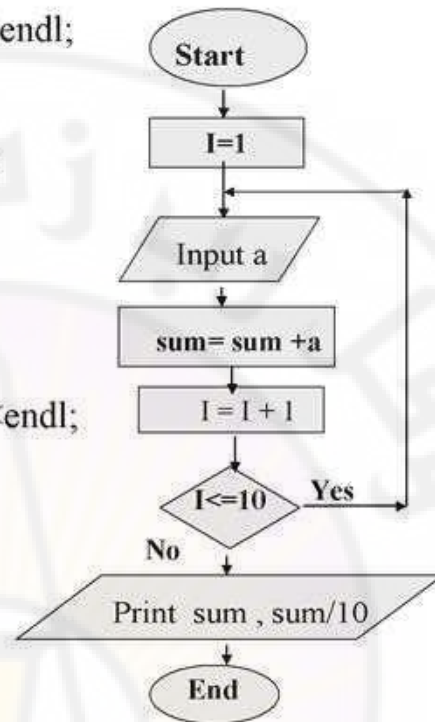
#include<iostream.h>
void main( )
{
    float a, sum=0.0, average ;
    int i=1;
    while( i<=10) {

```

```

cout<<"Enter the number : "<<endl;
cin>>a;
    sum = sum + a ; // sum+=a;
i++;
} //end while
average =sum/10;
cout<<"average="<<average<<endl;
} // end main

```



وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the number:

2.4 5.6 6.7 5.7 7.8 5.5 6.7 4.6 6.0 7.0

average = 5.8

Press any key to continue

الحلقة for هي حلقة تكرار عددية تعتمد على العداد وينتهي التكرار بانتهاء عدد مرات التكرار، أما بنية الحلقة while هي حلقة شرطية أي تعتمد على الشرط الذي يلي الأمر while حيث تتكرر التعليمات التي تليه طالما كان الشرط

صحيحاً ، وتنتهي الحلقة عندما يكون الشرط غير محققاً ، وبالتالي الاستخدام الأمثل للحلقة for هو تكرار تعليمة أو مجموعة تعليمات أكثر من مرة بشرط أن يكون عدد مرات التكرار محدداً، أما الاستخدام الأمثل للحلقة while هو التكرار بناء على شرط معين.

مثال (7-19)

البرنامج التالي يستخدم حلقة التكرار while والتي من خلالها يُطلب من المستخدم الاستمرار بإدخال محرف ما ويتم التوقف عن التكرار عندما يقوم المستخدم بإدخال المحرف Y أو المحرف N .

البرنامج:

```
//This program designs a dialogue between
// programmer and computer using while - loop.
#include <iostream.h>
void main() {
    char answer;
    cout<<"Do you want to continue (Y/N)?";
    cin>>answer;
    while ((answer!='Y')&&(answer!='N'))
    {
        cout<<"\n You must type a Y or an N \n";
        cout<<"Do you want to continue (Y/N)?";
        cin>>answer;
    }//end while
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Do you want to continue (Y/N)?a

You must type a Y or an N

Do you want to continue (Y/N)?y

You must type a Y or an N

Do you want to continue (Y/N)?n

You must type a Y or an N

Do you want to continue (Y/N)?b

You must type a Y or an N

Do you want to continue (Y/N)?Y

Press any key to continue

do - while ٣-٧ - ٣-٣ - بنية الحلقة

تستخدم الحلقة do-while لتكرار تنفيذ تعليمة أو مجموعة من التعليمات أكثر من مرة بناءً على شرط معين كما هو الحال مع الحلقة while ولكن الفرق بينهما هو أن الحلقة while تختبر الشرط أولاً ، فإذا كان الشرط صحيحاً true عندئذٍ تنفذ التعليمات الموجودة داخل الحلقة ، وإلا ويتم الانتقال إلى تنفيذ التعليمات التي تلي الحلقة، أما الحلقة do-while فتتخذ التعليمات الموجودة ضمن الحلقة ثم تختبر الشرط، فإذا كان الشرط صحيحاً true يتم تكرار تنفيذ الحلقة وإلا تُوقف التكرار أي يتم تنفيذ الحلقة مرة واحدة على الأقل ، والصيغة العامة لبنية الحلقة do-while كما يلي:

```
do
{
Statement I;
:
:
}
```

```

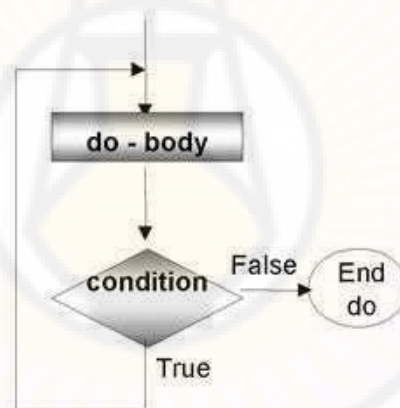
Statement n;
    increment or decrement;
}
while (condition);

```

حيث:

condition - شرط تكرار الحلقة وغالباً ما يحوي قيمة نهائية.
 decrement / increment - الزيادة المنتظمة في العداد أو النقصان.
 Statements - مجموعة التعليمات التي يتم تكرار تنفيذها عدداً محدداً من المرات.

يبين الشكل (٧ - ٨) المخطط التدفقي لبنية الحلقة do-while .



الشكل (٧ - ٨) - يبين المخطط التدفقي لبنية الحلقة do-while

مثال (7-20)

اكتب برنامجاً بلغة C++ يقوم بطباعة الأعداد الصحيحة من 10 حتى 1 وذلك باستخدام الحلقة do-while .

البرنامج:

```
#include<iostream.h>
void main( )
{
    int i=10;
    do
    {
        cout<<i<<" ";
        i--;
    }
    while (i>0);
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



RESULT

10 9 8 7 6 5 4 3 2 1

Press any key to continue

مثال (7-21)

اكتب برنامجاً بلغة C++ يقوم بحساب مجموع الأعداد الصحيحة من 100 حتى 200 وذلك باستخدام بنية التكرار do-while .

البرنامج:

```

/* This program uses the do-while loop to calculate
   the sum of 100 numbers from 100 to 200. */
#include<iostream.h>
void main( )
{
    int total=0, num=100;
    do{
        total+=num;
        num++;
    }
    while (num<=200);
    cout<<"The total is "<<total<<endl;
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

The total is 15150

Press any key to continue

مثال (7-22)

البرنامج التالي يستخدم الحلقة do-while لطباعة رسالة يطلب فيها من المستخدم إدخال كلمة سر ومقارنتها بالكلمة المخزنة ، فإذا كانت الكلمة المدخلة صحيحة انتهى البرنامج وإلا يتم تكرار استقبال كلمة السر مرة أخرى.

البرنامج:

```

#include<iostream.h>
#include<string.h>
void main( )
{
    char password[10];
    do {
        cout<<"Enter your password:";
        cin>>password;
    }
    while (strcmp(password,"Akram")!=0);

} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```

Enter your password: Omar
Enter your password: Kamel
Enter your password: akram
Enter your password: Akram
Press any key to continue

```

التابع (strcmp(s1,s2) موجود في المكتبة string.h ، ويقوم بمقارنة السلسلة المحرفية s1 مع s2 ، إذا كانت نتيجة المقارنة False يطلب إدخال كلمة سر ثانية ، وإذا كانت النتيجة True يتم الخروج من الحلقة (سوف يتم شرح التتابع المحرفية في فصل السلاسل المحرفية) .

مثال (7-23)

لدى شركة تجارية 30 مندوب مبيعات يتقاضى كل مندوب راتباً شهرياً أساسياً مقداره (10000) ليرة ، ويتقاضى عمولة قدرها 4% إذا كانت المبيعات الشهرية لا تتجاوز (200000) ليرة ، ويتقاضى عمولة قدرها 6% إذا كانت المبيعات الشهرية لا تتجاوز (400000) ليرة، ويتقاضى عمولة قدرها 7% إذا كانت المبيعات الشهرية أكثر من ذلك . والمطلوب :

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال الراتب الشهري وقيمة المبيعات الشهرية وطباعتها لجمين المندوبين
salary ,W ,Allsalary, amont

- حساب وطباعة الدخل الشهري الكامل للمندوبين .

ملاحظة : افترض salary الراتب الشهري الأساسي ، amont المبيعات الشهرية

، W العمولة ، Allsalary الراتب الشهري الكامل .

البرنامج :

```
#include<iostream.h>

void main( )
{
    double salary ,W ,Allsalary, amont;
    cout<<" salary & amont"<< endl;
    for ( int i=1 ; i<= 30 ; i++ )
    {
        cin>>salary>>amont ;
        cout<<"-----\n";
        if( amont <= 200000 )
```

```

W=amont*4/100;
else if( amont <= 400000 )
    W=amont*6/100;
else
    W=amont*7/100;
cout<<" W="<<W<<endl;
cout<<"-----\n";
Allsalary = salary + W ;
cout<<" Allsalary = "<<Allsalary<<endl;
cout<<"-----\n";
} //End for
} //End Program

```

٧-٤ - تعليمات التفريع

Branching Statement

٧-٤-١ - تعليمة التوقف break

تُستخدم تعليمة التوقف break من أجل تغيير مجرى تدفق التحكم ضمن البرنامج وتسبب التعليمة break عند تنفيذها مع الحلقات :
for, while , do- while
الخروج من هذه الحلقات ويتابع البرنامج بعدها التنفيذ مع أول تعليمة تلي الحلقة.

مثال (7-24)

البرنامج التالي يوضح لنا كيفية استخدام التعليمة break مع بنية التكرار .for

البرنامج:

```
//Using the break statemen in a for structure
#include<iostream.h>
void main( )
{
    int x;
    for(x=1;x<=10;x++)
    {
        if(x==5)
            break;
        cout<<x<<" ";
    }
    cout<<"\n Broke out of loop when x became "<<x<<endl
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

1 2 3 4

Broke out of loop when x became 5

Press any key to continue

٧-٤-٢- تعليمية الاستمرار continue

تُستخدم تعليمية الاستمرار continue من أجل تغيير مجرى تدفق التحكم ضمن البرنامج ، وتسبب التعليمية continue عند تنفيذها مع إحدى الحلقات :
For, while , do- while

تتجاوز تعليمة الاستمرار `continue` ما تبقى من التعليمات في جسم بنية التكرار والمتابعة مع المرور التالي في الحلقة .

مثال (7-25)

البرنامج التالي يوضح لنا كيفية استخدام التعليمة `continue` مع الحلقة `for` .
البرنامج:

```
//Using the continue statemen in a for structure
#include<iostream.h>
void main( )
{
    int x;
    for(x=1;x<=10;x++)
    {
        if(x==5)
            continue;
        cout<<x<<" ";
    }//end for
    cout<<"\n Used continue to skip printing the value 5 "<<endl;
} //end main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

1 2 3 4 6 7 8 9 10

Used continue to skip printing the value 5

Press any key to continue

٧-٤-٣- تعليمية return

تُستخدم التعليمية return للخروج من التابع الحالي ، ويُعاد تحكم التنفيذ إلى التعليمية التي تلي تعليمية استدعاء التابع مباشرة ، وتملك التعليمية return شكلين :

- الأول : يُعيد قيمة (كما هي في التوابع التي تعيد قيمة) .
 - الثاني : لا يعيد قيمة .
- لإعادة قيمة يكفي أن نضع القيمة المراد إعادتها بعد العبارة return مباشرة .
- مثال :

```
return ++count ;
```

يجب على نمط القيمة المعادة أن يطابق نمط التابع ، وعند التصريح عن التابع على أنه void ، لا داعي لإعادة أي قيمة ويكفي استخدام العبارة return فقط للخروج من التابع كما في الشكل التالي :

```
return ;
```

٧-٥- تعليمية الانتقال goto

تُستخدم هذه التعليمية من أجل تحويل سير تنفيذ البرنامج والانتقال إلى العنوان (Label) لتنفيذ التعليمات التي تلي Label إما بتحقيق شرط ما أو بدونه ، ونادراً ما تُستخدم هذه التعليمية في البرمجة .

مثال (7-26)

البرنامج التالي يوضح لنا كيفية استخدام تعليمية الانتقال المشروط goto .

البرنامج:

```
// goto example  
#include<iostream.h>
```



```

void main( )
{
    int n=10 ;
Label :
    cout<<n<<" ,";
    n-- ;
    if(n>0) goto Label ;
    cout<<endl;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

10, 9, 8, 7, 6, 5, 4, 3, 2, 1 ,

Press any key to continue

مثال (7-27)

البرنامج التالي يوضح لنا كيفية استخدام تعليمة الانتقال غير المشروط goto .

البرنامج:

```

#include<iostream.h>
void main( )
{
    int a,b,c;
    a=1; b=2;
    first :
    c=a*b ;
    cout<<"a="<<a<<endl;
}

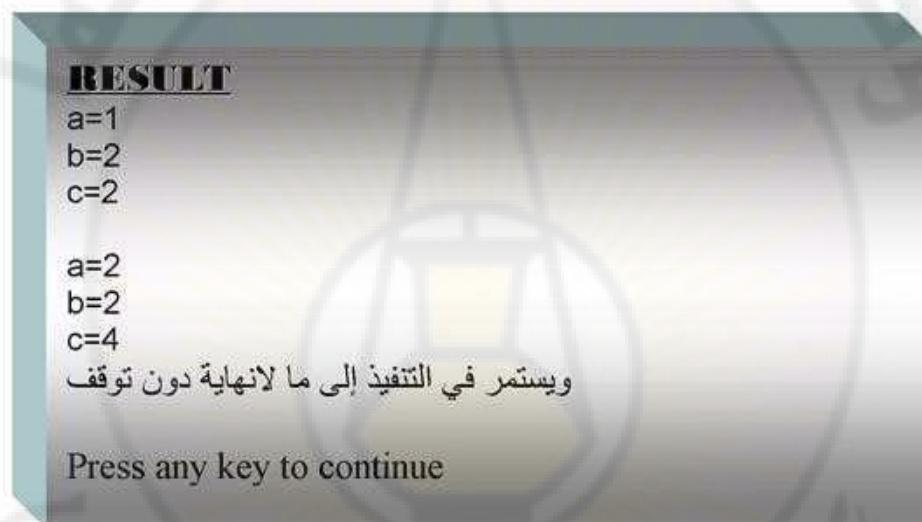
```

```

cout<<"b="<<b<<endl;
cout<<"c="<<c<<endl<<endl;
a++ ;
goto first ;
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



٧-٦ - تعليمة الخروج (exit())

نستخدم هذه التعليمة من أجل الخروج بشكل نهائي من البرنامج عند تحقق شرط ما.

مثال (7-28)

البرنامج التالي يوضح لنا كيفية استخدام تعليمة الخروج (exit()).
البرنامج:

```

#include<iostream.h>
#include<stdlib.h>
void main( )
{
    int i , n , sum=0 ;
    cout<<"Please enter 10 positive values :\n";
    for (i=1 ; i<=10 ; i++)
    {
        cin>>n ;
        if (n <=0)
        {
            cout<<"This is negative or zero number \n";
            exit(0);
        }
        sum+=n ;
    }
    cout<<"The sum of positive " ;
    cout<<"values are : "<<sum<<endl;
} //end main

```

وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Please enter 10 positive values :

5 7 3 8 9 4 9 6 3 7

The sum of positive values are :61

Press any key to continue

وعند التنفيذ الثاني للبرنامج سنحصل على الخرج التالي:

RESULT

Please enter 10 positive values :

6 7 -9

This is negative or zero number

Press any key to continue

=====

=====

=====



مسائل عامة محلولة

١- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- إدخال درجة الطالب grade .
- يختبر درجة الطالب هل هي A أو B أو C أو D أو F .
- حيث :

$C \in [70 - 80]$, $B \in [80 - 90]$, $A \in [90 - 100]$

$F < [60]$, $D \in [60 - 70]$

البرنامج :

```
//calculate and print the mark of a student
```

```
#include<iostream.h>
```

```
void main( )
```

```
{
```

```
float grade;
```

```
cout<< " Enter Your The mark \n " ;
```

```
cin>> grade; // 55
```

```
if (grade >= 90)
```

```
cout<<"The mark is A\n";
```

```
else if (grade >= 80
```

```
cout<<" The mark is B\n" ;
```

```
else if (grade >= 70 )
```

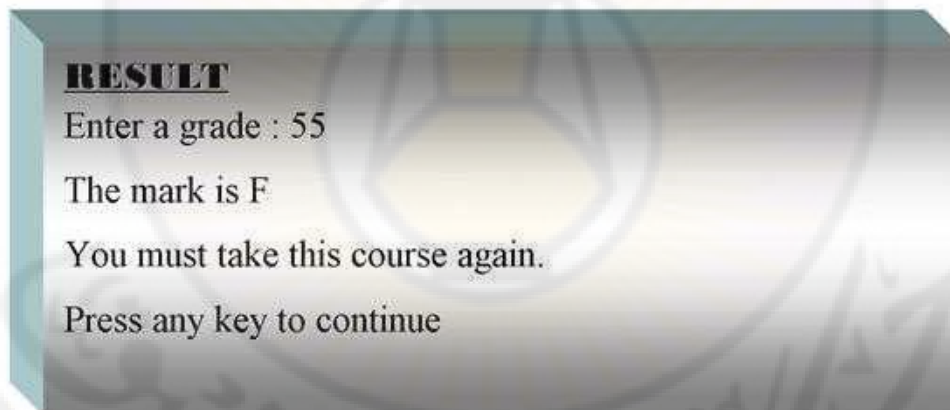


```

    cout<<" The mark is C\n" ;
else if (grade >= 60 )
    cout<<" The mark is D\n" ;
else if (grade < 60) {
    cout<<" The mark is F\n" ;
    cout<<" You must take this course again.\n" ;
}
} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



- ٢- اكتب برنامجاً بلغة C++ يقوم بما يلي :
 - إدخال عددين صحيحين a و b .
 - إيجاد وطباعة المضاعف المشترك الأصغر للعددين المدخلين.
- البرنامج:

//This program computes the lowest common multiple.

#include<iostream.h>

void main()

{

int a,b,lcm;

 cout<<"Enter two numbers ";

 cin>>a>>b;

if (a>b)

 {

if (a%b==0)

 lcm=a;

else

 lcm=a*b;

 }

else

 {

if(b%a==0)

 lcm=b;

else

 lcm=a*b;

 }

 cout<<"lcm="<<lcm<<endl;

}//end main

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter two numbers 5 8

lcm=40

Press any key to continue

٣- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- إدخال عددين صحيحين a و b .
- إدخال قيمة صحيحة للمتحول m بحيث إذا كانت m=1 يقوم البرنامج بإيجاد العدد الأكبر من بين العددين المُدخلين ، وإذا كانت m=2 يقوم البرنامج بحساب المتوسط الحسابي للعددين المُدخلين ، وإذا كانت m=3 يقوم البرنامج بإيجاد المضاعف المشترك الأصغر للعددين المُدخلين .
- استخدم الحلقة switch .

البرنامج:

```
#include<iostream.h>
void main( )
{
    int a,b,max,m,lcm;
    float sum=0.0,avg;
    cout<<"Enter two numbers a, b:";
    cin>>a>>b;
    cout<<"Enter the value m:";
    cin>>m;
```

```

switch (m)
{
case (1):
{
if(a>b)
max=a;
else
max=b;
cout<<"max="<<max<<endl;
break;
}
case (2):
{
sum=a+b;
avg=sum/2;
cout<<"avg="<<avg<<endl;
break;
}
case (3):
{
if (a>b)
{
if (a%b==0)
lcm=a;
else
lcm=a*b;
}
else
{

```

```

        if(b%a==0)
            lcm=b;
        else
            lcm=a*b;
    }
    cout<<"lcm="<<lcm<<endl;
    break;
}
default:
{
    cout<<"Please enter the number m from 1 to 3"<< endl;
}
}end switch
}end main

```

وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Enter two numbers a, b:5 9

Enter the value m:3

lcm=45

Enter the value m:4

Please enter the number m from 1 to 3

Press any key to continue

٤- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- إدخال عددين صحيحين a و b .
- إدخال قيمة محرفية (* أو / أو + أو - أو أي قيمة أخرى) .
- يستخدم التعليمة switch من أجل إجراء إحدى العمليات الحسابية الأساسية (الضرب أو القسمة أو الجمع أو الطرح) على العددين المُدخلين a و b .

البرنامج:

```
#include <iostream.h>

void main( )
{
    int a,b;
    char op;
    float result;
    cout<<"Please enter a simple expression";
    cin>>a>>op>>b;
    switch(op)
    {
        case '+':
            result=a+b;
            cout<<a<<" "<<op<<" "<<b
                <<" equals "<<result<<endl;
            break;
```

```

case '-' :
    result=a-b;
    cout<<a <<" "<<op<<" "<<b
        <<" equals "<<result<<endl;
break;
case '*' :
    result=a*b;
    cout<<a <<" "<<op<<" "<<b
        <<" equals "<<result<<endl;
break;
case '/' :
    if(b!=0)
    {
        result=(a/b);
        cout<<a <<" "<<op<<" "
            <<b <<" equals "<<result<<endl;
    }
    else
    {
        cout<<a<<"/"<<b<<"cannot be computed:";
        cout<<"denominator is 0."<<endl;
    }
break;

```

default:

```
cout<<op<<"is";  
cout<< " unrecognized operation"<<endl;  
} //end switch  
} //end main
```

وعند التنفيذ الأول للبرنامج سنحصل على الخرج التالي:

RESULT

Please enter a simple expression : 6 * 9
6 * 9 equals 54
Press any key to continue

وعند التنفيذ الثاني للبرنامج سنحصل على الخرج التالي:

RESULT

Please enter a simple expression : 9 / 0
9/0 cannot be computed: denominator is 0.
Press any key to continue

٥- اكتب برنامجاً بلغة ++C يقوم بإيجاد وطباعة درجة الحرارة العظمى خلال شهر تشرين الأول (31 يوم)، باستخدام الحلقة while .
البرنامج:

```

#include<iostream.h>
void main() {
    int i=1,t,max=-100,imax;
    while(i <= 31)
    {
        cout<<"Enter the temperature of day :";
        cin>>t;
        if(t>max)
        {
            max=t;
            imax=i;
        }
        i++;
    }//end while
    cout<<"max="<<max<<endl;
    cout<<"imax="<<imax<<endl;
} //end main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the temperature of day :

21 24 15 16 18 20 14 12 23 22 20 24 21 19 14
18 17 15 14 12 13 18 21 22 23 18 17 15 14 14 13

max=24

imax=2

Press any key to continue

مسائل عامة غير محلولة

١- لدى شركة تجارية مندوب مبيعات يتقاضى راتباً شهرياً أساسياً مقداره (10000) ليرة ، ويتقاضى عمولة قدرها 4% إذا كانت المبيعات الشهرية لا تتجاوز (200000) ليرة ، ويتقاضى عمولة قدرها 6% إذا كانت المبيعات الشهرية لا تتجاوز (400000) ليرة، ويتقاضى عمولة قدرها 7% إذا كانت المبيعات الشهرية أكثر من ذلك . والمطلوب :

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال الراتب الشهري وقيمة المبيعات الشهرية وطباعتها
- حساب وطباعة الدخل الشهري الكامل للموظف .
- ارسم المخطط التدفقي المناسب .

ملاحظة : افترض salary الراتب الشهري الأساسي ، amount المبيعات الشهرية ، W العمولة ، Allsalary الراتب الشهري الكامل .

٢- اكتب برنامجاً بلغة C++ يقوم بحساب مجموع الأعداد الصحيحة الفردية من 1 حتى 10، وطباعتها ، ثم ارسم المخطط التدفقي .

٣- اكتب برنامجاً بلغة C++ يقوم بحساب مجموع الأعداد الصحيحة الزوجية من 1 حتى 10، وطباعتها ، ثم ارسم المخطط التدفقي .

٤- اكتب برنامجاً بلغة C++ يقوم بحساب مجموع مضاعفات العدد 5 للأعداد الصحيحة من 1 حتى 100، وطباعتها ، ثم ارسم المخطط التدفقي .

٥- يعمل 100 عامل في معمل نسيج بشكل ساعي ، لنفرض أن H هي عدد ساعات العمل وأن R هي أجرة الساعة الواحدة و W هو الأجر اليومي .
والمطلوب :

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- قراءة وطباعة عدد ساعات العمل وأجرة الساعة لكل عامل .
- حساب وطباعة الأجر اليومي لكل عامل .
- ارسم المخطط التدفقي المناسب

٦- لدينا 300 طالباً يدرسون مقرر البرمجة والمطلوب :

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال درجة الطالب d وطباعة كلمة ناجح " Passed " إذا كانت درجة الطالب $d \geq 60$ ويطبع كلمة راسب " Failed " إذا كانت غير ذلك .
- ارسم المخطط التدفقي اللازم .
- استخدم الحلقة while

٧- اكتب برنامجاً بلغة C++ يقوم بإدخال 30 عدداً حقيقياً وطباعة كلمة " Negative " إذا كان العدد أصغر من الصفر ، ويطبع كلمة موجب إذا كان العدد أكبر أو يساوي الصفر .

- ارسم المخطط التدفقي المناسب .
- استخدم الحلقة do - while .

=====

=====

=====

الفصل الثامن

المصفوفات

Arrays



المصفوفات

٨-١- مقدمة

- تتقسم المعطيات إلى معطيات محرفية (char) و صحيحة (int) و حقيقية (float) ، وتسمى هذه الأنواع بالأنواع الرئيسية للمعطيات، ولكن هناك أنواع أخرى من المعطيات تسمى بالأنواع المشتقة Derived Data Types ، منها المصفوفات Arrays ، ومن أهم فوائد استخدام المصفوفات ، هي :
- يمكن استخدام المصفوفات من أجل الترتيب التصاعدي والتنازلي للعناصر والقيم المختلفة (القيم العددية).
 - يمكن استخدام المصفوفات في ترتيب الأسماء وفق الترتيب الأبجدي في النصوص الرمزية.
 - يمكن استخدام المصفوفات في عمليات ضرب المصفوفات.
 - يمكن استخدام المصفوفات في إيجاد مقلوب مصفوفة.

٨-٢- تعريف المصفوفة

Array Definition

المصفوفة هي بنية تُستخدم للاحتفاظ بمجموعة من القيم من نفس نمط المعطيات والمصفوفة هي نوع من أنواع بنى المعطيات، لها عدد محدود ومرتب من العناصر التي تكون جميعها من نفس النوع type، فمثلاً يمكن أن تكون جميعها صحيحة int أو حقيقية float أو محرفية char ، ولكن لا يمكن الجمع بين نوعين مختلفين في المصفوفة نفسها، وتنقسم المصفوفات إلى نوعين : مصفوفات أحادية البعد ومصفوفات متعددة الأبعاد (على سبيل المثال مصفوفات ثنائية الأبعاد).

٨ - ٣ - المصفوفات أحادية البعد

One-Dimensional Arrays

المصفوفة التي تتكون من سطر واحد أو عمود واحد ، تسمى مصفوفة أحادية البعد، ويمكن أن تكون المصفوفة أحادية البعد حرفية char أو حقيقية float أو صحيحة int ، أما دليل المصفوفة (index) ، فيجب أن يكون من النوع الصحيح int ، وكل مصفوفة تُستخدم في البرنامج ، لا بد من التصريح عنها، قبل استخدامها .

٨-٣-١- التصريح عن المصفوفة أحادية البعد

تحتل المصفوفات حجماً في الذاكرة لذا يجب على المبرمج تحديد نوع عناصر المصفوفة وعددها حتى يتسنى للمعرف تخصيص الحجم اللازم من الذاكرة لحفظ المصفوفة، وحتى تخبر المترجم بأن يخصص حجماً محدداً لنوع عناصر المصفوفة، على سبيل المثال، حجماً لـ 10 عناصر من النوع int .
والشكل العام للتصريح عن المصفوفة أحادية البعد هو التالي:

data type Array_ Name[size];

حيث:

data type - نوع المعطيات .

Array_ Name - اسم المصفوفة، والذي يتم وضعه من قبل المبرمج مع مراعاة قواعد تسمية المتحولات، حيث إن كل متحول في المصفوفة يدعى عنصراً وكل عنصر في المصفوفة يجب أن يكون من نوع المصفوفة.

size - حجم المصفوفة (عدد عناصر المصفوفة أو بعد المصفوفة).

إن المصفوفة **int a[n]** تحتوى على n عنصر من النوع int، ويمكن الوصول إلي أي عنصر من عناصرها بذكر اسم المصفوفة متبوعاً برقم موقع

العنصر في المصفوفة محاطاً بالأقواس [] ، حيث إن عناصر المصفوفة تكون مرقمة من (0) إلى $n-1$ ، ولهذا يشار إلى العنصر الأول في المصفوفة a بـ $a[0]$ والثاني $a[1]$ والسابع $a[6]$ وعموماً يحمل العنصر الأخير في المصفوفة a الدليل $a[n-1]$.

مثال (1-8)

للتصريح عن مصفوفة أحادية البعد من النوع الصحيح و تحتوي على عشر عناصر ، فإننا نستخدم التصريح التالي:

`int a [10];`

ومن خلال هذا التصريح ، نلاحظ أن عناصر المصفوفة a ، هي:

$a[0], a[1], a[2], a[3], a[4], a[5], a[6], a[7], a[8], a[9]$

$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$	$a[7]$	$a[8]$	$a[9]$
-45	6	0	72	0	-3	1	100	4	70

قيم عناصر المصفوفة $a[10]$

أيضاً يمكن الإعلان عن مصفوفات من أي نوع معطيات آخر ، فمثلاً للتصريح عن مصفوفة عناصرها من النوع `char` يمكن كما يلي :

`char a[ 10 ];`

يمكن أن نعرف عدد عناصر المصفوفة على أنه قيمة ثابتة ، ويتم ذلك باستخدام المحدد `#define` في بداية البرنامج ، ومن ثم يمكن استخدام هذا الثابت في أي مكان ضمن البرنامج ، أو استخدام `const` ولكن ضمن التابع `main()` ، وبالعودة إلى المثال السابق ، فيمكن أن نصرح ، عن عدد عناصر المصفوفة `a` على أنه من النوع الثابت كما يلي:

```
#define size 10
```

بعد ذلك، يمكن أن نصرح عن المصفوفة بالشكل التالي:

```
int a[size];
```

```
const size =10 ;
```

أو

٨-٣-٢ - إعطاء قيم ابتدائية لعناصر المصفوفة ثنائية البعد

يمكن إعطاء قيم ابتدائية لعناصر المصفوفة أحادية البعد بإتباع التصريح عن المصفوفة بعلامة المساواة (=) تليها لائحة من القيم المطلوب إسنادها لعناصر المصفوفة عندها، ويتم الفصل بين القيم بفواصل، وتحيط هذه اللائحة الأقواس الحاصرة { } ، على سبيل المثال لنأخذ التصريحات التالية لثلاث مصفوفات من أنواع مختلفة من المعطيات ، وفي الوقت نفسه إعطاء قيماً ابتدائية لعناصر المصفوفة (إجراء عملية تمهيد للعناصر) عند التصريح عنها ، كما يلي:

```
1:int n[10] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37};
```

```
2:char name[3] = { "Ahmad", "Omar", "Ali"};
```

```
3:char name[5] = { 'A', 'B', 'C', 'D', 'E'};
```

ومن خلال هذه التصريحات نبين ما يلي:

السطر 1 يعني التصريح عن المصفوفة `n` بعشر عناصر من النوع `int`

السطر 2 يعني التصريح عن المصفوفة `name` بثلاثة عناصر من النوع `char`

السطر 3 يعني التصريح عن المصفوفة name بخمسة عناصر من النوع char

مثال (2-8)

يقوم البرنامج التالي بإعطاء قيماً ابتدائية لعناصر المصفوفة من النوع integer عند التصريح عنها، و من ثم طباعة هذه القيم.

البرنامج:

```
//initializing an array
#include <iostream.h>
void main() {
int n[10] = {32,27,64,18,95,14,90,70,60,37} ;
for (int i=0 ; i< 10; i++) // print array
cout << i<<" " <<n[i]<<endl;
} //End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي



إذا كانت القيم الابتدائية الموجودة في اللائحة أكثر من حجم المصفوفة المحددة فإن المترجم سيولد خطأ، وإذا كانت أقل سيملاً المترجم بقية العناصر أصفاراً ، فإذا كان المطلوب إسناد عناصر مصفوفة مهما كان حجمها بأصفار ، يمكن التصريح عن المصفوفة بالشكل التالي :

```
int anyarray[10]={0};
```

سيتم إسناد القيمة 0 للعنصر الأول، أما العناصر المتبقية يسند لها القيمة صفر أيضاً كوننا لم نحدد قيمة لها.

إذا لم يتم تحديد عناصر المصفوفة من قبل المبرمج عند التصريح عنها، فإن المترجم يقوم بهذه المهمة من خلال عد القيم (العناصر)، الموجودة في الطرف الأيمن وحجز أماكن لعناصر المصفوفة وفق هذا العدد الناتج ، فعلى سبيل المثال :

```
int array[ ]={10,20,30};
```

هذا التصريح يعني: أن عدد عناصر المصفوفة array[] هو ثلاثة فقط، ويصلح هذا التصريح عندما نعطي عناصر المصفوفة قيمة ابتدائية .

٨-٤-٣- إدخال وإخراج عناصر المصفوفة أحادية البعد

بالإضافة إلى إعطاء قيمة ابتدائية لعناصر المصفوفة عند التصريح عنها، يمكن استخدام أمر الإدخال cin وذلك ضمن حلقة التكرار ، ومن أجل طباعة عناصر المصفوفة يُستخدم أمر الإخراج cout وذلك أيضاً ضمن حلقة تكرار ، عادة قبل استخدام المصفوفة يمكن تصغيرها ، وذلك بإعطاء قيم ابتدائية تساوي الصفر ، ويوضح المثال (8-3) كيفية إدخال وإخراج عناصر المصفوفة احادية البعد.

مثال (8-3)

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- تفسير عناصر المصفوفة myArray [5] .
- إدخال 5 أعداد صحيحة وتخزينها في المصفوفة.
- طباعة عناصر المصفوفة المدخلة.

البرنامج:

```
//This program uses an integer array
#include<iostream.h>

int main( ) {
    int myArray [5];
    for (int i =0;i<5;i++)
        myArray[i] = 0 ;
    for (int i =0;i<5;i++)
    {
        cout<<"The value of myArray["<<i<<"]:";
        cin>>myArray[i];
    }
    for (i=0;i<5;i++)
        cout<<i<<": "<<myArray[i]<<" ";
        cout<<endl;
    return 0;
}
```


وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

The value of myArray[0]:55
The value of myArray[1]:67
The value of myArray[2]:88
The value of myArray[3]:23
The value of myArray[4]:5
55 67 88 23 5

Press any key to continue

مثال (4-8)

اكتب برنامجاً بلغة C++ لحساب وطباعة المتوسط الحسابي لـ m عدداً حقيقياً (حيث $m \leq 100$) يتم إدخالها من لوحة المفاتيح وتخزينها في مصفوفة أحادية البعد myArray[m].
البرنامج:

```
//This program computes the average of array elements
#include<iostream.h>
#define size 100
int main( )
{
    float myArray [size],sum=0.0,avg;
    int i,m;
    cout<<"Enter the value of m ";
```

```

cin>>m;

for (i=0;i<m;i++)
{
    cout<<" myArray["<<i<<"]:";
    cin>>myArray[i];
    sum+=myArray[i];
}
avg=sum/m;
cout<<"The average is "<<avg<<endl;
for (i=0;i<m;i++) cout<<myArray[i]<<" ";
cout<<endl;
return 0;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of m 5

myArray[0]:2.5

myArray[1]:5.7

myArray[2]:7.8

myArray[3]:4.0

myArray[4]:8.9

The average is 5.78

2.5 5.7 7.8 4 8.9

Press any key to continue

نلاحظ ، أنه في هذا البرنامج استخدمنا بُعد المصفوفة كثابت، من خلال الملف الرئيسي:

```
#define size 100
```

وفي هذه الحالة يمكن استخدام قيمة الثابت size كبعد للمصفوفة في كل حلقات التكرار المستخدمة في البرنامج ، واستخدامنا متحول آخر للدلالة على بُعد المصفوفة وهو m والذي يتم إدخال قيمته من لوحة المفاتيح ، و يجب أن لا تتجاوز قيمة الثابت size المصرح عنها في بداية البرنامج عن 100.

مثال (5-8)

اكتب برنامجاً بلغة C++ يقوم بإدخال 10 أعداد صحيحة وتخزينها في مصفوفة أحادية البعد a[10] ، ومن ثم يقوم بإيجاد وطباعة القيمة العظمى (العدد الأكبر) لهذه الأعداد المدخلة.

البرنامج:

```
#include <iostream.h>

void main( ) {
    int a[10],max ;
    cout<<"Enter the elements\n";
    for(int i=0;i<10;i++)
    {
        cout<<"a["<<i<<"]:";
        cin>>a[i] ;
    }
}
```

```

max=a[0];
for(int i=0;i<10;i++)
{
    if (a[i]>max) max=a[i] ;
}
cout<<"The maximum is "<<max<<endl;
for(int i=0;i<10;i++) cout<<a[i]<<" ";
cout<<endl;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the elements

a[0]: 10

a[1]: 5

a[2]: 8

a[3]: 66

a[4]: 77

a[5]: 34

a[6]: 56

a[7]: 9

a[8]: 23

a[9]: 45

The maximum is 77

10 5 8 66 77 34 56 9 23 45

Press any key to continue

مثال (6-8)

- أكتب برنامجاً بلغة C++ يقوم بما يلي:
- إدخال 8 أعداد صحيحة وتخزينها في مصفوفة أحادية البعد a [8].
- ترتيب عناصر المصفوفة المدخلة على شكل تصاعدي.
- طباعة عناصر المصفوفة المدخلة قبل وبعد عملية الترتيب.
- البرنامج:

```
//Arrangement of array elements
#include<iostream.h>
#include<iomanip.h>
void main( ) {
    int i,j,x;
    int a[8];
    cout<<"Enter the array elements: "<<endl;
    for (i=0;i<8;i++)
    {
        cout<<"a["<<i<<"]="";
        cin>>a[i];
    }
    cout<<"The array elements before arrangement are : " <<endl;
    for (i=0;i< 8;i++)
        cout<<a[i]<<setw(3);
    cout<<endl;
```



```

for (i=0;i<7;i++)
{
    for (j=i;j<7;j++)
    {
        if(a[i]>a[j+1])
        {
            x=a[i];
            a[i]=a[j+1];
            a[j+1]=x;
        }
    }
}

cout<<"The array elements after arrangement are \n :";
for (i=0;i< 8;i++)
cout<<a[i]<<setw(3);
cout<<endl;
} //end main

```

ملاحظة:

استخدمنا في هذا البرنامج التابع (setw) من أجل ترك فراغات بين المخرجات ،
ويعد استخدام هذا التابع بديل عن استخدام علامتي التنصيص المزدوجة " " ،
وعندها يجب استخدام الملف الرأسي `#include<iomanip.h>` .

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT
Enter the array elements:
a[0]=5
a[1]=7
a[2]=9
a[3]=3
a[4]=2
a[5]=4
a[6]=3
a[7]=7
The array elements before arrangement are
5 7 9 3 2 4 3 7
The array elements after arrangement are
2 3 3 4 5 7 7 9
Press any key to continue
```

٨ - ٤ - المصفوفة ثنائية البعد

Tow-Dimensional Array

يمكن للمصفوفات في لغة ++C أن تكون متعددة الأبعاد ويمكن أن يكون كل بعد بحجم مختلف ، والاستخدام الشائع للمصفوفات متعددة الأبعاد هو تمثيل الجداول Tables، والجداول (٨-١) يحتوي على معطيات مرتبة في صورة صفوف وأعمدة، ولتمثيل الجدول نحتاج لبعدين البعد الأول يمثل الأسطر والبعد الثاني يمثل الأعمدة.

وبين الشكل التالي مصفوفة A تحتوي على ثلاثة صفوف وأربعة أعمدة، ويتم تمثيل أي عنصر في المصفوفة A على الصورة $A[i][j]$

	Column 0	Column1	Column2	Column 3
Row 0	$A[0][0]$	$A[0][1]$	$A[0][2]$	$A[0][3]$
Row 1	$A[1][0]$	$A[1][1]$	$A[1][2]$	$A[1][3]$
Row 2	$A[2][0]$	$A[2][1]$	$A[2][2]$	$A[2][3]$

الجدول (٨-١) - يحتوي على معطيات مرتبة في صورة صفوف وأعمدة

حيث:

A - اسم المصفوفة.

i - رقم السطر الذي ينتمي إليه العنصر.

j - رقم العمود الذي ينتمي إليه العنصر.

نلاحظ أن كل العناصر الموجودة في السطر الأول مثلاً يكون الدليل الأول لها هو 0 وكل العناصر الموجودة في العمود الرابع يكون الدليل الثاني لها هو 3.

٨-٤-١ - التصريح عن المصفوفة ثنائية البعد

إن المصفوفات تحتل حجماً في الذاكرة لذا يجب على المبرمج تحديد نوع عناصر المصفوفة وعددها حتى يتسنى للمعرف تخصيص الحجم اللازم من

الذاكرة لحفظ المصفوفة، والشكل العام للتصريح عن المصفوفة ثنائية البعد كما يلي:

Data Type Array_Name [size1][size2];

حيث:

Data Type - نوع المعطيات .

Array_Name - اسم المصفوفة، والذي يتم وضعه من قبل المبرمج مع مراعاة لقواعد تسمية المتحولات، وحيث أن كل متحول في المصفوفة يدعى عنصراً وكل عنصر في المصفوفة يجب أن يكون من نوع المصفوفة.

size1 - عدد الأسطر في المصفوفة.

size2 - عدد الأعمدة في المصفوفة.

يتم التصريح عن مصفوفة **a** تحتوى على **x** صف و **y** عمود كما يلي :

int a[x][y] ;

عند استخدام مصفوفة أحادية البعد لا بد من استخدام حلقة تكرار واحدة (for أو while أو do-while) ، بينما عند استخدام مصفوفة ثنائية البعد لا بد من استخدام حلقتي تكرار أو ما يسمى بحلقات التكرار المتداخلة Nested loops.

٨-٤-٢ - إعطاء قيم ابتدائية لعناصر المصفوفة ثنائية البعد

يمكن التصريح عن المصفوفة ثنائية البعد وإعطائها قيماً ابتدائية (إجراء عملية تمهيد للعناصر)، كما يلي :

int b[2][2] = { {1,2} , {3,4} } ;

حيث:

b[1][1]=4 , b[1][0]=3 , b[0][1]=2 , b[0][0]=1

أيضاً نلاحظ في المصفوفة متعددة الأبعاد إذا تم إسناد قيم أولية لا يتوافق عددها مع حجم المصفوفة فإن المترجم سيملاً بقية العناصر أصفاراً. ويوضح البرنامج في المثال (7-8) كيفية إسناد قيم أولية للمصفوفات متعددة الأبعاد عند التصريح عنها.

مثال (7-8)

```
// initializing multidimensional arrays
#include<iostream.h>

int main( )
{
    int array1[2][3] = { {1, 2, 3}, {4, 5, 6} },
        array2[2][3] = {1, 2, 3, 4, 5} ,
        array3[2][3] = { {1, 2}, {4} } ;

    cout << "values in array1 by row are : " << endl;
    for (int i=0; i<2; i++) {
        for (int j=0; j<3; j++)
            cout << array1 [ i ] [ j ] << ' ';
        cout << endl;
    }
    cout << "values in array2 by row are : " << endl;
    for (int i=0; i<2; i++) {
        for (int j=0; j<3; j++)
```



```

cout << array2 [ i ] [ j ] << " ";
cout << endl;
}
cout << "values in array3 by row are : " << endl;
for (int i=0; i<2; i++) {
    for (int j=0; j<3; j++)
        cout << array3 [ i ] [ j ] << " ";
    cout << endl;
}
} //End main

```

RESULT

values in array1 by row are :

1 2 3

4 5 6

values in array2 by row are :

1 2 3

4 5 0

values in array3 by row are :

1 2 0

4 0 0

Press any key to continue

٨-٤-٣- إدخال وإخراج عناصر المصفوفة ثنائية البعد

يمكن إدخال عناصر المصفوفة من قبل المستخدم ، وأيضاً يمكن إخراج (طباعة) عناصر المصفوفة ثنائية البعد بواسطة حلقتي (حلقتي for ، أو while ، أو do/while ، أو خليط بينهم) .
عادة قبل استخدام المصفوفة يمكن تصغيرها ، وذلك بإعطاء قيم ابتدائية تساوي الصفر ، ويوضح المثال (8 - 8) كيفية إدخال وإخراج عناصر المصفوفة ثنائية البعد .

مثال (8-8)

اكتب برنامجاً بلغة C++ يقوم بإدخال 12 عدد حقيقي وتخزينها في مصفوفة ثنائية البعد [4] [3] a ، ثم يقوم البرنامج بطباعة عناصر المصفوفة المدخلة على شكل مصفوفة ثنائية (الطباعة يجب أن لا تكون على سطر واحد).

البرنامج:

```
//This program uses 2D_array for input and output
#include<iostream.h>
#include<iomanip.h>
int main( )
{
    int i,j;
    float a[3][4];
    cout<<"Enter the array elements\n " ;
```

```

for(i=0;i<3;i++)
{
    for(j=0;j<4;j++)
    {
        cout<<"a["<<i<<"]["<<j<<"]:";
        cin>>a[i][j];
    }
}
for(i=0;i<3;i++)
{
    for(j=0;j<4;j++)
        cout<<a[i][j]<<setw(3);
    cout<<endl;
}
return 0;
} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the array elements

a[0][0]:1

a[0][1]:2

a[0][2]:3

a[0][3]:4

a[1][0]:5

a[1][1]:6

a[1][2]:7

a[1][3]:8

a[2][0]:9

a[2][1]:10

a[2][2]:11

a[2][3]:12

1 2 3 4

5 6 7 8

9 10 11 12

Press any key to continue

مثال (8-9)

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال درجات 10 طلاب في خمسة مقررات وتخزينها في مصفوفة ثنائية البعد $a[10][5]$.
- حساب وطباعة معدل كل طالب في المقررات الخمسة .
- حساب وطباعة المعدل الأكبر .

البرنامج:

```

//multi-dimensional arrays
#include<iostream.h>

void main( )
{
    int Mark[10][5] , ormax ;
    double sum=0.0 , avr[10], maxav ;
    cout << " Enter Mrka[ ] [ ] values : " << endl;
    for (int i=0; i<10; i++ )
        for (int j=0; j<5; j++ )
            cin >> Mark[ i ] [ j ] ;
    for (int k=0; k<10; k++) {
        sum=0.0 ;
        for (int j=0; j<5; j++){
            sum=sum + Mark[ k ] [ j ] ;
        }
        avr[ k ]=sum / 5 ;
    }
    maxav = avr[0];
    ormax =1;
    for (int i2=0; i2<10; i2++) {
        if( avr[ i2 ] > maxav ) {
            maxav = avr[ i2 ] ;
            ormax = i2 ;
        }
    }
}

```



```

    }
}

for (int i1=0; i1<10; i1++) {
    cout<< " av ["<<i1<< "]= " << avr[i1] << endl;
    cout<<" The greatest average is " << maxav << endl ;
    cout<< " The greatest number is " << ormax <<endl;
}

} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

61	65	66	71	78
66	68	70	68	77
77	78	89	80	67
66	77	88	90	67
78	88	68	78	67
68	89	90	78	88
67	78	89	67	78
87	67	76	89	67
89	78	89	78	78
87	67	78	89	89

av [0]=68.2
 av [1]=69.8
 av [2]=78.2
 av [3]=77.6
 av [4]=75.8
 av [5]=82.6
 av [6]=75.8
 av [7]=77.2
 av [8]=82.4
 av [9]=82

The greatest
 average is 82.6
 The greatest
 number is 5
 Press any key to continue

مسائل عامة محلولة

١- اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال 12 عدداً صحيحاً وتخزينها في مصفوفة ثنائية البعد $a[4][3]$.
- إدخال 6 أعداد صحيحة وتخزينها في مصفوفة ثنائية البعد $b[3][2]$.
- إجراء عملية ضرب المصفوفتين $a[4][3]$ و $b[3][2]$ وتخزين النتيجة في المصفوفة $c[4][2]$.
- طباعة عناصر المصفوفات الثلاثة $a[4][3]$ و $b[3][2]$ و $c[4][2]$.

توضيح رياضي: لنفرض أن:

$$a[4][3] = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 5 & 6 & 7 \end{bmatrix}$$

عندئذ:

$$a[4][3] \times b[3][2] = \begin{bmatrix} 3 & 4 & 5 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 5 & 6 & 7 \end{bmatrix} \times \begin{bmatrix} 2 & 2 \\ 3 & 7 \\ 9 & 2 \end{bmatrix}$$

ومنه نجد:

$$a[4][3] \times b[3][2] = \begin{bmatrix} 3*2 + 4*3 + 5*9 & 3*2 + 4*7 + 5*2 \\ 1*2 + 1*3 + 1*9 & 1*2 + 1*7 + 1*2 \\ 2*2 + 2*3 + 2*9 & 2*2 + 2*7 + 2*2 \\ 5*2 + 6*3 + 7*9 & 5*2 + 6*7 + 7*2 \end{bmatrix}$$

وبالتالي:

$$a[4][3] \times b[3][2] = \begin{bmatrix} 63 & 44 \\ 14 & 11 \\ 28 & 22 \\ 91 & 66 \end{bmatrix}$$

وحيث أن دستور ضرب المصفوفتين بشكل رياضي هو التالي:

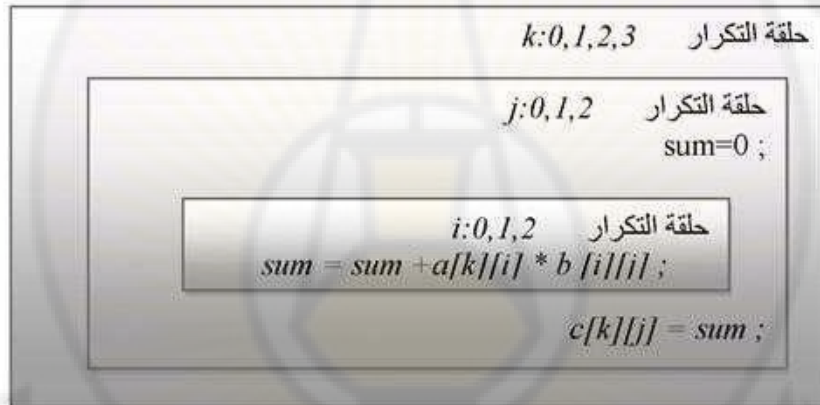
$$c[k][j] = \sum_{i=0}^2 a[k][i] \times b[i][j]; ; k = 0,1,2,3; ; j = 0,1$$

والعبارات البرمجية المقابلة لهذا الشكل الرياضي هي التالية:

$sum = sum + a[k][i] * b[i][j];$

$c[k][j] = sum$

وحيث أن هذه العبارات البرمجية بحاجة إلى ثلاث حلقات تكرار متداخلة هي:



والآن سنقدم البرنامج اللازم لضرب المصفوفات .

البرنامج:

```
//This program multiples 2D_array arrays
```

```
#include<iostream.h>
```

```
#include<iomanip.h>
```

```

void main( )
{
    int i,j,k,s;
    int a[4][3]={ {3,4,5},{1,1,1},{2,2,2},{5,6,7}};
    int b[3][2]={ {2,2},{3,7},{9,2}};
    int c[4][2];
    for(k=0;k<4;k++)
    {
        for(j=0;j<2;j++)
        {
            s=0;
            for(i=0;i<3;i++)
            {
                s+=a[k][i]*b[i][j];
            }\\end i loop
            c[k][j]=s;
        }\\end j loop
    }\\end k loop
    cout<<"-----The first array is -----"<<endl;

    for(i=0;i<4;i++)
    {

```

```

        for(j=0;j<3;j++)
        {
            cout<<a[i][j]<<setw(5);
        }\\end j loop
        cout<<endl;
    }\\end i loop
    cout<<"-----The second array is -----"<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<2;j++)
        {
            cout<<b[i][j]<<setw(5);
        }\\end j loop
        cout<<endl;
    }\\end i loop
    cout<<"-----The result array is -----"<<endl;
    for(i=0;i<4;i++)
    {
        for(j=0;j<2;j++)
        {
            cout<<c[i][j]<<setw(5);
        }\\end j loop
        cout<<endl;
    }

```



```
}\\end i loop  
}\\The end of main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT  
-----The first array is -----  
3  4  5  
1  1  1  
2  2  2  
5  6  7  
-----The second array is -----  
2  2  
3  7  
9  2  
-----The result array is -----  
63 44  
14 11  
28 22  
91 66  
Press any key to continue
```

٢- اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال 9 أعداد صحيحة وتخزينها في مصفوفة ثنائية البعد `a [3][3]` .
- حساب مجموع ومتوسط الأعداد التي تقع على وفوق القطر الرئيسي للمصفوفة.
- طباعة عناصر المصفوفة المدخلة على شكل مصفوفة مربعة.
- طباعة مجموع ومتوسط الأعداد الذين تم إيجادهما .

توضيح رياضي:لنأخذ الشكل الرياضي للمصفوفة ثنائية البعد :

$$a[3][3] = \begin{bmatrix} a[0][0] & a[0][1] & a[0][2] \\ a[1][0] & a[1][1] & a[1][2] \\ a[2][0] & a[2][1] & a[2][2] \end{bmatrix}$$

من خلال هذا الشكل الرياضي ، نلاحظ أن عنصر المصفوفة $a[i][j]$ يقع على القطر الرئيسي إذا كان رقم السطر i مساوياً لرقم العمود j ويقع عنصر المصفوفة $a[i][j]$ فوق القطر الرئيسي إذا كان رقم السطر i أصغر من رقم العمود j ، بينما يقع عنصر المصفوفة $a[i][j]$ تحت القطر الرئيسي إذا كان رقم السطر i أكبر من رقم العمود j .

البرنامج:

```
/*This program computes the sum and average of array elements which are located on and over the main diagonal */
```

```
#include<iostream.h>
```

```
void main( )
```

```
{
```

```
int i,j,a[3][3],c=0;
```

```
float sum=0.0,av;
```

```
cout<<"Enter the array elements:"<<endl;
```

```
for(i=0;i<3;i++)
```

```
{
```

```
for(j=0;j<3;j++)
```

```
{
```

```
cout<<"a["<<i<<"]["<<j<<"].";
```

```

        cin>>a[i][j];
    }
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(i<=j)
        {
            sum+=a[i][j];
            c+=1;
        }
    }
}
av=sum/c;
cout<<"The sum of elements is "<<sum<<endl;
cout<<"The average of elements is "<<av<<endl;
cout<<"The elements of entered 2D_array are:"<<endl;
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    cout<<a[i][j]<<" ";
    cout<<endl;
}

```

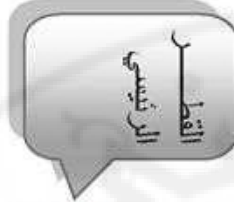
```
}  
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT  
Enter the array elements:  
a[0][0]:7  
a[0][1]:5  
a[0][2]:6  
a[1][0]:8  
a[1][1]:7  
a[1][2]:4  
a[2][0]:5  
a[2][1]:9  
a[2][2]:5  
The sum of elements is 34  
The average of elements is 5.66667  
The elements of entered 2D_array are:  
7    5    6  
8    7    4  
5    9    5  
Press any key to continue
```

ملاحظة

لنأخذ الشكل الرياضي للمصفوفة المربعة الثنائية البعد $a[5][5]$:



$$a[5][5] = \begin{bmatrix} a[0][0] & a[0][1] & a[0][2] & a[0][3] & a[0][4] \\ a[1][0] & a[1][1] & a[1][2] & a[1][3] & a[1][4] \\ a[2][0] & a[2][1] & a[2][2] & a[2][3] & a[2][4] \\ a[3][0] & a[3][1] & a[3][2] & a[3][3] & a[3][4] \\ a[4][0] & a[4][1] & a[4][2] & a[4][3] & a[4][4] \end{bmatrix}$$

من خلال هذا الشكل الرياضي للمصفوفة المربعة (عدد الأسطر يساوي عدد الأعمدة) ، سنقدم الجدول الذي من خلاله سنتعرف على موقع كل عنصر $a[i][j]$ من عناصر المصفوفة المربعة الثنائية البعد وذلك بالنسبة لكلا القطرين الرئيسيين والثانوي.

القطر	على	فوق	تحت
الرئيسي	$i=j$	$i < j$	$i > j$
الثانوي	$i+j=size-1$	$i+j < size-1$	$i+j > size-1$

حيث $size$ هو بعد المصفوفة (وفي مثالنا: $size = 5$ ، عدد الأسطر يساوي عدد الأعمدة ويساوي ٥).

مسائل عامة غير محلولة

١- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- إدخال 5 أعداد صحيحة وتخزينها في مصفوفة أحادية البعد [5] n .
- حساب مربعات ومكعبات الأعداد المدخلة وتخزينها في المصفوفتين s[5] و c[5] على الترتيب.
- طباعة عناصر المصفوفات الثلاثة على أسطر مستقلة.

=====

٢- اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال 25 عدداً صحيحاً من لوحة المفاتيح وتخزينها في مصفوفة ثنائية البعد y [5][5] .
- إيجاد العدد الأصغر من بين جميع الأعداد الواقعة على وفوق القطر الرئيسي.
- طباعة عناصر المصفوفة المدخلة على شكل مصفوفة مربعة.
- طباعة قيمة العدد الأصغر الذي تم إيجاده.

=====

٣- اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال 25 عدداً حقيقياً من لوحة المفاتيح وتخزينها في مصفوفة ثنائية البعد x [5][5] .
- حساب المتوسط الحسابي لجميع الأعداد الواقعة تحت القطر الثانوي.
- طباعة عناصر المصفوفة المدخلة على شكل مصفوفة مربعة.
- طباعة قيمة المتوسط الحسابي الناتج.

- ٤- اكتب برنامجاً بلغة C++ يقوم بما يلي :
- إدخال 16 عدداً حقيقياً من لوحة المفاتيح وتخزينها في مصفوفة ثنائية البعد [4][4] x .
 - حساب المتوسط الحسابي لعناصر المصفوفة الواقعة في السطر الأول.
 - إيجاد العنصر الأكبر من بين عناصر المصفوفة الواقعة في العمود الأول.
 - طباعة عناصر المصفوفة المدخلة على شكل مصفوفة مربعة.
 - طباعة النتائج التي تم التوصل إليها من خلال الطالبين الثاني والثالث.



الفصل التاسع

التوابع

Functions



التوابع

٩-١- مقدمة

تتجلى إحدى الفوائد الأساسية من استخدام التوابع عند كتابة البرامج بتقسيم العمليات الحسابية الطويلة والمعقدة إلى مجموعات من العمليات الصغيرة والبسيطة نسبياً، وينتج عن ذلك إمكانية إعادة استخدام هذه التوابع في العديد من البرامج بدلاً من إعادة كتابة العمليات نفسها في كل مرة، وتسمح التوابع، عند كتابتها بالطريقة المناسبة، بإخفاء التفاصيل الصغيرة للعمليات مما يسهل على المبرمج توضيح البنية العامة لبرامجه ويخفف من صعوبة تعديل هذه البرامج، وقد برهنت التجربة على أن أفضل طريقة لتطوير وصيانة برنامج كبير، تتمثل في عملية تجزئته إلى قطع وأجزاء أصغر يمكن التحكم بها بسهولة أكبر من البرنامج الأصلي.

لقد أضافت لغة C++ عدة مزايا للتوابع زيادة عن لغة C، ومن أهمها:

- ١- الوسطاء المرجعية.
- ٢- الوسطاء الافتراضية.
- ٣- الأنواع المرجعية المُعادة.
- ٤- التوابع الأعضاء.
- ٥- التحميل الزائد للتوابع.
- ٦- توابع العمليات.
- ٧- التوابع البانية والمدمرة.
- ٨- توابع التحويل.
- ٩- التوابع الافتراضية.

١٠- قوالب التوابع.

١١- توابع معالجة الاستثناءات.

١٢- بواني التهيئة .

٩-٢- فوائد استخدام التوابع

من أهم فوائد التوابع في البرمجة ، هي :

١. تساعد التوابع المخزنة في ذاكرة الحاسوب على اختصار البرنامج إذ يكتفى باستدعائها باسمها فقط لنقوم بالعمل المطلوب .
٢. تساعد البرامج المخزنة في ذاكرة الحاسوب أو التي يكتبها المستخدم على تلافي عمليات التكرار في خطوات البرنامج التي تتطلب عملاً مشابهاً لعمل تلك التوابع .
٣. تساعد التوابع الجاهزة في تسهيل عملية البرمجة.
٤. يوفر استعمال التوابع من المساحات المستخدمة في الذاكرة.
٥. كتابة برنامج في ++C بشكل تابع واضح المعالم يجعل البرنامج واضحاً لكل من المبرمج والقارئ على حد سواء.
٦. التوابع تمكن المبرمج من تقسيم البرنامج إلى وحدات modules، وكل تابع في البرنامج يمثل وحدة قائمة بذاتها، ولذا نجد أن المتغيرات المعرفة في التابع تكون متحولات محلية (Local) ونعني بذلك أن المتحولات تكون معروفة فقط داخل التابع.

٩-٣- إنشاء التوابع في البرنامج

عادة يتم كتابة برامج ++C بأن يتم الجمع بين التوابع المكتوبة من قبل المبرمج

والتوابع الجاهزة المتوفرة في المكتبة المعيارية للغة C++ ، والتي تحتوي على

مجموعة غنية من التوابع الخاصة تقوم بما يلي :

- تنفيذ العمليات الحسابية الرياضية الشائعة.
 - تنفيذ عمليات المعالجة على سلاسل المحارف.
 - تنفيذ عمليات الإدخال والإخراج .
 - تنفيذ عمليات التحقق وكشف الأخطاء وغيرها من العمليات المفيدة.
- إن هذه العمليات تساعد المبرمج وتسهل عليه عملية البرمجة، لأنها توفر الكثير من الإمكانيات التي قد يحتاج إليها.

التابع Function، هو مجموعة من التعليمات التي تنفذ عندما يتم استدعائه من مكان ما في البرنامج ، وفي الواقع، التابع هو برنامج جزئي يمكن أن ينفذ على معطيات معينة وبعد ذلك يعيد قيمة إلى التابع المستدعي.

كل برنامج بلغة C++ يملك تابع واحد على الأقل هو التابع الرئيسي main()، والذي يتم استدعائه بشكل آلي عند تنفيذ البرنامج، فالتابع الرئيسي main() يمكن أن يستدعي توابع معينة والتي بدورها قد تستدعي نفسها أو تستدعي توابع أخرى .

عند استخدام أي تابع مُعد من قبل المبرمج في برنامج بلغة C++، يجب

أن يتوفر لهذا التابع ثلاثة عناصر أساسية هي:

- التصريح عن التابع Declaring Function .
- استدعاء التابع Calling Function .
- تعريف التابع Defining Function .

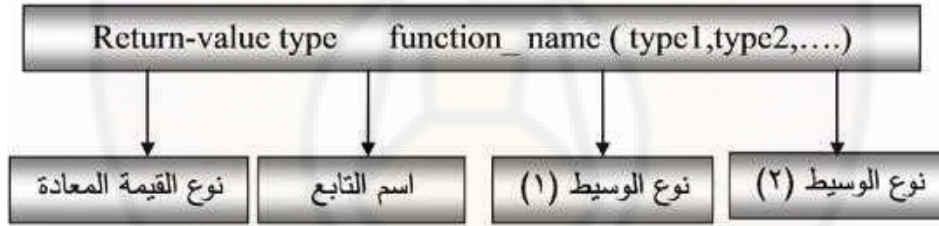
٩-٣-١- التصريح عن التابع

Declaring Function

كل تابع يُستخدم في برنامج بلغة C++ يجب أن يصرح عنه قبل الاستخدام ، ويكون التصريح عادة بعد ترويسة (الملف الرأسي) البرنامج المعروفة :

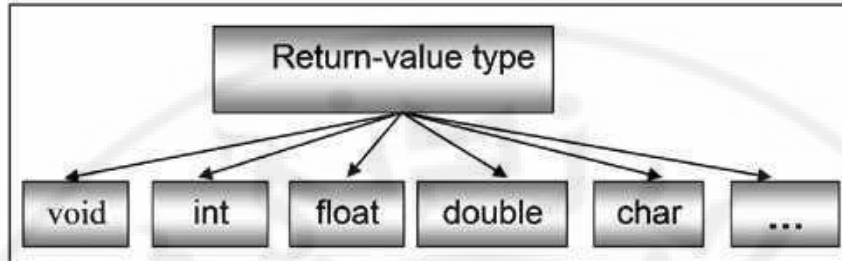
```
#include <iostream.h>
```

والتصريح عن التابع يعني بالنسبة للمترجم Compiler ، إخباره عن اسم التابع، ونوع القيمة المعادة ونوع الوسطاء وعددها ، ويبين الشكل (٩-١) الشكل العام للتصريح عن التابع .



الشكل (٩-١) الشكل العام للتصريح عن التابع

نلاحظ أن كل تابع يملك الاسم الخاص به والذي يوضع من قبل المبرمج ، وغالباً يكون اسم التابع قريباً من محتواه، وفيما يتعلق بنوع القيمة المعادة ، تحددتها طبيعة المسألة المدروسة، فقد تكون من أي نوع من أنواع المعطيات المعروفة في لغة C++ ، كما هو موضح الشكل (٩-٢) .



الشكل (٩-٢) نوع القيمة المعادة

وعندما يولد المترجم تعليمات لاستدعاء تابع ما فإنه يحتاج إلى معرفة اسم التابع وعدد وسطائه وأنواعها ونوع القيمة المعادة، لذا علينا كتابة نموذج أو (تصريح) للتابع قبل إجراء أي استدعاء له ، وبالتالي تصريح التابع هو سطر واحد يبلغ المترجم عن اسم التابع وعدد وسطائه وأنواعها ونوع القيمة المعادة بواسطة التابع ، ويشبه تصريح التابع ، السطر الأول في تعريف التابع، لكن تليه فاصلة منقوطة، فمثلا في تصريح التابع التالي:

```
int sum(int) ;
```

النوع **int** بين القوسين يخبر المترجم بأن الوسيط الذي سيتم تمريره إلى التابع سيكون من النوع **int** و **int** التي تسبق اسم التابع تشير إلى نوع القيمة المعادة بواسطة التابع، ويمكن أن تكون من نمط **double** أو **char** كما يلي :

```
double sum1(double) ;
```

```
char sum2(char) ;
```

أو لا يعيد قيمة وهو من نمط **void**:

```
void sum(int , double) ;
```


لنفترض أن لدينا تابع اسمه `sum` ويأخذ هذا التابع وسيطين من النوع `int` وحيث أن هذا التابع يقوم بجمع عددين صحيحين، عندئذ يمكن أن نصرح عن هذا التابع كما يلي:

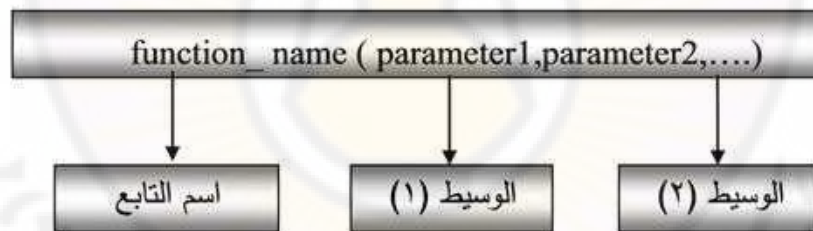
```
int sum ( int , int );
```

من الواضح أن نوع القيمة المعادة هو `int` ، لأن حاصل جمع عددين صحيحين هو عدد صحيح.

٩-٣-٢ - استدعاء التابع

Calling Function

كل تابع يستخدم في برنامج بلغة `C++` يجب أن يتم استدعاؤه من التابع الرئيسي `main` أو من أي تابع آخر، ويبين الشكل (٩-٣) الشكل العام للاستدعاء.



الشكل (٩-٣) - يبين الشكل العام للاستدعاء

يؤدي استدعاء التابع إلى انتقال التنفيذ إلى بداية التابع، ويمكن تمرير بعض الوسائط إلى التابع عند استدعائه وبعد الانتهاء من تنفيذ التابع يعود التنفيذ للعبارة التي تلي استدعاء التابع في البرنامج الأصلي ، وبإمكان التابع أن يعيد قيمة إلى العبارة التي قامت باستدعائه، وإذا كان التابع لا يعيد شيئاً يجب استعمال الكلمة الأساسية `void` كنوع إعادة له للإشارة إلى ذلك .

يوجد ثلاث طرق يمكن بها إرجاع التحكم إلى النقطة التي تم فيها استدعاء التابع ، وهي :

١. إذا كان التابع لا يرجع قيمة ، يرجع التحكم تلقائياً عند الوصول إلى نهاية التابع .

٢. باستخدام العبارة `return ;`

٣. إذا كان التابع يرجع قيمة فالعبارة `return expression ;` تقوم بإرجاع قيمة التعبير `expression` إلى النقطة التي استدعته .

مثلاً : لنصرح عن التابع `sum` يعيد قيمة من النوع `int` ويحتوي على وسيطين من النوع `int` ، ومن أجل استدعاء هذا التابع يجب أن يحتوي برنامج `C++` الذي يتضمن هذا التابع، على الاستدعاء التالي للتابع:

`sum ( a , b );`

حيث أن:

`sum` - اسم التابع.

`a` - الوسيط الأول (يجب أن يُعطى قيمة محددة أو يتم ادخله من قبل المستخدم).

`b` - الوسيط الثاني (يجب أن يُعطى قيمة محددة أو يتم ادخله من قبل المستخدم).

من غير الضروري أن تكون أسماء الوسطاء في التصريح هي نفسها المستعملة في تعريف التابع ، في الواقع، المترجم يتجاهلها لكنها تكون مفيدة أحياناً للذين يقرؤون البرنامج ، فمثلاً لنفترض أن الوسيطين `x` و `y` تمثلان إحداثيات نقطة على الشاشة، عندئذ لنأخذ التصريح التالي:

`void draw_dot ( int, int );` [تصريح التابع]

هذا التصريح كافٍ للمعرف لكن المبرمج قد لا يعرف أيهما `x` و `y`، لذلك سيكون من المفيد أن نكتب :

`void draw_dot (int x, int y);`

٩-٣-٣- تعريف التابع

Defining Function

كل تابع يُستخدم في برنامج بلغة C++ يجب أن يكون له تعريف داخل البرنامج (عدا التوابع الرياضية والتوابع الجاهزة الأخرى)، وتعريف التابع يتضمن ترويسة التابع وجسم التابع، حيث أن ترويسة التابع هي بالضبط نموذج التابع (التصريح عن التابع) ولكن هنا يجب ذكر اسم الوسيط بعد النوع، بالإضافة إلى عدم استخدام الفاصلة المنقوطة في نهاية التصريح، بينما جسم التابع هو مجموعة التعليمات المحصورة بين قوسين { } ، والتي من خلالها يقوم التابع بتنفيذ مهمته.

لا بد من الإشارة إلى أن تعريف التابع يُعطي علماً للمترجم كيف يعمل التابع (مهمة التابع)، ويبين الشكل (٩-٤) الشكل العام لتعريف التابع .

Return-value type function_name (parameter list)

{

جسم التابع

(declarations and statements)

}

الشكل (٩-٤) الشكل العام لتعريف التابع

حيث:

- ✓ Return-value type : نوع القيمة المعادة بواسطة التابع والذي يمكن أن يكون أي نوع من أنواع معطيات C++، وإذا كان التابع لا يرجع أي قيمة يكون نوع إعادته void.
- ✓ function_name: اسم التابع والذي يتبع في تسميته قواعد تسمية المعرفات (identifiers).
- ✓ parameter list: هي لائحة الوسطاء الممرة إلى التابع والتي يمكن أن تكون خالية (void) أو تحتوى على وسيط واحد أو عدة وسطاء تفصل بينها فاصلة ويجب ذكر كل وسيط على حدة.
- ✓ declarations and statements: تمثل جسم التابع والذي يطلق عليه في بعض الأحيان كتلة (block) ، ويمكن أن تحتوى الكتلة (block) على تصريح عن المتحولات ولكن تحت أي ظرف لا يمكن أن يتم تعريف تابع داخل جسم تابع آخر.
- ✓ السطر الأول في تعريف التابع يدعى المصريح declarator والذي يحدد اسم التابع ونوع المعطيات الذي يعيدها التابع وأسماء وسطائه وأنواعها .

مثال (1-9)

لقد صرحنا سابقاً عن التابع sum أنه من النوع int ، وبوسيطين من النوع int ، ويمكن تعريف التابع sum كما يلي :

```
int sum ( int x , int y )  
{  
    int s;
```

```

s=a+b;
return s;
}

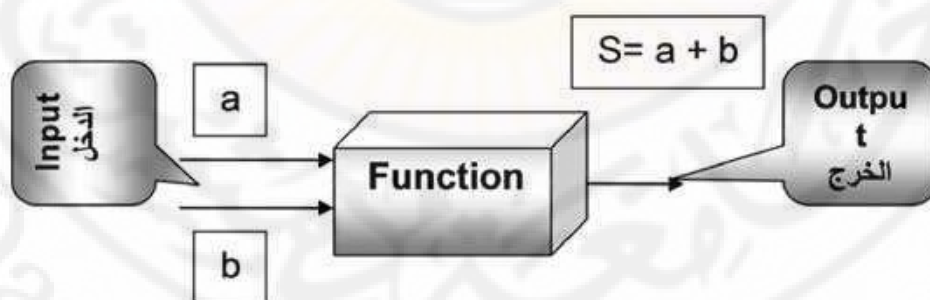
```

ويوضح المثال (9 - 2) كيفية استخدام التابع sum في البرنامج .

مثال (2-9)

اكتب برنامجاً بلغة ++C يقوم بما يلي:

- إدخال عددين صحيحين a و b .
 - يحتوي على تابع اسمه sum ، مهمته (وظيفته) حساب مجموع عددين صحيحين.
 - يطبع النتيجة (المجموع) في التابع الرئيسي main .
- من المعروف أن كل برنامج يحتاج إلى مدخلات (معطيات) ومن خلال هذه المدخلات يستطيع أن يُعطي النتائج المطلوبة (المُخرجات)، وإن كل تابع ، في غالب الأحيان يكون له دخل وخرج كما هو موضح في الشكل (9-5) .



الشكل (9-5) - الدخل والخرج في التابع

البرنامج:

```
#include<iostream.h>

int sum(int,int);    // Function Prototype

int main()
{
    int a,b;
    cout<<"Enter the value of a and b \n";
    cin>>a>>b;
    // Calling Function
    cout<<"The sum of "<<a<<"and "<<b<<"is"
        <<Sum(a,b)<<endl;
    return 0;
}

// Defining Function
int sum(int x,int y)
{
    int s;
    s=x+y;
    return s;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of a and b

10

20

The sum of 10 and 20 is 30

Press any key to continue

كل المتحولات المصرح عنها داخل التابع تُدعى متحولات محلية Local Variables ، وتكون معرفة فقط في التابع الحاوي عليها ، بينما المتحولات المصرح عنها في التابع الرئيسي main() تُدعى متحولات عامة Global Variables وتكون معرفة فقط في التابع الرئيسي. التابع الرئيسي main() يمكن أن يستدعي نوابغ أخرى، وهذه النوابغ بدورها يمكن أن تستدعي نفسها أو نوابغ أخرى.

مثال (3-9)

اكتب برنامجاً بلغة C++ يقوم بما يلي:

- يُدخل عددين حقيقيين a و b .
- يستخدم تابع اسمه max ، مهمته (وظيفته) إيجاد العدد الأكبر من بين العددين المدخلين.
- يطبع النتيجة (قيمة العدد الأكبر) في التابع الرئيسي main .

البرنامج:

```

#include<iostream.h>

float max(float , float );// Function Prototype

int main( )
{
    float a,b;
    cout<<"Enter the values  of a and b";
    cin>>a>>b;
    // Calling Function
    cout<<"The maximum  of "<<a<<"and  "<<b<<"is"
        <<max(a,b)<<endl;

    return 0;
}

// Defining Function
float max(float x,float y)
{
    if (x>y)
        return x;
    else
        return y;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of a and b

5.7

10.3

The maximum of 5.7 and 10.3 is 10.3

Press any key to continue

في المثال (9-3) يمكن تعريف التابع max بطريقة أخرى نستخدم فيها مفهوم المتحول المحلي من أجل إسناد القيمة العظمى التي تم إيجادها وبعد ذلك يقوم التابع بإرجاع هذه القيمة من خلال المتحول المحلي إلى التابع المستدعي والذي هو التابع الرئيسي:

```
float max(float x,float y)
{
    float mx; //Local Variables
    if (x>y)
        mx=x;
    else
        mx=y;
    return mx
}
```

٩ - ٤ - مكتبة التوابع الرياضية

Math Library Functions

تحتوى مكتبة التوابع الرياضية على العديد من التوابع التي تستخدم في

تنفيذ العمليات الرياضية الحسابية، فمثلاً المبرمج الذي يرغب في حساب وطباعة

الجذر التربيعي للعدد 900 قد يكتب العبارة التالية :

```
cout << sqrt ( 900);
```

عند تنفيذ هذه العبارة يتم استدعاء التابع المكتبي (sqrt) لحساب الجذر التربيعي

للعدد بين القوسين (900)، ويسمى العدد بين القوسين وسيط التابع argument

وعليه فالعبارة السابقة تقوم بطباعة العدد 30 .

ويأخذ التابع (sqrt) وسيط من النوع double وتكون النتيجة قيمة من

نفس النوع وينطبق هذا على جميع التوابع الرياضية.

عند استعمال التوابع الرياضية في أي برنامج بلغة C++ يجب تضمين

الملف math.h والذي يحتوى على هذه التوابع، ويوضح الجدول (٩-١) أهم

التوابع الرياضية التي يمكن استخدامها من خلال مكتبة لغة C++ ، ويتضمن

الجدول أيضاً على وصف التوابع وأمثلة عليها .

Function	Description	Example
sqrt(x)	x الجذر التربيعي لـ	sqrt (9.0) is 3
pow(a,b)	الرفع إلى قوة	a^b pow(10,5)
exp(x)	exp(1.0) is 2.718282	e^x
fabs(x)	x القيمة المطلقة لـ	if $x > 0$ fabs(x) = x = 0 fabs(x) = 0 < 0 fabs(x) = -x
ceil(x)	لأصغر عدد صحيح x تقريب أكبر من	ceil(9.2) is 10.0 ceil(-9.8) is -9.0
floor(x)	لأكبر عدد صحيح x تقريب أصغر من	floor(9.2) is 9 floor(-9.8) is -10.0
log(x)	x اللغوريتم الطبيعي لـ	log (2.718282) is 1.0
log10(x)	x اللغوريتم العشري لـ	log10(1.0) is 0.0
sin(x)	x تابع الجيب لـ	sin(0.0) is 0.0
cos(x)	x تابع التنجيب لـ	cos(0.0) is 1.0
tan (x)	x تابع الظل لـ	tan(0.0) is 0.0

الجدول (٩-١) - التوابيع الرياضية لمكتبة لغة C++

مثال (4-9)

يوضح البرنامج التالي كيفية استخدام بعض التوابع الرياضية:

البرنامج:

```
#include<iostream.h>
#include<math.h>
int main( )
{
    double x;
    int y,z;
    cout<<"Enter any float number:";
    cin>>x;
    cout<<"Round "<<x <<" to next integer is  "
        <<ceil(x)<<endl;
    cout<<"Round "<<x <<" to previous integer is  "
        <<floor(x)<<endl;
    cout<<"Enter another number :";
    cin>>y;
    cout<<"The square root of "<<y<<" is "<<sqrt(y)<<endl;
    cout<<"The exponential of "<<y<<" is "<<exp(y)<<endl;
    cout<<"The log  of "<<y<<" is "<<log(y)<<endl;
    cout<<"The log10 of "<<y<<" is "<<log10(y)<<endl;
```

```

cout<<"Enter another number :";
cin>>z;
cout<<"The absolute value of "<<z<<" is "<<fabs(z)<<endl;
return 0;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```

Enter any float number:45.75
Round 45.75 to next integer is 46
Round 45.75 to previous integer is 45
Enter another number :9
The square root of 9 is 3
The exponential of 9 is 8103.08
The log of 9 is 2.19722
The log10 of 9 is 0.954243
Enter another number :-6
The absolute value of -6 is 6
Press any key to continue

```

٩-٥- التتابع بدون وسطاء

Functions with Empty Parameter Lists

في لغة C++ يُكتب التابع الذي لا يمتلك وسطاء إما بكتابة void بين القوسين الذين يتبعان اسم التابع أو تركهما فارغين ، فمثلاً التصريح عن التابع :

```
void print ( ) ;
```

يشير إلى أن التابع print لا يأخذ أي وسيط وهو لا يرجع قيمة ، ويبين المثال (5-9) الطريقتين اللتين تكتب بهما التابع التي لا تأخذ وسطاء .

مثال (5-9)

يوضح البرنامج كيفية استخدام تابع بدون وسطاء و لا تُعيد أية قيمة.

```
// Functions that take no arguments
#include <iostream.h>
void f1 ();
void f2 ( void );
void main()
{
    f1 ();
    f2 ();
} //End main
void f1 () {
    cout << "Function f1 takes no arguments" << endl;
}
void f2 ( void ) {
    cout << "Function f2 also takes no arguments" << endl;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Function f1 takes no arguments

Function f2 also takes no arguments

Press any key to continue

٩-٦ - التابع الفورية (السطرية)

Inline Functions

تحتاج بعض التطبيقات التي يعتبر فيها وقت تنفيذ البرنامج أمراً حيوياً وحاسماً، لإبدال عملية استدعاء واستخدام تابع بما يسمى التابع الفوري (السطري). ويتم تعريف التابع الفوري باستعمال نفس التركيب القواعدي المستخدم لتعريف التابع العادي ، لكن بدلاً من وضع جسم التابع في مكان مستقل ، يضعها المترجم في السياق الطبيعي للبرنامج مكان ظهور استدعاء التابع، ويتم جعل التابع كتابع فوري عن طريق استخدام الكلمة الأساسية **inline** في تعريف التابع، و يستخدم التابع الفوري فقط إذا كان التابع قصير ويستخدم مرات عديدة في البرنامج.

مثال (9-6)

أكتب برنامجاً بلغة C++ يقوم بحساب وطباعة مكعب أي عدد صحيح يتم إدخاله من لوحة المفاتيح ، حيث أن التابع المستخدم يجب أن يكون تابعاً فورياً (سطرياً) .

البرنامج:

```
#include <iostream.h>
```

```
inline float cube(float s)
```

```
{
```

```
    return s*s*s;
```

```
}
```

```
void main( )
```

```
{
```

```
    cout<<"\n Enter the side length of your cube : ";
```

```

float side;
cin>>side;
cout<<" volume of cube is " <<cube(side) <<endl;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the side length of your cube : 6

volume of cube is 216

Press any key to continue

مثال (7-9)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة جداء أي عددين صحيحين يتم إدخالهما من لوحة المفاتيح، حيث أن التابع المستخدم يجب أن يكون تابعاً فورياً (سطرياً) .

البرنامج:

```

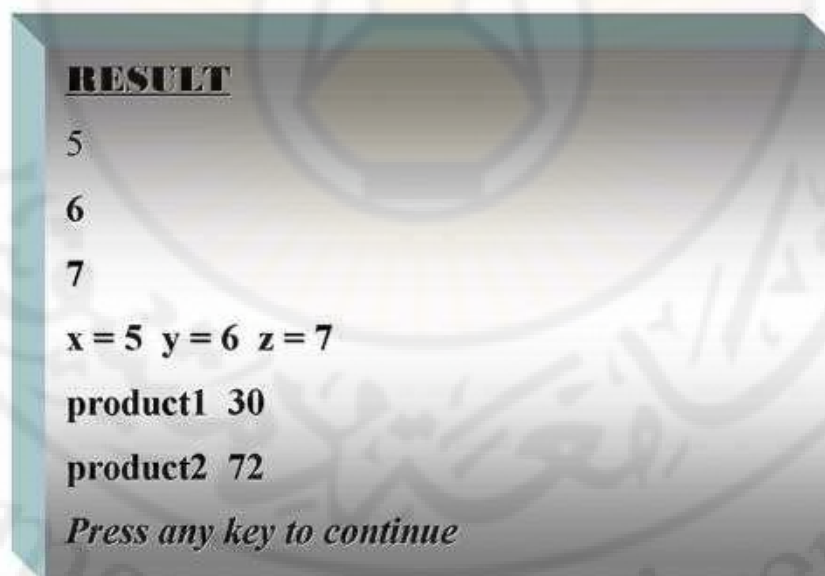
#include <iostream.h>
inline int mult( int a, int b)
{
    return (a*b);
}

```



```
void main( )  
{  
    int  x, y, z;  
    cin >> x >> y >> z;  
    cout << "x = " << x << " y = " << y << " z = " << z << endl;  
    cout << "product1" << mult (x ,y) << endl;  
    cout << "product2" << mult (x +z, y) << endl;  
} //End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



```
RESULT  
5  
6  
7  
x = 5 y = 6 z = 7  
product1 30  
product2 72  
Press any key to continue
```

٩-٧- التحميل الزائد للتوابع

Overloading Functions

التحميل الزائد للتوابع يعنى استخدام الاسم نفسه لعدة توابع، لكن كل تابع يجب أن يكون له تعريف مستقل، وعند استدعاء التابع، يبحث المترجم عن نوع وسطاء التابع وعددها لمعرفة التابع المقصود، ولكي يميز المترجم بين تابع وآخر يحمل الاسم نفسه، يقوم بعملية تعرف بتشويه الأسماء (names mangling)، وتتألف هذه العملية من إنشاء اسم جديد خاص بالمترجم عن طريق دمج اسم التابع مع أنواع وسطائه.

مثال (8-9)

البرنامج التالي يقوم بتحميل التابع square بشكل زائد وذلك من أجل حساب وطباعة الجذر التربيعي لعددتين من النوعين int و double على الترتيب .
البرنامج:

```
#include <iostream.h>
int square(int x) ;
double square(double y) ;
void main ( )
{
    cout<< " The square of integer 7 is"<<square(7) << endl
    cout<<"The square of double 7.5 is "<<square(7.5)<< endl;
} //End main
```

```
int square(int x)
```

```
{  
    return x*x;  
}
```

```
double square(double y)
```

```
{  
    return y*y;  
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

The square of integer 7 is 49

The square of double 7.5 is 56.25

Press any key to continue

مثال (9-9)

البرنامج التالي يقوم بتحميل التابع abs بشكل زائد وذلك من أجل حساب وطباعة القيمة المطلقة لثلاثة أعداد من الأنواع int و double و long على الترتيب

البرنامج:

```
#include <iostream.h>

int abs (int );
double abs(double );
long abs(long );
void main( )
{
    cout<< abs (-10)<<"\n";
    cout<< abs (-11.0)<<"\n";
    cout<< abs (-9L)<<"\n";
} //End main

int abs ( int i )
{
    cout<<"using integer abs( )\n";
    return  i<0 ? -i : i ;
}

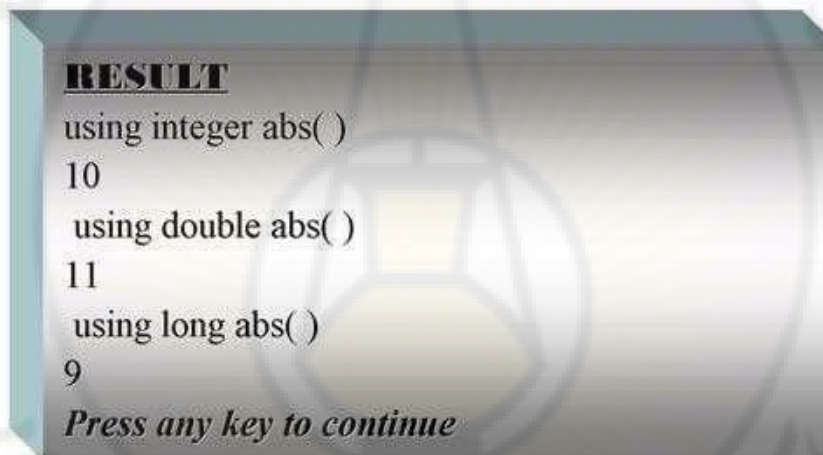
double abs ( double d )
{
    cout<<" using double abs( )\n";
    return  d<0.0 ? -d : d ;
}
```

```

long abs( long l )
{
    cout<<" using long abs( )\n";
    return l<0.0 ? -l : l;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



```

RESULT
using integer abs( )
10
using double abs( )
11
using long abs( )
9
Press any key to continue

```

٩ - ٨ - الوسطاء الافتراضية

Default Arguments

تتيح الوسطاء الافتراضية تجاهل وسيط أو أكثر عند استدعاء التابع ، وعند وجود وسيط ناقص يزود تصريح التابع قيمة ثابتة لذلك الوسيط المفقود . أن الشكل العام للتصريح عن التابع Box (على سبيل المثال) ، الذي وسطاءه افتراضية هو التالي:

```
Box (int length=1, int width=1, int height=1 ) ;
```


وعندما يستدعى التابع Box() بدون وسطاء، يستخدم التابع القيم الافتراضية للحساب، أو يستدعى التابع بوسيط أو بوسيطين أو ثلاث للحساب.

مثال (9-10)

البرنامج التالي يوضح لنا كيفية استخدام الوسيط الافتراضي ، وفي هذا البرنامج ، تم استخدام تصريح التابع Box_volume لتزويده بثلاثة وسطاء افتراضية ، وتحدد القيمة بعد علامة المساواة ، وحيث إن قيمة هذه الوسطاء هي 1 لكل وسيط ويستدعي البرنامج الرئيسي main التابع Box_volume بطريقتين: أولاً: بدون وسطاء لذا تم حساب box_volume باستخدام القيم الافتراضية للوسطاء ليعيد التابع القيمة 1 كحجم للصندوق .

ثانياً : بوسيط واحد وهو 10 ليعيد التابع القيمة 10 كحجم للصندوق، في هذه الحالة length = 10 ، أو بوسيطين أو ثلاثة وسطاء لحساب حجم الصندوق .

البرنامج:

```
#include <iostream.h>

int Box_volume (int length=1, int width=1, int height=1)
{
    return length*width*height ;
}

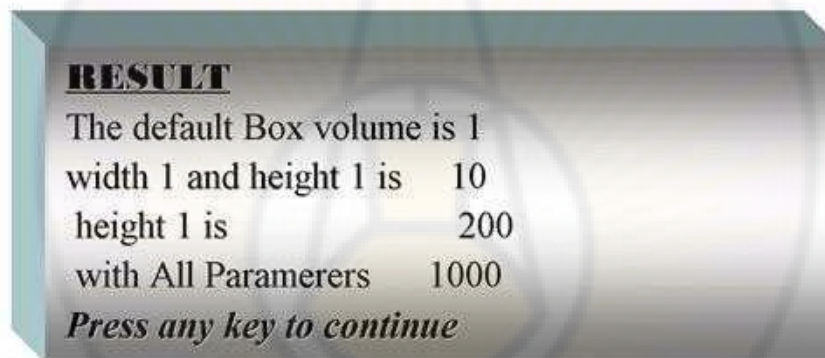
void main( )
{
    cout<<"The default box volume is " ;
    cout<<Box_volume( ) <<endl ;
    cout<<"width 1 and height 1 is " ;
```

```

cout<<Box_volume(10)<<endl;
cout<<" height 1 is " ;
cout<<Box_volume(10,20)<<endl;
cout<<" with All Paramerers " ;
cout<<Box_volume(10,20,5)<<endl;
} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



```

RESULT
The default Box volume is 1
width 1 and height 1 is 10
height 1 is 200
with All Paramerers 1000
Press any key to continue

```

٩ - ٩ - التمرير بالقيمة والتمرير بالعنوان

Pass by Value and Pass by Reference

من المعروف أنه إذا كان لدينا تابع بوسطاء ، فمن أجل استدعاء مثل هكذا تابع يجب علينا أن نمرر قيماً له ، وفي لغة C++ يوجد ثلاثة طرق لهذه الغاية:

١. التمرير بالقيمة (pass-by-value) .
٢. التمرير بالعنوان (pass-by-reference) .
٣. التمرير بالمؤشر (pass-by-pointer) (سيتم التطرق لهذه الطريقة في بحث المؤشرات).

١ - التمرير بالقيمة (pass-by-value)

عند تمرير وسيط بطريقة الاستدعاء بالقيمة يجري إنشاء نسخة عن قيمة الوسيط لتمرر بعدها إلى التابع المستدعى وبالتالي لا تؤثر أية تغيرات تحدث على هذه النسخة على القيمة الأصلية للمتحول (للمتغير) ضمن التابع المنفذ لعملية الاستدعاء.

لنفرض أنه لدينا متحولين صحيحين في برنامج ما، ونريد استدعاء تابع يقوم بتبديل قيمتي المتحولين ، حيث:

```
int x=1;  
int y=2;
```

ولهذه الغاية ، فإن تابع التبديل يمكن أن يأخذ الشكل البرمجي التالي:

```
void swap (int a, int b)  
{  
    int temp;  
    temp =a;  
    a=b;  
    b=temp;  
}
```

يقوم هذا التابع بتبديل القيمتين a و b ، لكن إذا استدعينا هذا التابع كما يلي :

```
swap( x,y);
```

سنجد أن قيمتي x و y لم تتغيرا ، وذلك لأن الوسطاء العادية للتابع يتم تمريرها بالقيمة وينشئ التابع متحولات جديدة كلياً عن a و b .

٢- التمرير بالعنوان (pass-by-reference)

التمرير بالعنوان (المرجع) هو طريقة تمكن التابع (على سبيل المثال التابع swap() من الوصول إلى المتحولات الأصلية x و y والتعامل معها بدلاً من إنشاء متحولات جديدة ، ولإجبار تمرير الوسيط بالعنوان نضيف الحرف & إلى نوع معطيات الوسيط في تعريف التابع وتصريحه. ويبين المثال (9 - 11) كيفية استخدام طريقتي التمرير بالقيمة والتمرير بالعنوان.

مثال (9-11)

أكتب برنامجاً بلغة C++ يقوم بحساب وطباعة مربع أي عددين صحيحين a و b وذلك بطريقتين : التمرير بالقيمة والتمرير بالعنوان.

```
#include <iostream.h>

int sa( int ) ;
int sb( int& ) ;
void main ( ) {
    int a ,b;
    cout<<"Enter any numbers a and b " ;
    cin>>a >>b;
    cout << " sa="<< sa(a) <<endl;
    cout<<a<<endl; // a not change
    cout << " sb="<<sb(b) <<endl;
    cout<<b<<endl; // b is change
} // End main
```

```
int sa ( int x )
{
    return x = x*x ;
}
int sb ( int &y )
{
    return y =y*y;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

sa=25

a = 5

sb=36

b = 36

Press any key to continue

٩ - ١٠ - التوابع العودية

Recursive Functions

من الميزات التي ظهرت في اللغات الحديثة تابع العودية (التكرار والتوليد) ، وتتلخص هذه الميزة بإمكانية أن يستدعي التابع نفسه ، وأن الفائدة التي يجنيها المبرمج من استدعاء التابع نفسه باستعمال تابع العودية هو أن هذا التابع يستخدم لحل بعض المسائل المرتبة ترتيباً متكرراً أصلاً كمسائل ترتيب الأرقام تصاعدياً أو تنازلياً في برنامج quicksort ، كما أن هذا النوع من التوابع يفيد في برمجة بعض مسائل الذكاء الصناعي (Artificial Intelligent) .

وعند استخدام تابع العودية نحتاج أن نستعمل التعليمة الشرطية if لإجبار التابع على العودة إلى مكان انطلاقه بعد استدعاء نفسه في البرنامج، وإن لم نستعمل التعليمة if فإن التابع لا يعود .

مثال (9-12)

اكتب برنامجاً بلغة C++ ، يقوم بما يلي :

- حساب n العاملي (!) باستخدام التابع العودي.
- إيجاد مجموع n عدد صحيح مرتب باستخدام مفهوم العودية .

```
#include<iostream.h>
```

```
int fact( int ) ;
```

```
int sum( int ) ;
```

```
void main( ) {
```

```
int n = 5 ;
```

```
cout<<"Factorial "<< n << " is "
```

```

    << fact(n) << endl;
    cout<<"Summing "<< n <<" is "
    << sum(n) << endl;
} //End main
int fact( int n) {
    if (n == 1 && == 0)
        return 1;
    else
        return n * fact(n-1);
} //End fact
int sum( int n) {
    if (n == 1)
        return 1;
    else
        return n + sum(n-1);
} //End sum

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Factorial 5 is 120

Summing 5 is 15

Press any key to continue

مثال (9-13)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة سلاسل fibonacci (فيبوناتسي).

0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 ,.....

وذلك باستخدام تابع العودية.

توضيح رياضي:

تبدأ سلاسل fibonacci من القيمتين 0 و 1 وتتمتع بالخاصة التالية:

" كل عدد من أعدادها يساوي مجموع العددين السابقين له "

ولذلك يمكن تعريف سلاسل fibonacci بشكل عودي على الشكل التالي:

Fibo(0)=0

Fibo(1)=1

Fibo(n)=fibo(n-1)+fibo(n-2)

البرنامج:

```
/*This program computes the fibonacci numbers  
using recursive*/  
#include <iostream.h>
```

```
unsigned long fibonacci(unsigned long);
```

```
int main( )
```

```
{
```

```
    unsigned long result,number;
```

```
    cout<<"Enter an integer: ";
```

```
    cin>>number;
```

```

for(int i=0;i<=number;i++)
{
    result=fibonacci(i);
    cout<<result<<" , ";
}
cout<<"..."<<endl;
return 0;
}
unsigned long fibonacci(unsigned long n)
{
    unsigned long fib;
    if ((n==0)||(n==1))
        return n;
    else
        fib=fibonacci(n-1)+fibonacci(n-2);
    return fib;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter an integer: 11

0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , ...

Press any key to continue

٩- ١١- توليد الأعداد العشوائية

يوجد في مكتبة C++ تابع عشوائي rand ، يولد أعداداً صحيحة من 0 إلى RAND_MAX ، وهو عبارة عن ثابت مُعرف في الملف الرئيسي `<stdlib.h>` ، يجب أن تكون قيمة RAND_MAX تساوي القيمة 32767 على الأقل ، وهي القيمة الموجبة العظمى لتمثيل قيمة صحيحة بـ 16Bit ، إذا كان التابع rand يقوم فعلياً بتوليد أعداد صحيحة عشوائية فهذا يعني أن الأعداد ما بين 0 و RAND_MAX لها جميعاً نسبة الاحتمال نفسها ، ويتم اختيارها في كل مرة نستدعي هذا التابع ، لكن يمكن لمجال تحول قيم التابع rand أن يكون مختلفاً عن ذلك الذي نحتاجه من أجل تطبيق معين .

فعلى سبيل المثال في برنامج يحاكي عملية رمي قطعة نقود فنحن نحتاج إلى قيمتين فقط هما : 0 من أجل الطرة head ، و 1 من أجل النقش tail ، أما بالنسبة لبرنامج يحاكي عملية رمي حجر النرد ذي ستة وجوه ، فإننا نحتاج إلى ست قيم ضمن المجال من 1 حتى 6 .

وحتى نبين كيفية عمل التابع rand ، لنقم بكتابة برنامج يحاكي عملية رمي نرد ذي ستة وجوه 20 مرة حيث يتم طباعة نتيجة كل عملية من عمليات الرمي نجد في الملف الرئيسي `<stdlib.h>` نموذج التابع rand ، سوف نقوم باستخدام العملية % مع التابع rand على الشكل التالي: `rand() % 6` .

نستخدم `rand() % 6` من أجل توليد أعداد صحيحة ضمن المجال من 0 حتى 5 ، ثم نقوم بإزالة الأعداد المولدة حسب الطريقة السابقة بإضافة 1 إلى كل منها من أجل الحصول على نتائج تقع ضمن المجال من 1 حتى 6 ، ويوضح المثال (9-14) ذلك .

يحدد الاستدعاء `setw( 5 )` طول مجال الطباعة بحيث يشمل مسافة خمسة مواضع (أماكن) للحروف ، وهو معرف في الملف الرئيسي

<iomanip.h> include في مكتبة C++ والذي يحتوي على نماذج التتابع المعيارية الخاصة بالدخل/الخرج . فإذا كانت القيمة المطلوب طباعتها تشكل مسافة أقل من خمسة مواضع عندها تجري طباعة هذه القيمة ضمن المسافة المخصصة لها مع تنسيقها بإزاحتها نحو اليمين ، أما إذا كانت هذه القيمة تشغل أكبر من المسافة المخصص لها عندها تتم طباعة هذه القيمة بما يناسب المواضع المناسبة لها .

مثال (9-14)

اكتب برنامجاً بلغة الـ C++ يقوم بمحاكاة عملية رمي حجر النرد ذي الأوجه الستة، 20 مرة وبحيث يطبع نتيجة كل عملية من عمليات الرمي.

البرنامج:

```
#include<iostream.h>
# include<iomanip.h>
#include<stdlib.h>
int main( )
{
    for(int i=1;i<=20;i++)
    {
        cout<<setw(5)<<1+rand( )%6;
        if(i%5==0)
            cout<<endl;
    }
}
```

```
return 0;
}
```

وعند تنفيذ البرنامج (تنفيذ أول) سنحصل على الخرج التالي:

RESULT				
6	6	5	5	6
5	1	1	5	3
6	6	4	4	2
6	2	3	4	1
Press any key to continue				

وعند تنفيذ البرنامج (تنفيذ ثاني) سنحصل على الخرج التالي:

RESULT				
6	5	5	6	6
3	5	1	1	5
2	4	4	6	6
1	4	3	2	6
Press any key to continue				

باستخدام التابع rand() فإننا نحصل على متتالية الأعداد المولدة عشوائياً نفسها وذلك مهما تم تنفيذ البرنامج الحاوي على هذا التابع . ولكن إذا أردنا أن نحصل على متتاليات مختلفة من الأعداد المولدة عشوائياً فيجب أن نستخدم التابع srand() مع التابع rand() .

يأخذ التابع (srand) قيمة صحيحة من النمط unsigned كوسيط ويعيد تنظيم التابع (rand) بحيث يعطي سلسلة مختلفة من الأعداد العشوائية في كل مرة يتم فيها تنفيذ البرنامج.

مثال (9-15)

كرر المثال السابق من أجل توليد مجموعات من الأعداد العشوائية من 1 إلى 6 باستخدام التابع (srand) .

البرنامج:

```
#include<iostream.h>
#include<iomanip.h>
#include<stdlib.h>
void main() {
    unsigned int seed;
    cout<<"Enter seed: ";
    cin>>seed;
    srand(seed);
    for(int i=1;i<=20;i++)
    {
        cout<<setw(10)<<1+rand()%6;
        if(i%5==0)
            cout<<endl;
    }
} //end main
```

وعند تنفيذ البرنامج (تنفيذ أول) سنحصل على الخرج التالي:

```
RESULT

Enter seed: 58
1      4      1      2      3
2      4      4      1      1
2      2      4      1      2
5      5      3      3      6

Press any key to continue
```

وعند تنفيذ البرنامج (تنفيذ ثاني) سنحصل على الخرج التالي:

```
RESULT

Enter seed: 58
6      4      5      5      5
3      6      4      2      3
3      4      2      4      5
4      3      6      1      3

Press any key to continue
```

من خلال البرنامج السابق نلاحظ أننا استخدمنا تعليمة من سطر واحد من أجل محاكاة عملية رمي حجر نرد سداسي الوجوه، وهي:

$$\text{face} = 1 + \text{rand}() \% 6;$$

هذه التعليمة تسمح بإسناد قيمة صحيحة (عشوائية) إلى المتغير face ، وهذه القيمة تقع ضمن المجال $1 \leq \text{face} \leq 6$ حيث:

✓ " 6 " هو عرض المجال.

✓ " 1 " القيمة المضافة من أجل التحكم ببداية المجال.

بناءً على ذلك ، يمكن أن نعمم التعليمات السابقة ، لتصبح على الشكل التالي:

$$n=a+\text{rand}(\)\%b;$$

حيث:

✓ **b** عامل التكبير .

✓ **a** تعبر عن قيمة الإزاحة (وهي تساوي أيضاً أول قيمة موجودة في مجال

تحويل القيم العشوائية المولدة).



٩ - ١٢ - تمرير المصفوفات كوسطاء للتتابع

عند تمرير مصفوفة أحادية البعد كوسيط لتابع عند استدعائه في التابع () main يجب استخدام اسمها فقط بدون أقواس ، فعلى سبيل المثال ، إذا تم التصريح عن المصفوفة a على الشكل التالي :

```
int a[ 20 ] ;
```

فإن الاستدعاء التالي للتابع sumArray() كما يلي :

```
sumArray( a , 20 ) ;
```

يأخذ التابع اسم المصفوفة وحجمها كوسيطين له ، وعند تمرير مصفوفة إلى تابع فإنه يتم تمرير حجمها أيضاً حتى يتعامل مع العدد المحدد من عناصر المصفوفة الممررة .

إن استقبال المصفوفة المرسله عند استدعاء التابع يجب أن تحتوي قائمة وسطاء التابع عند تعريفه على اسم المصفوفة مع الأقواس [a] ، ويجب كتابة الجزء الرأسي للتابع sumArray() على الشكل التالي :

```
void sumArray( int a[ ] , int size ) ;
```

للدلالة على أن التابع sumArray() يتوقع استقبال مصفوفة من الأعداد الصحيحة حسب الوسيط a وهي تحتوي على size عنصراً . ليس من الضروري وضع حجم المصفوفة بين القوسين عند كتابة الوسيط [20 a] ، وفي حال القيام بذلك فإن المترجم يقوم بتجاهله ، وعلى اعتبار أن المصفوفات يتم تمريرها بالعنوان فإن التابع المُستدعى يقوم باستخدام اسم المصفوفة a من أجل الرجوع إلى المصفوفة الأصلية الموجودة لدى التابع الذي تم الاستدعاء ضمنه المصفوفة ، ويوضح المثال (9 - 16) كيفية تمرير مصفوفة كوسيط لتابع .

مثال (9-16)

```
// compute the sum of the elements of the array
#include <iostream.h>
void sumArray ( int a [ ] , int n ) ;
void main( ) {
    int a[ 10 ] ;
    for (int i= 0; i<10 ; i++)
        cin>> a[i];
    sumArray ( a, 10 ) ;
} //End main
void sumArray ( int a [ ] , int n) {
    int total = 0;
    for (int i= 0; i< n ; i++)
        total = total+ a[i];
    cout <<" total of array element values is " << total << endl;
}
```

وعند تنفيذ البرنامج نحصل على الخرج التالي:

RESULT

```
11 22 33 44 55 66 77 88 99 100
total of array element values is 595
Press any key to continue
```

يوجد العديد من الحالات التي لا نرغب فيها بتغيير قيم عناصر المصفوفة ، وعلى اعتبار أن عملية تمرير مصفوفة تتم بالعنوان فإنه من الصعب التحكم بعملية التغيير . لكن يتوفر في C++ التعليمية `const` لمنع القيام بأي تغيير على قيم عناصر المصفوفة الممررة عند استخدام التعليمية `const` من قبل أي مصفوفة معطاة كوسيط فإن عناصرها تصبح عبارة عن قيم ثابتة ضمن جسم التابع ، ولا يمكن تغيير قيمها وأي محاولة لذلك ستؤدي إلى حدوث خطأ أثناء ترجمة البرنامج، ما يساعد المبرمج على تصحيح برنامجه بشكل لا يسمح بالقيام بعمليات التغيير غير المطلوبة ، ويبين البرنامج في المثال (9-17) كيفية استخدام التعليمية `const` مع التابع `sumArray()` .

مثال (9-17)

```
// compute the sum of the elements of the array
#include <iostream.h>
void sumArray ( int a [ ] , int n ) ;
void main( )
{
    const int a[ 10 ] ;
    for (int i= 0; i<10 ; i++)
        cin>> a[i];
    sumArray ( a, 10 ) ;
} //End main
void sumArray ( int a [ ] , int n ) {
    int total = 0;
```

```

for (int i= 0; i< n ; i++)
    total = total+ a[i];
cout <<" total of array element values is " << total << endl;
}

```

وعند تنفيذ البرنامج نحصل على الخرج التالي:

RESULT

```

11 22 33 44 55 66 77 88 99 100
total of array element values is 595
Press any key to continue

```

عند تمرير مصفوفة ثنائية البعد كوسيط لتابع عند استدعائه في التابع `main()` يجب استخدام اسمها فقط بدون أقواس ، فعلى سبيل المثال ، إذا تم التصريح عن المصفوفة `array` على الشكل التالي :

```
int array [2][3] ;
```

فإن الاستدعاء التالي للتابع `printArray()` كما يلي :

```
printArray(array) ;
```

إن استقبال المصفوفة المرسلة عند استدعاء التابع يجب أن تحتوي قائمة وسطاء التابع عند تعريفه على اسم المصفوفة مع الأقواس `array [2][3]` ، ويجب كتابة الجزء الرأسي للتابع `printArray()` على الشكل التالي :

void printArray(int array [][3]) ;

للدلالة على أن التابع (printArray) يتوقع استقبال مصفوفة ثنائية البعد من الأعداد الصحيحة حسب الوسيط array1 ، ويجب تحديد البعد الثاني للمصفوفة وليس من الضروري تحديد البعد الأول للمصفوفة ، وعند وضع البعد الأول لحجم المصفوفة بين القوسين [3] [2] array ، وفي حال القيام بذلك فإن المترجم يقوم بتجاهله ، وعلى اعتبار أن المصفوفات يتم تمريرها بالعنوان فإن التابع المُستدعى يقوم باستخدام اسم المصفوفة array من أجل الرجوع إلى المصفوفة الأصلية الموجودة لدى التابع الذي تم الاستدعاء المصفوفة ضمنه، ويوضح المثال (9 - 18) كيفية تمرير مصفوفة ثنائية البعد كوسيط لتابع .

مثال (9-18)

يوضح البرنامج التالي كيفية إسناد قيم أولية للمصفوفات متعددة الأبعاد عند التصريح عنها، وكيفية تمرير المصفوفات كوسائط للتوابع وطباعتها ، ثم استخدام التابع (printArray) لطباعة عناصر المصفوفة .

```
// initializing multidimensional arrays
```

```
#include<iostream.h>
```

```
void printArray(int [ ][3]);
```

```
int main( ) {
```

```
//continued
```

```
int array1[2][3] = { {1, 2, 3}, {4, 5, 6} },
```

```
array2[2][3] = {1, 2, 3, 4, 5} , array3[2][3] = { {1, 2},{4}} ;
```

```
cout << "values in array1 by row are : \n";
```



```

printArray( array1 ) ;
//Continued
cout << "values in array2 by row are :\n ";
printArray(array2);
    cout << "values in array3 by row are : \n";
printArray(array3);
} //End main

void printArray(int a[ ][3]) {
    for (int i=0; i<2; i++) {
        for (int j=0; j<3; j++)
            cout << a[ i ] [ j ] << ' ';
            cout << endl;
        }
    } //End function

```

RESULT

values in array1 by row are :

1 2 3

4 5 6

values in array2 by row are :

1 2 3

4 5 0

values in array3 by row are :

1 2 0

4 0 0

Press any key to continue

مثال (9-19)

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- إدخال 12 عدداً صحيحاً ، ويتم تخزينها في مصفوفة ثنائية البعد a[3][4]
- طباعة عناصر المصفوفة a .
- حساب المتوسط الحسابي لعناصر المصفوفة وطباعته .
- استخدم التتابع في البرنامج كما يلي :
- التابع () printArray لطباعة عناصر المصفوفة .
- التابع () printArray1 لحساب مجموع عناصر المصفوفة وحساب المتوسط الحسابي .

```
// multidimensional arrays
#include<iostream.h>
void printArray(int a[ ][4]);
void printArray1(int a[ ][4]) ;
void main( ) {
int a[3][4] ;
cout << " Enter a[ ][ ] values : \n";
for (int i=0; i<3; i++)
for (int j=0; j<4; j++)
cin >> a[ i ][ j ] ;
printArray(a);
printArray1(a);
} //End main
```

```

void printArray ( int a[ 3][4] )
{
    for (int i=0; i<3; i++)
        for (int j=0; j<4; j++)
            cout << a[ i ] [ j ] <<" "<< endl;
}

void printArray1(int a[ 3][4] ) {
    int sum=0;
    for (int i=0; i<3; i++)
        for (int j=0; j<4; j++)
            sum = sum + a[ i ] [ j ] ;
    double avr = sum/12;
    cout << "avr = " << avr << endl;
}

```

RESULT

Enter a[][] values :

11 22 33 44 55 66 77 88 99 12 13 14

11 22 33 44

55 66 77 88

99 12 13 14

Avr = 44

Press any key to continue

مسائل عامة محلولة

١ - اكتب برنامجاً بلغة C++ يقوم بما يلي:

- إدخال أي عدد صحيح a .
- استخدام تابع وليكن اسمه prime يُعيد قيمة ، من أجل حساب عدد قواسم أي عدد صحيح.
- يختبر إذا كان العدد المُدخل هو عددٌ أولي أم لا، ويطبع الرسائل المناسبة لذلك.

البرنامج:

```
/*This program to test if some number is prime or unprime
using function */
#include <iostream.h>
int prime(int x); //Function prototype
int main( )
{
    int a,k;
    cout<<"Enter any integer a ";
    cin>>a;
    k= prime (a);
    if (k>2)
        cout<<"Your number"<<a<<" is unprimed"<<endl;
    else
        cout<<"Your number " <<a<<" is prime\n ";
    return 0;
}
```

```
//Function definition
int prime (int x)
{
    int f=0;
    for (int i=1;i<=x;i++)
    {
        if (x%i==0)
            f+=1;
    }
    return f;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter any integer a 13
Your number 13 is prime
Press any key to continue

٢- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- يُدخل أي عدد صحيح a .
- يستخدم تابع وليكن اسمه factorial ، مهمته حساب العامل (!) لأي عدد صحيح .
- يطبع النتيجة في التابع الرئيسي main .

البرنامج:


```

//This program computes the factorial of any integer
#include <iostream.h>
long factorial(int);
int main( )
{
    int a;
    cout<<"Enter the value of number a : ";
    cin>>a;
    if (a<0)
        cout<<"Error"<<endl;
    else if ((a==0)||(a==1))
        cout<<a<<"!="<<1<<endl;
    else
        cout<<"The factorial of "<<a<<" is
        "<<factorial(a)<<endl;
    return 0;
}
long factorial(int x)
{
    long f=1;
    int i=1;
    while(i<=x)
    {
        f=f*i;
        i++;
    }
    return f;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the value of number a : 10

The factorial of 10 is 3628800

Press any key to continue

٣- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- يُدخل أي عدد صحيح n .
- يستخدم تابع اسمه factiral ، مهمته حساب العامل (!) لأي عدد صحيح .
- يقوم بحساب قيمة المقدار الرياضي التالي:
$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$
- يطبع النتيجة في التابع الرئيسي main

البرنامج:

```
//This program computes the value of //e=2.718281828...
```

```
#include <iostream.h>
```

```
long factiral(int);
```

```
int main( )
```

```
{
```

```
    int i,n;
```

```
    double e=1.0;
```

```

    cout<<"Enter the value of number n: ";
    cin>>n;
    if (n>0)
    {
        for(i=1;i<=n;i++)
            e=e+(1/(double)factual(i));
        cout<<"The value of e is " <<e<<endl;
    }
    else
    {
        cout<<"Enter another value of n \n";
    }
    return 0;
}

long factual(int x)
{
    long f=1;
    int i=1;
    while(i<=x)
    {
        f=f*i;
        i++;
    }
}

```

```

    }
    return f;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT
Enter the value of number n: 10
The value of e is 2.71828
Press any key to continue

=====

- ٤- اكتب برنامجاً بلغة C++ يقوم بما يلي :
- إدخال عناصر المصفوفة $a[3][3]$ وهي مصفوفة ثنائية البعد ذات أعداد حقيقية .
 - طباعة عناصر المصفوفة a .
 - حساب المتوسط الحسابي للأعداد التي تقع فوق القطر الرئيسي للمصفوفة وطباعته .
 - استخدم التوابيع في كتابة البرنامج .

a_{00}	a_{01}	a_{02}
a_{10}	a_{11}	a_{12}
a_{21}	a_{21}	a_{21}

```

// multidimensional arrays
#include<iostream.h>

void printArray(int a[ ][3]);
void printArray1(int a[ ][3]);
void main( ) {
    int a[3][3];
    cout << " Enter a[ ][ ] values : " << endl;
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            cin >> a[ i ][ j ];
    printArray(a);
    printArray1(a);
} //End main

void printArray1(int a[ 3][3])
{
    int sum=0;
    int n = 0;
    for (int i=0; i<3; i++) {
        for (int j=0; j<3; j++)
            if( i<=j ) {
                sum=sum + a[ i ][ j ];
                n = n+1;
            }
    }
}

```



```

    }
    double avr = sum / n ;
    cout << "avr = " << avr ;
    cout << endl;
} //End printArray1
void printArray ( int a[ 3][3] )
{
    for (int i=0; i<3; i++)
        for (int j=0; j<3; j++)
            cout << a[ i ] [ j ] << " " << endl;
} //End printArray

```

RESULT

Enter a[][] values :

11 22 33 55 66 77 99 12 13

11 22 33

55 66 77

99 12 13

Avr = 44

Press any key to continue

مسائل غير عامة

١- اكتب برنامجاً بلغة C++ يقوم بحساب العامل (!) للأعداد من 1 إلى 5.

٢- اكتب برنامجاً بلغة C++ يحتوي على تابعين : التابع الأول اسمه Factorial مهمته حساب العامل (!) لأي عدد صحيح موجب ، أما الثاني اسمه SumFactorial مهمته حساب المقدار التالي :

$$sum = n! + \frac{n!}{x!} - m!$$

التابع main يستدعي كلا من التابعين وطباعة قيمة المقدار Sum (التابع SumFact يستدعي التابع Factorial) .

٣- اكتب برنامجاً بلغة C++ يقوم بما يلي : إدخال قيم المتحولات a,b,x وهي من النمط double ويحتوي البرنامج على تابعين :

❖ التابع الأول اسمه FunctionY مهمته حساب المقدار التالي :

$$y = x^2 + 2x + 10$$

❖ أما التابع الثاني اسمه FunctionZ مهمته حساب المقدار Z والمعرف

كما يلي :

$$Z = (y - a)^2 + (y + b)^2$$

❖ التابع main يستدعي كلا من التابعين لطباعة قيمة Z (التابع FunctionZ يستدعي التابع FunctionY) .

٤- اكتب برنامجاً بلغة C++ يقوم بإدخال أي عدد حقيقي x من قبل المستخدم ويحتوي على ثلاثة توابع لحساب قيمة التابع الرياضي التالي :

$$yy = \begin{cases} 5x^5 + 2x + 5 & \text{if } 1 \leq x < 5 \\ x^6 & \text{if } 5 \leq x < 100 \\ 2x + 1 & \text{otherwise} \end{cases}$$

يستدعي التابع main كلاً من التوابع الثلاثة لطباعة قيمة yy .

=====

٥- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- ❖ إدخال أي عدد صحيح x .
- ❖ يستخدم تابعاً اسمه fun_prime لإيجاد قواسم العدد المدخل x .
- ❖ يختبر البرنامج إذا كان العدد المدخل x أولياً أو غير أولياً وذلك بناءً على خرج التابع fun_prime، وطباعة النتيجة في التابع main .

=====

٦- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- ❖ إدخال أي عددين صحيحين p و k .
- ❖ يستخدم تابعين :
- 📌 التابع الأول اسمه Factorial لحساب قيمة $p!$ (العاملية)
- 📌 التابع الثاني اسمه F_permutation لحساب قيمة

المقدار الرياضي التالي:

$$permutation = \frac{p!}{(p-k)!}$$

❖ التابع main يستدعي كلاً من التابعين لطباعة قيمة Permutation وطباعة النتيجة في التابع main.

=====

٧- اكتب برنامجاً بلغة C++ يقوم بما يلي :

❖ يدخل عدداً حقيقياً ويخزنها في مصفوفة أحادية البعد A[20].

❖ ويحتوي البرنامج على تابعين:

- التابع الأول اسمه (SumAve) مهمته حساب المتوسط الحسابي لعناصر المصفوفة.

- أما التابع الثاني اسمه EvenArry مهمته حساب مجموع الأعداد الزوجية في المصفوفة .

❖ التابع main يستدعي التابعين ويطبع المتوسط ومجموع الأعداد الزوجية .

=====

٨- اكتب برنامجاً بلغة C++ يقوم بما يلي :

❖ يدخل 20 عدداً حقيقياً ويخزنها في مصفوفة أحادية البعد A[20].

❖ ويحتوي البرنامج على تابعين:

- التابع الأول اسمه (MaxArry) مهمته حساب العنصر الأكبر للمصفوفة .

- أما التابع الثاني اسمه OddArry مهمته حساب مجموع الأعداد الفردية في المصفوفة .

❖ يستدعي التابع main التابعين ويطبع العنصر الأكبر ومجموع الأعداد الفردية .

٩- اكتب برنامجاً بلغة C++ يقوم بما يلي :

❖ يدخل 20 عدداً حقيقياً ويخزنها في مصفوفة أحادية البعد A[20].

❖ ويحتوي البرنامج على تابعين:

- التابع الأول اسمه SumAve() مهمته حساب المتوسط الحسابي لعناصر المصفوفة.

- أما التابع الثاني اسمه EvenArry مهمته حساب مجموع الأعداد الزوجية في المصفوفة .

❖ التابع main يستدعي التابعين ويطبع المتوسط ومجموع الأعداد الزوجية

=====

١٠- اكتب برنامجاً بلغة C++ يقوم بما يلي :

❖ يدخل 20 عدداً حقيقياً ويخزنها في مصفوفة أحادية البعد A[20].

❖ ويحتوي البرنامج على تابعين:

- التابع الأول اسمه MaxArry() مهمته حساب العنصر الأكبر للمصفوفة .

- أما التابع الثاني اسمه OddArry مهمته حساب مجموع الأعداد الفردية في المصفوفة .

❖ التابع main يستدعي التابعين ويطبع العنصر الأكبر ومجموع الأعداد الفردية .

=====

١١- أكتب برنامجاً بلغة C++ يقوم بما يلي:

❖ إدخال 10 أعداد صحيحة وتخزينها في مصفوفة أحادية البعد a[10].

❖ يستخدم تابع من النوع void اسمه sort لترتيب الأعداد المدخلة بشكل تصاعدي.

- ❖ التابع main يستدعي التابع sort ويطبع طباعة الأعداد المدخلة بعد عملية الترتيب .

=====

١٢- اكتب برنامجاً بلغة C++ يقوم بما يلي:

- ❖ إدخال 9 أعداد صحيحة وتخزينها في مصفوفة ثنائية البعد x[3][3] .
- ❖ طباعة عناصر المصفوفة x[3][3] على شكل مصفوفة مربعة .
- ❖ يستخدم تابع من النوع void لإيجاد كافة الأعداد الأولية من بين عناصر المصفوفة المدخلة ، ومن ثم طباعة هذه الأعداد وأيضاً طباعة عددها .
- ❖ يستدعي التابع main التابع لطباعة النتائج .

=====



الفصل العاشر

المؤشرات

Pointers





المؤشرات Pointers

١٠-١- مقدمة

إن المؤشرات أحد أهم الإمكانات المتوفرة في لغة C++ ، وتعتبر أحد أصعب الإمكانات المتاحة في لغة C++ من حيث الاستخدام والسيطرة عليها .
ونتخلص أهمية المؤشرات بالأمور التالية :

- ✓ تستعمل المؤشرات لتحسين أداء استدعاء التوابع .
- ✓ تستعمل لاستحداث الذاكرة ودعم الروتينات والتوابع .
- ✓ تعمل على زيادة فعالية التعامل مع المصفوفات متعددة الأبعاد ، كما مر معنا في فصل سابق .

رأينا في فصول سابقة ، أنه يمكن استخدام العناوين من أجل تنفيذ عمليات وصول إلى المتحولات بالعناوين ، وتسمح المؤشرات بمحاكاة عمليات الوصول بالعناوين بالإضافة إلى إمكانية خلق واستخدام بنى معطيات ديناميكية، أي بنى معطيات يمكن أن تنمو ، أو تنقلص أثناء تنفيذ البرنامج ، مثل : المكدرات Stacks ، وصفوف الانتظار queues ، والأشجار trees... إلخ .
يتم في هذا الفصل شرح المفاهيم المرتبطة بالمؤشرات ، كما أنه يساعد على توضيح العلاقة بين المصفوفات والمؤشرات .

١٠-٢- تعريف المؤشرات

المؤشرات هي عبارة عن متحولات تحتوي على عناوين في الذاكرة كقيم مخزنة ضمنها .

يحتوي المتحول عادة على قيمة محددة مخزنة مباشرة ضمنه ، لكن المؤشر يتضمن عنواناً في الذاكرة لمتحول يحتوي على قيمة محددة ، هذا يعني أن :

- المتحول يدل بشكل مباشر على قيمة .
- المؤشر يدل بشكل غير مباشر على قيمة .

مثال (1-10)

ليكن لدينا الشكل (١٠-١) التالي :

a	10
b	20
c	5
p	1000

الشكل (١٠-١)

لدينا مؤشر `int *p` ;

المؤشر `p` أُعد ليُشير إلى `a` ، والرمز `(*)` يدل على أن المتحول هو من نوع مؤشر ، ولذلك يجب وضع `(*)` قبل التصريح عن أي مؤشر، فمثلاً :

```
int x = 5 ;
```

```
int *p ;
```


`p = &a ;`

إن `x` متحول من نوع `int` ، بينما `p` مؤشر من نوع `int` أيضاً ، وبالتالي `P = &a` تعني إسناد قيمة عنوان `x` إلى `p` ، أي أن `p` يشير إلى قيمة `x` .
في أغلب الأحيان فإن المبرمج لا يحتاج إلى معرفة العنوان المحدد لأي متحول معطى لأن المترجم (Compiler) مسؤول عن ذلك ، ولكن إذا رغب المبرمج بمعرفة هذه العناوين فهو يستطيع عن طريق الرمز (&) .
إن المسافة بين العناوين موحدة ، إذا كان طول الكلمة في المعالج 64bit فإن :

❖ النمط `int` يأخذ 4 Byte .

❖ النمط `double` يأخذ 8 Byte

❖ النمط `char` يأخذ 2 Byte .

١٠-٣- التصريح عن المؤشرات

يصرح عن المؤشرات كما يصرح كمثال بقية المتحولات ويجب التصريح عن المؤشرات قبل استخدامها ، فإن الصيغة العامة للتصريح عن المؤشر هي :

`Type * name ;`

يمكن التصريح عن المؤشرات من أنماط مختلفة ، مثلاً :

`int * P ;`

`double * P1 ;`

`char * P2 ;`

يجب وضع الرمز (*) قبل اسم المتحول للدلالة على أنه مؤشر، ويمكن التصريح عن المؤشرات كما يلي :

```
int * P , * d , * C ;
```

يجب إعطاء المؤشرات قيمة ابتدائية ، إما عند التصريح عنها أو باستخدام تعليمات الإسناد ، كما يلي :

```
int a = 10 ;
```

```
int * P ;
```

```
P = & a ;
```

الرمز & هو للدلالة على عنوان المتحول و لقد وجدنا في فصل سابق ، أن الأدوات & و * أدوات أحادية تتعامل مع قيمة واحدة ، حيث :

- ❖ & : علاقة منطقية .
- ❖ * : علاقة ضرب .

يمكن إعطاء القيمة 0 أو NULL أو أي عنوان آخر كقيمة ابتدائية لمؤشر ، حيث يؤشر المؤشر الحاوي على قيمة 0 أو NULL على لا شيء ويدل الرمز NULL على قيمة ثابتة معرفة ضمن الملف الرئيسي iostream وغيره من الملفات الرئيسية ، وتعد القيمة 0 القيمة الصحيحة الوحيدة التي يمكن إسنادها مباشرة إلى متحول من النمط مؤشر .

عندما نكتب $P = \&a$ ، هذا يعني إعطاء عنوان a إلى المؤشر P ، وإذا كانت $a = 10$ وعنوانها 1000 (الشكل (٩-١)) فإن P تحوي

عنوان a أي : $P = 1000$ ، أما إذا كتبنا أن $S = * \& a$ أي :
 $S = 10$ لذلك نرى أن :

$*P = * \& a$ هذا يكافئ أن $*P = a = 10$

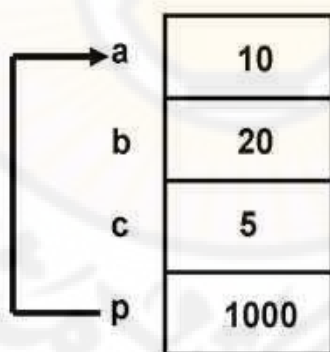
الأداتان تعملان الواحدة معكوس الأخرى ، حيث أن العملية $\&$ (عملية العنوان)
 هي إحدى العمليات الأحادية التي تفيد في تحديد العنوان المعامل المعطى لها .

١٠-٤- الإدخال والإخراج

إن الإدخال والإخراج مشابه لإدخال وإخراج المتحولات ، لنعود إلى
 الشكل (١٠-٢) الذي يوضح المتحولات والمؤشرات عليها ، ونقوم بما يلي :
 □ في الإخراج : نقوم بطباعة المتحول a الذي يساوي إلى 10 كما يلي :

```
cout << *P << endl ;
```

```
cout << a << endl ;
```



الشكل (١٠-٢)

يمكن إعطاء قيمة بشكل مباشر للمتحول المؤشر كما يلي :
 $*P = 5$;

مما يؤدي إلى إسناد القيمة 5 إلى المتحول a .

□ في الإدخال

يمكن استخدام نفس الأسلوب للمتحولات ، مثلاً :

`cin >> *P ;`

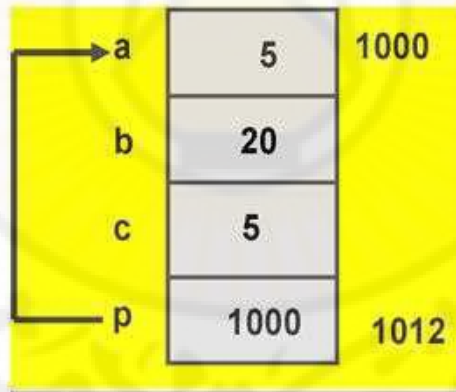
إن عملية الإدخال تغير الشيء الذي يُوْشِر عليه P وليس قيمة P

نفسه ، وإذا أدخلنا القيمة 5 هذا يشبه أن $a = 5$ ، أي أن :

`cin >> a ;`

إذ يجد الحاسوب عنوان موقع الذاكرة الذي يُوْشِر عليه P ويضع القيمة 5

فيه ، وهذه القيمة تكون قيمة a ، كما هو مبين في الشكل (١٠-٣) .



الشكل (١٠-٣)

مثال (2-10)

`int a= 10 ;`

```
int *p ;
```

```
p = &a ;
```

```
a = 5 ;
```

لنفرض أننا أضفنا على البرنامج السطر التالي :

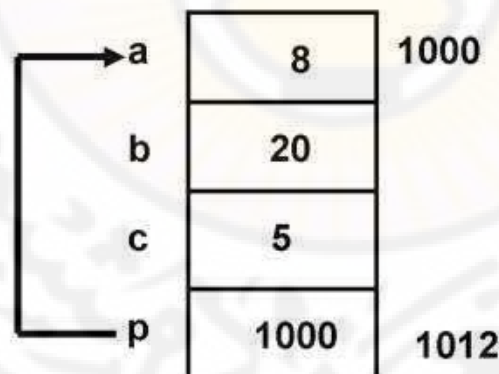
```
*P = * P + 3
```

فإن قيمة a ستتغير من جديد من 5 إلى 8 وهي تكافئ :

```
a = a + 3 ;
```

إذ يجد المعالج عنوان موقع الذاكرة الذي يُوْشر إليه P ويضيف 3 إلى قيمة

ذلك الموقع ، كما هو مبين في الشكل (١٠-٤) .



الشكل (١٠-٤)

يوضح الجدول (١٠-١) مقارنة بين العمليات على المؤشر والعمليات على المتحولات .

الأثر نفسه لدى التعليمة التالية	عندما يشير P إلى a فإن
a = 33 ;	* P = 33 ;
a = a + 2 ;	* P = * P + 2 ;
cout << a ;	cout << * P ;
cin >> a ;	cin >> * P ;

الجدول (١٠-١) - يبين مقارنة بين المؤشرات والمتحولات

إذا كان استخدام * P مكافئ لاستخدام a لماذا هذا الاستخدام ؟
لأن ذلك يحقق التمرير بالعنوان (Pass by Reference) حيث يمكن أن يغير
التابع قيمة أحد وسائطه ، أي يستطيع التابع أن يستخدم * P لتعديل a .

مثال (3-10)

البرنامج التالي يوضح كيفية طباعة قيم بعض المتحولات المدخلة وعناوينها ،
نعرف المتحولات : a و b و s و d1 ، أما المؤشرات هي : *d ، *c ، *p
#include <iostream.h>

```
void main ( ) {
    int a = 5, b = 6 , s ;
    int *p , *c, *d , d1;
    p=&a; c=&b;
    *p = *p + 3 ;
```

```

s = *p ;
cin>> d1;
d=&d1;
cout<< "a=" << a<<endl;
cout<< "p=" << p<<endl;
cout<< "&a=" << &a<<endl;
cout<< "*p=" << *p<<endl;
cout<< "s=" << s<<endl;
cout<< "a=" << *&a<<endl;
cout<< "*d=" << *d<<endl;

} // End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

99

a=8

p=0x0012FF7C

&a=0x0012FF7C

*p =8

s =8

a =8

*d =99

Press any key to continue

١٠-٥- استدعاء التتابع Function Calling

يوجد في لغة C++ ثلاثة طرق لتمرير الوسيط إلى التتابع :

١. التمرير بالقيمة Call - by - Value .
٢. التمرير بالمرجع (العنوان) Call - by - Reference .
٣. التمرير بالعنوان باستخدام المؤشرات Call - by - Reference with Pointers .

عند تمرير وسيط بطريقة الاستدعاء بالقيمة يجري إنشاء نسخة عن قيمة الوسيط لتمرر بعدها إلى التابع المستدعي وبالتالي لا تؤثر أية تغييرات تحدث على هذه النسخة على القيمة الأصلية للمتحول ضمن التابع المنفذ لعملية الاستدعاء (كما وجدنا ذلك في فصل سابق) .

عند تمرير وسيط بطريقة الاستدعاء بالعنوان بأن نضع إشارة & بعد نمط الوسيط في نموذج التابع وكذلك عند كتابة الجزء الرأسي من تعريف التابع، حيث يتم نقل عنوان (address) متعلق بقيمة الوسيط المنقول إلى التابع ويستخدم هذا العنوان داخل التابع لتزويده بالقيمة المطلوبة لذا فإن قيمة الوسيط تتغير داخل التابع، أما عند تمرير الوسيط بواسطة المؤشرات فإننا نضع إشارة * بعد نمط الوسيط في نموذج التابع وكذلك عند كتابة الجزء الرأسي من تعريف التابع، وهو يتعامل مع القيمة الأصلية للمتحول كما في طريقة تمرير الوسيط بالعنوان .

عند تمرير وسيط للتابع بطريقة الاستدعاء بالمرجع (العنوان) أو عن طريق المؤشرات فإننا نتعامل مع القيمة الأصلية للمتحول وأي تغيير يحدث يتم على القيمة الأصلية للمتحول وليس على نسخة من هذا المتحول .

مثال (4-10)

اكتب برنامجاً لحساب مربع عدد صحيح وطباعته ، بطريقة الاستدعاء بالقيمة ثم بالعنوان ، وأخيراً التمرير بالعنوان باستخدام المؤشرات .

```
#include <iostream.h>
int sa( int ) ;
int sb( int& ) ;
int sc( int* ) ;
void main ( )
{
    int a=5 , b= 6 ,c=7;
    cout << " sa=" << sa(a)<<endl;
    cout<<a<<endl;// a not change
    cout << " sb=" <<sb(b)<<endl;
    cout<<b<<endl;// b not change
    cout<<" sc = " << sc( &c) << endl ;
    cout<< c <<endl;// c not change
    sa(a) ; sb(b) ; sc(&c) ;
} //End main
```

```

int sa ( int x)
{
    return x*x;
}
int sb ( int&y)
{
    return y*y;
}
int sc ( int *z)
{
    return *z * *z;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



```

RESULT
sa=25
sb=36
sc= 49
a = 5
b = 6
c = 7
Press any key to continue

```

تساعد طريقة تمرير الوطاء بالعنوان على إعطاء التابع المسند إليه إمكانية الوصول المباشر إلى معطيات التابع المستدعى وتغيرها إذا رغب ذلك .

□ إذا عرفنا التتابع السابقة كما يلي :

```
int sa ( int x)
{
    return x=x*x ;
}

int sb ( int&y) {
    return y=y*y;
}

int sc ( int *z)
{
    return *z= *z * *z ;
}
```

□ نحصل على النتيجة التالية :

RESULT

sa=25

sb=36

sc= 49

a = 5

b = 36

c = 49

Press any key to continue

هنا تم تغيير قيم المتحولات b و c والحفاظ على قيمة a ، أي تغيير قيمة المعطيات الأصلية للمتحولات في حالة الاستدعاء بالمرجع والمؤشر .
□ يمكن استخدام العنونة للدلالة على متحولات أخرى داخل التابع :

```
int count = 10 ;  
int &y = count ;  
y++ ; // زيادة بمقدار واحد
```

□ جميع العمليات المطبقة على y هي عمليات تطبق على المتحول الأصلي نفسه count ، زيادة أو نقصان بمقدار واحد ... الخ .

مثال (5-10)

المثال التالي يوضح أن التغيير يتم على المتحول الأصلي وليس على النسخة المولدة .

```
#include <iostream.h>  
  
void main ( ) {  
  
    int count=10;  
  
    int &y=count;  
  
    cout<< "y=" << y<<endl;  
  
    cout<< "count=" << count<<endl;  
  
    y++ ;
```

```

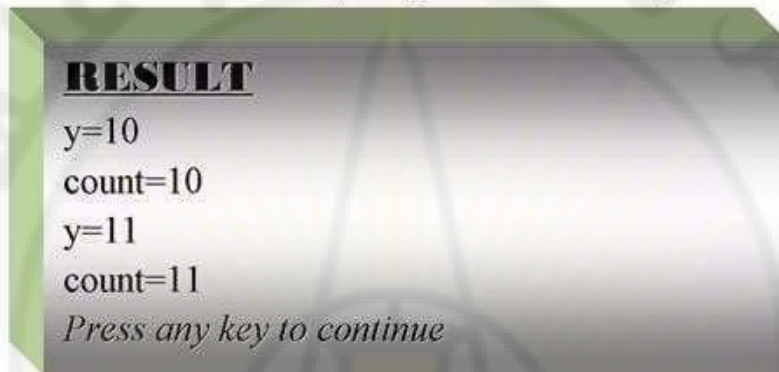
cout<< "y=" << y<<endl;

cout<< "count=" << count<<endl;

}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



ملاحظة (1) : يمكن للتتابع أن تعيد قيمة مرجعية (عنوان) عندها يجب التصريح عن المتحول أنه من النمط static داخل التابع .

ملاحظة (2) : عند عدم إعطاء قيم ابتدائية لمتحول مرجعي عند التصريح عنه سوف لا يعمل البرنامج ويعطي خطأ .

```

int count = 10 ;

int &y ;

```

لا توجد قيمة ابتدائية هنا ، لذلك سوف يعلن البرنامج عن خطأ .

مثال (6-10)

المثال التالي يوضح عملية الاستدعاء بالقيمة والعنوان والمؤشرات ، وهو يشبه

المثال (5-10) ، ويقوم البرنامج التالي بحساب مكعب ثلاثة أعداد بطرق الاستدعاء الثلاثة .

/ This program calculates the cube of integer number using Pointers */*

```
#include <iostream.h>
void cubeV(int);
void cubeR(int&);
void cubeP(int*);
int main( ) {
    int a,b,c;
    cout<<"Enter the number ";
    cin>>a;
    cin>>b;
    cin>>c;
    cubeV(a);
    cubeR(b);
    cubeP(&c);
    cout<<"The cube of a is "<<a<<endl;
    cout<<"The cube of b is "<<b<<endl;
    cout<<"The cube of c is "<<c<<endl;
    return 0;
} //End main
```

```

void cubeP(int* x) {
    *x = *x * *x * *x;
}
void cubeR(int& x)
{
    x=x*x*x;
}

void cubeV(int x)
{
    x=x*x*x;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT
Enter the number 6 7 8
The cube of a is 6
The cube of b is 343
The cube of c is 512
Press any key to continue

مثال (7-10)

اكتب برنامجاً بلغة C++ لحساب وطباعة حجم مكعب طول ضلعه L وذلك باستخدام الاستدعاء بواسطة المؤشرات .

/ Write a program that cubes integer , passe variable by reference using pointers */*


```

#include <iostream.h>

void cube( int* );

void main ( ) {
    int a=5 , b=6 ;
    cout<< " a="<<a<<endl;
    cout<< " b="<<b<<endl;
    cube(&a);//pass address of a
    cube(&b);//pass address of b
    cout << " cube a="<< a<<endl;
    cout << " cube b="<<b<<endl;
} //End main

void cube ( int* p)
{
    *p=*p * *p * *p;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

a=5

b=6

cube a=125

cube b=216

Press any key to continue

١٠-٦- تابع التبديل (swap)

نفترض لدينا متحولين a , b ونريد التبديل بين القيمتين ، مثلاً إذا كانت $a = 5$, $b = 6$ بعد التبديل تصبح $a = 6$, $b = 5$ ، من السهل فعل ذلك باستخدام متحول ثالث $temp$ كوسيط وظيفته تخزين قيمة مؤقتة به ، كما يلي :

$temp = a$;

$a = b$;

$b = temp$;

إذا لم تمرر الوسائط بالعنوان بواسطة المؤشرات فتهمل التغيرات على المتحولات، وتستخدم المؤشرات لكي يتم تمرير المعطيات بالعنوان .

مثال (8-10)

البرنامج التالي يوضح كيفية إدخال عددين صحيحين ، ثم إجراء التبادل بينهما من خلال استدعاء تابع التبديل ، ثم طباعة العددين قبل التعديل وبعده .

```
#include <iostream.h>
```

```
void swap(int *p1, int *p2);
```

```
void main ( ) {
```

```
    int a=5 , b=6 ;
```

```
    cout << " a=" << a << endl;
```

```
    cout << " b=" << b << endl;
```

```

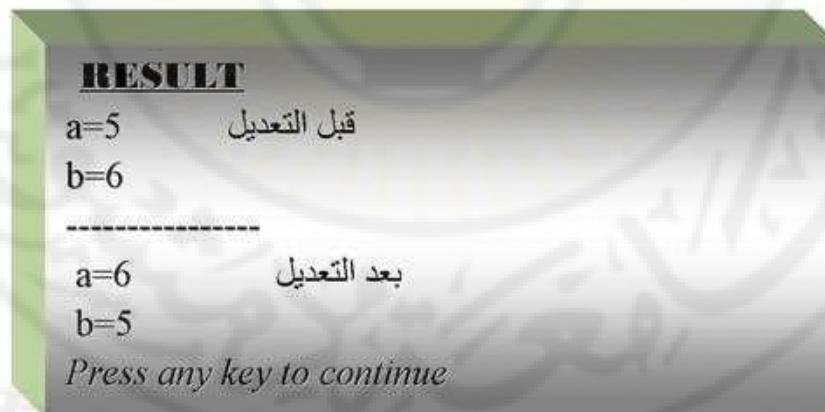
swap(&a, &b);

cout << " a=" << a << endl;
cout << " b=" << b << endl;
}

void swap(int *p1, int *p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



```

RESULT
a=5      قبل التعديل
b=6
-----
a=6      بعد التعديل
b=5
Press any key to continue

```

□ حيث *P1 و *P2 عدنان صحيحان يمكن استخدامهما كما نستخدم أي متحولين صحيحين قبل التعديل .

١٠-٧- الفرز الفقاعي Bubble Sort

غالباً ما تكون قيم المتحولات غير مرتبة بالشكل الذي يمكن للحاسوب أن يتعامل معها ، وغالباً ما يقوم المرء بترتيب درجات الطلاب بشكل تصاعدي أو ترتيب الأسماء بشكل هجائي أو تصنيف الناس حسب أعمارهم أو أطوالهم من أجل أن يؤدي الحاسوب دوره بكفاءة عالية .

مثال (10-10)

يقوم البرنامج التالي بترتيب مجموعة من الأعداد الصحيحة ترتيباً تصاعدياً وذلك باستخدام مصفوفة ويتم إدخال عناصرها من لوحة المفاتيح ، ونستخدم في مثالنا تابع التبديل السابق من أجل ترتيب الأعداد تصاعدياً وبذلك يتكون البرنامج من تابع أساسي يحوي المصفوفة وهذا التابع يستدعي بدوره تابع التبديل ويتم تمرير المتحولات بواسطة العنوان .

```
#include <iostream.h>
void sort(int* , int );
void swap(int * , int * );
void main ( ) {
    const int d=10;
    int a[d] ;
    for (int i = 0; i < d; i++) {
        cout << "Enter array element #" << i << ": ";
        cin >> a[i];
    }
```

```

    sort( a , d);
    cout << "Here are all the array elements, sorted:"
          << endl;
    for (i = 0; i < d; i++)
        cout << a[i] << " ";
}

```

// Sort array function: sort array named a, having n elements.

```

void sort (int*a , int n)
{
    for(int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
            if (a[j] < a[i])
                swap(&a[i], &a[j]);
    }
}

```

// Swap function.

```

void swap(int *p1, int *p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}

```


وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```
Enter array element # 0: 8
Enter array element # 1: 9
Enter array element # 2: 2
Enter array element # 3: 77
Enter array element # 4: 5
Enter array element # 5: 11
Enter array element # 6: 6
Enter array element # 7: 9
Enter array element # 8: 2
Enter array element # 9: 33
```

Here are all the array elements, sorted:

```
2 2 5 6 8 9 9 11 33 77
```

Press any key to continue

١٠-٨- المؤشرات والمصفوفات

Pointers & Arrays

يوجد في لغة C++ علاقة وثيقة بين المؤشرات والمصفوفات يمكن استخدام أحدهما بدل الآخر ، وكل عملية يمكن تنفيذها على المصفوفات ، يمكن تنفيذها أيضا باستخدام المؤشرات ، وهذا الأمر يعطي سرعة في تنفيذ البرامج ولكن البرامج تصبح أكثر تعقيداً .

يمكن استخدام المؤشرات للقيام بأي عملية تستدعي استخدام مصفوفة مع تحديد أدلة عناصرها المعينة بذلك ، لنفرض لدينا مصفوفة $a[5]$ ، ومؤشر P يمكن أن نضع في P عنوان بداية المصفوفة كما هو موضح في الشكل (١٠-٥) .

$a[0] \quad a[1] \quad a[2] \quad a[3] \quad a[4]$

الشكل (١٠-٥)

نعلم أن التصريح التالي :

$\text{int } a[5];$

يشير إلى أن a هي مصفوفة مكونة من خمسة عناصر من النمط int كما يلي :

$a[0], a[1], a[2], a[3], a[4]$

لنفرض أن p هو مؤشر يشير إلى عدد صحيح فنصرح عنه كما يلي :

$\text{int } *p;$

بما أن المصفوفة عبارة عن مؤشر على أول عنصر من عناصرها فإن :

$P = a;$

وبالتالي نحصل على التكافؤ التالي :

$p = a \quad P = \&a[0];$ هذا يكافئ

$\text{cout} \ll a[3], \text{cout} \ll *(p + 3)$ يكافئ

يمكن الوصول إلى العنصر $a[3]$ باستخدام المؤشرات

$*(P + 3) \equiv a[3]$ وهذا يكافئ $*(P + 3);$

كما هو موضح في الشكل (١٠-٦) .

a[0]	a[1]	a[2]	a[3]	a[4]
*P	*(P+1)	*(P+2)	*(P+3)	*(P+4)

الشكل (١٠-٦)

إن وضع القوسين في الشكل (١٠-٦) لعناصر المصفوفة هو أن (*) لها أولوية على عملية الجمع (+) ، لأن $*P + 3$ تعني أضف القيمة 3 إلى محتوى العنصر $a[0]$.

إن المصفوفة مثل المؤشر ، أي :

$$*(a + 3) \text{ هذا يكافئ } a[3]$$

إن عنوان $a[3]$ & يكافئ $(P + 3)$ ، وأن $P[1]$ يدل على العنصر a

$[1]$ من المصفوفة ، لأن المصفوفة عبارة عن مؤشر ثابت يدل على بداية

المصفوفة ، لذلك يعتبر الشكل التالي غير مقبول :

$$a += 3 ;$$

مثال (10-10)

هدف البرنامج التالي هو توضيح فكرة التكافؤ بين المصفوفات والمؤشرات .

لنفترض لدينا البرنامج التالي يطبع عناصر المصفوفة كما يلي :

- بدون استخدام المؤشرات .
- باستخدام المؤشرات .
- باستخدام المصفوفة $a[]$ كمؤشر .

- باستخدام المؤشر P كمصفوفة P [] .

البرنامج :

```
#include <iostream.h>
void main ( )
{
    int a [ ] = { 10,20,30,40};
    int *p =a;
        // output 1
    for(int i=0 ; i<4 ; i++)
    {
        cout<< " a [ " <<i<< " ] = "<<a[ i ]<<endl;
    }
    cout<<endl;

        // output 2
    for ( int j=0 ; j<4; j++ )
    {
        cout<< " *( a+ " <<j<< " ) = "<<*( a+j )<<endl;
    }
    cout<<endl;
```

```

// output 3
for(int k=0;k<4;k++)
{
    cout<< " P[ " <<k<< " ] = "<<p[ k ]<<endl;
}
cout<<endl;

// output 4
for(int m=0;m<4;m++)
{
    cout<< " *( p + " <<m<< " ) = "<<*(p+m)<<endl;
}

} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT	RESULT	RESULT	RESULT
a[0] = 10	*(a+ 0) = 10	P[0] = 10	*(p + 0) = 10
a[1] = 20	*(a+ 1) = 20	P[1] = 20	*(p + 1) = 20
a[2] = 30	*(a+ 2) = 30	P[2] = 30	*(p + 2) = 30
a[3] = 40	*(a+ 3) = 40	P[3] = 40	*(p + 3) = 40
Press any key to continue			

مثال (11-10)

يبين البرنامج التالي فكرة التكافؤ بين المصفوفات والمؤشرات ولكن باستخدام التتابع ، إن التابع () copy1 و () copy2 يقومان بنسخ سلسلة حروف بطريقتين مختلفتين .

البرنامج :

```
#include <iostream.h>

void copy1(char *, const char * );
void copy2(char *, const char * );

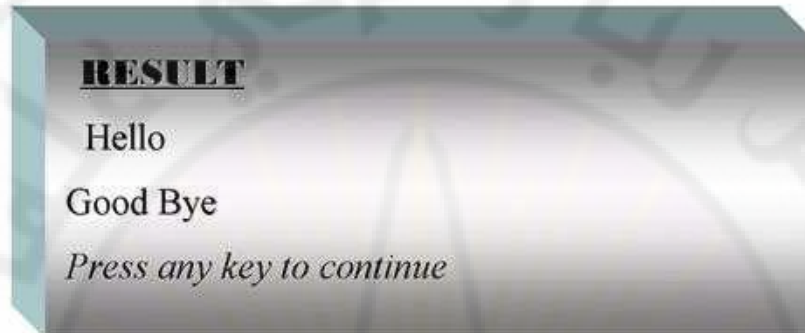
void main( ) {
    char str1[10],str2[ ]=" Hello " ;
    char str3[10],str4[ ]=" Good Bye " ;
    copy1( str1,str2);
    cout<<str1<<endl;
    copy2( str3,str4);
    cout<<str4<<endl;
}

void copy1( char *s1,const char *s2)
{
    for(int i=0; ( s1[i]=s2[i])!='\0';i++) ;
}

void copy2( char *s1,const char *s2)
{
```

```
for( ; ( *s1= *s2 )!='\0'; s1++ , s2++) ;
}
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:



عندما نصرح عن تابع أحد متحولاته هو مصفوفة ، هذا يعني أننا نصرح عن متحول من نمط مؤشر . إن المصفوفة لا تمرر بالقيمة كوسيط لتابع ما ، حيث أن التابع لن يقوم أبداً بخلق نسخة عن المصفوفة الممررة بشكل تلقائي بل سيتعامل مع المصفوفة نفسها ، وبذلك فعندما نقوم بتعديل المصفوفة داخل التابع سيظهر لنا أننا نعدل المصفوفة ذاتها خارج التابع ، كما هو موضح في المثال (10 - 12) .

مثال (10-12)

إن التابع () print في المثال التالي يستعمل الأقواس المربعة [] من أجل تمرير المصفوفة كوسيط له ، مع أن صيغة الأقواس المربعة وصيغة المؤشر تشبه تماماً عملية تمرير المصفوفات كوسطاء ، إلا أن استخدام صيغة الأقواس المربعة أوضح للقارئ .

إن تمرير الوسيط n سوف يتم في كل مرة ، وأن تمرير عنوان المصفوفة لا يكون كافياً فيجب تمرير حجم المصفوفة .

البرنامج :

```
#include <iostream.h>

void func1(int a[ ],const int n )
{
    for(int i=0;i<n;i++)
        a[i] = i * i - i ;
}

void func2(int *a ,const int n )
{
    for(int i=0; i <n; i++)
        a[i] = i * i + i;
}

void print ( int a[ ],char name ,const int n)
{
    for ( int i=0;i<n;i++)
        cout<<name<<"["<<i<<"]=" <<a[i]<<endl;
}

void main( ) {
    int a[5]={10,20,30,40,50} ;
```

```

int b[5]={100,200,300,400,500} ;
print(a,'a',5) ;
cout << endl ;
print(b,'b',5) ;
cout << endl ;
func1(a,5) ;
func2(a,5) ;
cout << endl ;
print(a,'a',5) ;
cout << endl ;
print(b,'b',5) ;
cout << endl ;
} // End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

<u>Result</u>	<u>Result</u>	<u>Result</u>	<u>Result</u>
a[0]=10	b[0]=100	a[0]=0	b[0]=0
a[1]=20	b[1]=200	a[1]=0	b[1]=2
a[2]=30	b[2]=300	a[2]=2	b[2]=6
a[3]=40	b[3]=400	a[3]=6	b[3]=12
a[4]=50	b[4]=500	a[4]=12	b[4]=20

Press any key to continue

مثال (10-13)

اكتب برنامجاً بلغة C++ يقوم بحساب وطباعة المتوسط الحسابي لعشرة أعداد حقيقية باستخدام تابع وسطاء عبارة عن مؤشرات من النوع الحقيقي .

/ This program computes the average of ten numbers using
function with pointers */*

```
#include <iostream.h>
```

```
void av(float*,float*);
```

```
void main()
```

```
{
```

```
    float a[10];
```

```
    float aver;
```

```
    float *b=&aver;
```

```
    for (int i=0;i<10;i++) {
```

```
        cout<<" a[ " <<i<<" ] " ;
```

```
        cin>>a[i];
```

```
    }
```

```
    cout<<"b="<<b<<endl;
```

```
    av(a,b);
```

```
    cout<<"aver="<<aver<<endl;
```

```
    cout<<"The average of entered numbers is " <<aver<<endl;
```

```
}//End main
```



```

void av(float *x,float *n) {
    int i;
    float sum=0.0;
    for (i=0;i<10;i++)
        sum+=x[i];
    *n=sum/10;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```

a[ 0 ]=1.1
a[ 1 ]=2.2
a[ 2 ]=3.3
a[ 3 ]=4.4
a[ 4 ]=5.5
a[ 5 ]=6.6
a[ 6 ]=7.7
a[ 7 ]=8.8
a[ 8 ]=9.9
a[ 9 ]=10.10
b=0x0012FF54
aver=5.96

```

The average of entered numbers is 5.96

Press any key to continue

١٠-٩- مصفوفة المؤشرات

Array Of Pointers

يمكن تنظيم أي مجموعة من المؤشرات ضمن إطار مصفوفة ، كما يلي :

- `int a [ 5 ]` مصفوفة عادية .
- `int * a [ 5 ]` مصفوفة المؤشرات .

مثلاً :

```
a [ 4 ] = & x ;  
Cout << *a [ 4 ] ;
```

ويمكن أن نصرح عن المصفوفة كمصفوفة محارف كما يلي :

```
char * a [ 5 ] = { " course1 " , " course2 " , " course3 " ,  
" course4 " , ' course5 " } ;
```

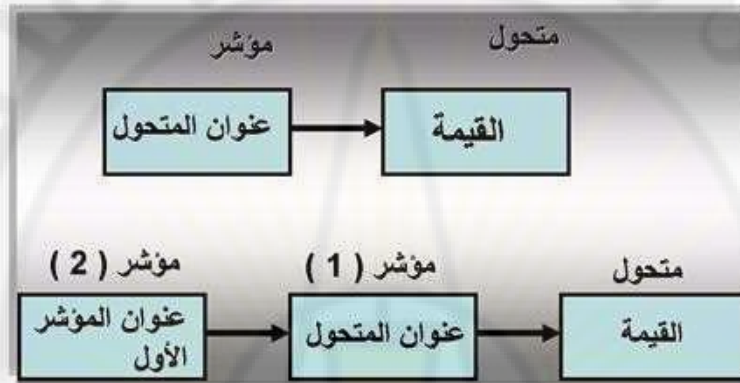
ويمكن أن تكون مصفوفة المؤشرات كوسيط في تابع ، كما يلي :

```
void func ( int , * a [ ] )  
{  
    for ( int i = 0 ; i < 5 ; i ++ )  
        cout << *a [ i ] << endl ;  
}
```

١٠-١٠- مؤشرات المؤشرات

Pointers To Pointers

إذا اعتبرنا أن المؤشر متحول يمكن أن نشير إليه بمؤشر آخر ،
والشكل (٧-١٠) التالي يوضح هذه العملية :



الشكل (٧-١٠) - مؤشرات المؤشرات

ونلاحظ من الشكل (٧-١٠) ، أن قيمة المؤشر الأول هي عنوان المتحول ، وأن قيمة المؤشر الثاني هي عنوان المؤشر الأول ، وقيمة المؤشر الثالث هي عنوان المؤشر الثاني ، وهكذا ، ويدل التصريح التالي على مؤشر المؤشر :

`int ** x ;`

(* * x) يدل على مؤشر لمؤشر يشير إلى متحول من النمط int ، ويوضح المثال (10 - 14) عملية مؤشر المؤشر .

مثال (10-14)

البرنامج التالي يوضح فكرة مؤشر المؤشر .

البرنامج :

```
#include <iostream.h>

void main( ) {
    int d , *a , **b ;
    d=20 ;
    a= &d ;
    b=&a ;
    cout << " d = " <<d<< endl ;
    cout << " *a = " <<*a<< endl ;
    cout << " **b = " <<**b<< endl ;
} //End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

d = 20

*a = 20

**b = 20

Press any key to continue

١٠-١١ - مؤشرات التتابع Pointers of Functions

يتضمن المؤشر على تابع عنوان ذلك التابع في الذاكرة ، فعندما يقوم المترجم بترجمة التابع تتحول شيفرة التابع من شيفرة مصدرية Source إلى شيفرة هدف Target ، وتتأسس نقطة دخول للتابع ، ويتم استدعاء التابع لبرنامج ينفذ حالياً باستعمال نقطة الدخول ، ولذلك يمكن استعمال المؤشر لاستدعاء التابع لأنه يحتوي على عنوان نقطة الدخول ، ويحصل على عنوان التابع من اسمه بدون أقواسه أو متحولاته المعتمدة ، كما كنا نستعمل اسم المصفوفة بدون أقواس أو أسماء عداداتها.

مثال (10-15)

اكتب برنامجاً بلغة C++ يقوم بما يلي :

- ❖ إدخال عدد صحيح a .
 - ❖ حساب وطباعة مربع العدد المدخل a باستخدام مؤشر تابع .
- البرنامج :

```
//This program uses pointer to function
```

```
#include <iostream.h>
```

```
int sq(int);
```

```
int (*s)(int) ; //Pointer to function
```

```
void main() {
```

```
    int a;
```



```

cout<<"Enter any number a:";
cin>>a;
s=sq;
cout<<"The square of "<<a<<" is "<<s(a)<<endl;
} //End main

int sq(int x)
{
    return x * x;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

Result

Enter any number a :10

The square of 10 is 100

Press any key to continue

التصريح `int (*s)(int) ;` ، يعني أن S هو مؤشر على تابع بسيط واحد من النوع `int` ويعيد قيمة من النوع `int` .

مثال (10-16)

- اكتب برنامجاً بلغة C++ يقوم بما يلي :
- إدخال عددين صحيحين a , b .
 - حساب وطباعة مجموع العددين a , b باستخدام مؤشر إلى تابع .
- البرنامج :

```
//This program uses pointer to function
#include <iostream.h>
int sq(int);
int cube(int);
int sum( int(*) (int),int );
int (*s)(int,int) ; //Pointer to function
int main( )
{
    int a;
    cout<<"Enter any number a:";
    cin>>a;
    cout<<"The square of "<<a<<" is "<<sum(sq,a)<<endl;
    cout<<endl;
    cout<<"The cube of "<<a<<" is "<<sum(cube,a)<<endl;
    return 0;
} //End main
```

```

int sum(int(*pff)(int k),int n)
{
    int s;
    s=(*pff)(n);
    return s;
}
int sq(int k)
{
    return k*k;
}
int cube(int k)
{
    return k*k*k;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter any number a:10

The square of 10 is 100

The cube of 10 is 1000

Press any key to continue

مثال (10-17)

هذا المثال يوضح طريقة البحث الثنائي باستخدام المؤشرات
. The binary search

//The binary search function

```
int bin_sea(long *p,long key,int n) {  
    long high,low,mid;  
    low=0;  
    high=n-1;  
    while(low<=high)  
    {  
        mid=(low+high) / 2 ;  
        cout<<"mid="<<mid<<" "<<"low="<<low  
        <<" " <<"high="<<high<<" "  
        <<"p["<<mid<<"]="<<p[mid]<<endl;  
        if( key == p[mid] ) {  
            return mid ;  
        }  
        else if(key<p[mid])  
            high = mid-1;  
        else  
            low=mid+1;  
    }
```

```

    }
    return -1;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter any number between 0 and 20: 6 11

mid=4 low=0 high=9 p[4]=8

mid=1 low=0 high=3 p[1]=2

mid=2 low=2 high=3 p[2]=4

mid=3 low=3 high=3 p[3]=6

Your element is found and his index is 3

Press any key to continue

مثال (18-10)

هذا المثال يوضح طريقة البحث الخطي باستخدام المؤشرات

.The linear search

//The linear search using pointers

```
#include<iostream.h>
```

```
int lin_sea(float*,int,int);
```



```

int main() {
    float x[10],s;
    int i , key;
    for (i=0;i<10;i++) {
        cout<<"Enter the element "<<i<<" of numbers :";
        cin >> x[i];
    }

    cout<<"Enter any number : ";
    cin>>key;
    s=lin_sea(x,key,10);
    if (s != -1)
        cout<<"Your element is found and his index is "
            << s << endl;
    else
        cout<<"Your element is not found and his index is "
            << s <<endl;
    return 0;
}

```

//The linear search function

```

int lin_sea(float*p,int key,int n) {
    int i , mid , count = 0 ;
    for (i=0;i<n;i++) {
        count+=1;
    }
}

```

```

        if(key == p[i] ) {
            cout<<"count="<<count<<endl;
            return i;
        }
    }
    cout<<"count="<<count<<endl;
    return -1;
}

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```

Enter the element 0 of numbers :2
Enter the element 1 of numbers :4
Enter the element 2 of numbers :7
Enter the element 3 of numbers :11
Enter the element 4 of numbers :22
Enter the element 5 of numbers :9
Enter the element 6 of numbers :44
Enter the element 7 of numbers :5
Enter the element 8 of numbers :33
Enter the element 9 of numbers :8
Enter any number : 44
count=7
Your element is found and his index is 6
Press any key to continue

```

الفصل الحادي عشر

السلاسل

Strings





السلاسل Strings

١١-١- مقدمة

تشكل المحارف البنية الأساسية المؤلفة لكتل نصوص البرامج المكتوبة بلغة C++ ، ويتألف كل برنامج من سلسلة من المحارف التي يشكل تجمعها مع بعضها البعض نصاً ذو معنى ويمكن تفسيره بواسطة الحاسوب على أنه سلسلة من التعليمات اللازمة لإتمام مهمة معينة .

تتألف سلاسل المحارف من مجموعة من المحارف التي تُعامل على أنها وحدة واحدة ، وهي قد تتضمن حروف الأبجدية الإنكليزية الكبيرة والصغيرة (Aa,Bb,...) والأرقام العربية (1,2,3,...) وبعضاً من الحروف الخاصة مثل (; , \$, + , ...) .

تُعد المحارف سلاسل محرفية في لغة C++ ، وهي عبارة عن مصفوفات للمحارف المنتهية بالمحرف الصفري ('\0') ، ويتم الوصول إليها بواسطة مؤشر على أول حرف من حروفها . إن سلسلة المحارف في لغة C++ هي عبارة عن مؤشر على أول محرف من محارفها وهي مثل المصفوفات .

١١-٢- أساسيات السلاسل المحرفية

يمكن التصريح عن سلسلة محارف بالشكل التالي :

```
char s [ ] = " Hello " ;
```

```
char *sp = " Hello " ;
```

وهو إسناد القيمة الابتدائية " Hello " إلى كل من المتحولين السابقين ، ويمكن كتابة التصريح كما يلي :

```
char s [ ] = { " Hello " } ;
```


أو على الشكل التالي :

```
char s[] = {'H','e','l','l','o','\0'};
```

عند التصريح عن مصفوفة محارف على أنها تتضمن سلسلة ، فيجب أن تكون هذه المصفوفة ذات حجم كاف بحيث يمكن تخزين محارف السلسلة والمحرف الصفري ضمنها ، كما هو موضح في المثال (1).

مثال (1-11)

```
//This program uses pointer increment and decrement
#include<iostream.h>
#include<string.h>
void main( ) {
    char color[]="blue";
    char *colorPtr;
    colorPtr=color;
    cout<< "The color is "<<color<<endl;
    cout<<"colorPtr= "<<colorPtr<<endl;
    cout<<"*colorPtr = "<<*(colorPtr)<<endl;
    colorPtr++;
    cout<<"*colorPtr = "<<*(colorPtr)<<endl;
    colorPtr--;
    cout<<"*colorPtr = "<<*(colorPtr)<<endl;
} //End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

```
RESULT
The color is blue
colorPtr= blue
*colorPtr = b
*colorPtr = l
*colorPtr = b
Press any key to continue
```

يمكن إسناد سلسلة من المحارف إلى مصفوفة [10] a باستخدام تعليمة الإدخال cin ، مثل :

```
cin >> a ;
```

تُخزن السلسلة المدخلة من قبل المستخدم في المصفوفة a ، ونقوم التعليمة بقراءة المحارف المدخلة حتى الوصول إلى فراغ أو حرف جدولة tab أو محرف سطر جديد أو محرف نهاية الملف ، ويجب أن لا يكون طول السلسلة أطول من 9 محارف لأن المحرف العاشر مخصص للمحرف الصفري .

نرغب أحياناً في إدخال سطر كامل إلى مصفوفة ، من أجل ذلك ، يتوفر في لغة C++ التابع cin.getline() الذي يأخذ ثلاثة وسطاء فعليين هم :

- ❖ المصفوفة التي نرغب في تخزين السطر فيها .
- ❖ وطول هذه المصفوفة
- ❖ والمحرف الذي يحدد نهاية السطر .

مثلاً ، الجزء البرمجي التالي :

```
char sent [ 100 ] ;  
cin.getline ( sent , 100 , '\n' ) ;
```

يقوم هذا الجزء البرمجي بالتصريح عن المصفوفة sent المؤلفة من 100
محرف ، ثم يقرأ سطرًا من النص المدخل بواسطة لوحة المفاتيح على المصفوفة
sent ، وينتهي التابع عمله عندما يقرأ المحرف '\n' أو عندما يصادف محرف
نهاية الملف أو عندما يتجاوز عدد المحارف المدخلة أقل من قيمة الوسيط الثاني
بواحد ، كما هو موضح في المثال (2-11) .

مثال (2-11)

```
/* This program prints some information using cin.getline  
Function */  
#include<iostream.h>  
#include<iomanip.h>  
#include<string.h>  
void main( ) {  
    char name[20];  
    char address[25];  
    char city[25];  
    char telephone[20];  
    cout<<"Enter the name :";  
    cin.getline(name,20,'\n');  
    cout<<"Enter the address :";
```

```

cin.getline(address,25,'\n');
cout<<"Enter the city and state :";
cin.getline(city,25,'\n');
cout<<"Enter the telephone number :";
cin.getline(telephone,20,'\n');

cout<<" The name is : "<<name<<endl;
cout<<"The address is: " <<address<<endl;
cout<<"The city and state are: "<<city<<endl;
cout<<"The telephone number is: "<<telephone
<<endl;
} //End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

Enter the name : **Waseem**
Enter the address : **Baghdad Street**
Enter the city and state : **Damascus**
Enter the telephone number : **011-4356677**

The name is : **Waseem**
The address is: **Baghdad Street**
The city and state are: **Damascus**
The telephone number is: **011-4356677**

Press any key to continue

١١-٣- توابع مكتبة السلاسل المحرفية

يتوافر في لغة C++ عدد من توابع المحارف والتي تستخدم في التعامل مع معطيات سلاسل المحارف في مقارنتها ، وفي البحث عنها ، وفي تقسيمها إلى عدة أقسام منطقية ومن أجل تحديد أطولها، وتوجد هذه التوابع ضمن الملف الرأسي :

include < string . h >

مثل : نسخ سلاسل المحارف ، ولصق سلاسل المحارف ، والمقارنة بين سلاسل المحارف ، وتحديد أطوال سلاسل المحارف ، وتقسيم سلاسل المحارف وغيرها .

١١-٣-١- نسخ سلاسل المحارف

يمكن نسخ سلاسل المحارف باستخدام التابعين **strcpy()** & **strncpy()** .

أ- التابع **strcpy(S1,S2)**

يقوم هذا التابع بنسخ محتوى السلسلة S2 إلى محتوى السلسلة S1 ، حيث S1 وسيطه الأول و S2 وسيطه الثاني ، وإعادة قيمة S1 الناتجة ، والشكل العام لاستدعاء هذا التابع كما يلي :

strcpy (S1 , S2) ;

لا يحتاج هذا التابع إلى نسخ المحرف الصفري لوسيطه الثاني .

ب- التابع **strncpy(S1 , S2 , n)**

يقوم هذا التابع بنسخ n محرف من S2 إلى S1 وإعادة قيمة S1 الناتجة ، والشكل العام لاستدعاء هذا التابع كما يلي :

strcpy (S1 , S2 ,12) ;

المثال (3-11) يوضح كيفية استخدام التابع strcpy () و التابع strcpy () .

مثال (3-11)

اكتب برنامجاً بلغة C++ يقوم بتعريف ثلاث سلاسل (مصفوفات) كما يلي :

: - انسخ السلسلة x إلى السلسلة y ثم اطبع السلسلتين .
: - انسخ 14 حرف من السلسلة x إلى السلسلة y ثم اطبع السلسلتين .

// Using strcpy and strncpy.

```
#include <iostream.h>
```

```
#include <string.h> // prototypes for strcpy and strncpy
```

```
int main( ) {
```

```
    char x[ ] = " Happy Birthday to You " ;
```

```
    char y[ 25 ];
```

```
    char z[ 15 ];
```

```
    strcpy( y, x ); // copy contents of x into y
```

```
    cout << "The string in array x is: " << x
```

```
        << "\nThe string in array y is: " << y << "\n";
```

```
    // copy first 14 characters of x into z
```

```
    strncpy( z, x, 14 ); // does not copy null character
```

```
    z[ 14 ] = '\0'; // append '\0' to z's contents
```

```
    cout << "The string in array z is: " << z << endl;
```

```
return 0; // indicates successful termination
} // End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

The string in array x is: **Happy Birthday to You**

The string in array y is: **Happy Birthday to You**

The string in array z is: **Happy Birthday**

Press any key to continue

١١-٣-٢- لصق سلاسل المحارف

يمكن لصق سلاسل المحارف باستخدام التابعين `strcat( )` & `strncat( )`

أ- التابع `strcat( S1 , S2 )`

يقوم التابع بإضافة السلسلة `S2` إلى نهاية السلسلة `S1` ، بحيث المحرف الأول لـ `S2` يحل محل المحرف الصفري لـ `S1` وتعاد السلسلة `S1` الناتجة ، والشكل العام لاستدعاء هذا التابع كما يلي :

`strcat( S1 , S2 )`

ب- التابع `strncat( S1 , S2 , n )`

يقوم التابع بإضافة `n` محرف من السلسلة `S2` إلى نهاية السلسلة `S1` وإعادة قيمة `S1` الناتجة ، والشكل العام لاستدعاء هذا التابع كما يلي :

strcat(S1 , S2,n)

المثال (4-11) يوضح كيفية عمل التابعين strcat() و strncat() .

مثال (4-11)

اكتب برنامجاً بلغة C++ يقوم بتعريف ثلاث سلاسل (مصفوفات) كما يلي :
، s3[40]=" " ، s2[]="New Year" ، S1[20] = " Happy "
والمطلوب :

- انسخ السلسلة s2 إلى السلسلة s1 ، ثم اطبع السلسلتين.
- انسخ 6 محارف من السلسلة s1 إلى السلسلة s3 ثم اطبع السلسلتين.
- انسخ السلسلة s1 إلى السلسلة s3 ، ثم اطبع السلسلتين.

البرنامج :

// Using strcat and strncat.

#include <iostream.h>

#include <string .h> // prototypes for *strcat* and *strncat*

void main()

{

char s1[20] = "Happy " ;

char s2[] = " New Year " ;

char s3[40] = " " ;

cout << "s1 = " << s1 << "\ns2 = " << s2;

strcat(s1, s2); // concatenate s2 to s1

cout << "\n\nAfter strcat(s1, s2):\n s1 = "

<< s1 << "\ns2 = " << s2;

```

// ----- concatenate first 6 characters of s1 to s3-----
strncat( s3, s1, 6 ); // places '\0' after last character
cout << "\n\nAfter strncat(s3, s1, 6):\ns1 = "
      << s1 << "\ns3 = " << s3;
strcat( s3, s1 ); // --- concatenate s1 to s3-----
cout << "\n\nAfter strcat(s3, s1):\ns1 = "
      << s1 << "\ns3 = " << s3 << endl;
// indicates successful termination
} // End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

```

s1 = Happy
s2 = New Year
After strcat(s1, s2):
s1 = Happy New Year
s2 = New Year
After strncat(s3, s1, 6):
s1 = Happy New Year
s3 = Happy
After strcat(s3, s1):
s1 = Happy New Year
s3 = Happy Happy New Year
Press any key to continue

```


١١-٣-٣- المقارنة بين سلاسل المحارف

يمكن مقارنة سلاسل المحارف باستخدام التابعين `strcmp()` و `strncmp()`

أ- التابع `strcmp( S1 , S2 )`

يقوم التابع بمقارنة السلسلتين `S1` و `S2` ويعيد التابع القيمة صفر في حال

$S1 = S2$ ، وإذا كانت $S1 < S2$ يعيد القيمة (-1) ، وإذا كانت $S1 > S2$

يعيد القيمة $(+1)$ ، والشكل العام لاستدعاء هذا التابع كما يلي :

`strcmp( S1 , S2 )`

ب- التابع `strncmp( S1 , S2 , n )`

يقوم التابع بمقارنة `n` محرف من السلسلة `S2` مع السلسلة `S1` ، ويعيد

التابع القيمة صفر في حال $S1 = S2$ ، وإذا كانت $S1 < S2$ يعيد القيمة (-1) ،

وإذا كانت $S1 > S2$ يعيد القيمة $(+1)$ ، والشكل العام لاستدعاء هذا التابع كما

يلي :

`strcmp( S1 , S2,n )`

المثال (5-11) يوضح كيفية عمل التابعين `strcmp()` و `strncmp()` .

مثال (5-11)

اكتب برنامجاً بلغة `C++` يقوم بتعريف ثلاث سلاسل (مصفوفات) كما يلي :

`*s1 = "Happy New Year"` ، `*s2 = "Happy New Year"` ،

`*s3 = "Happy Holidays"` ، والمطلوب :

- اطبع السلسلة `s1` و `s2` و `s3` .

- قارن السلسلة `s1` مع السلسلة `s2` ثم اطبع النتيجة.

- قارن السلسلة s2 مع السلسلة s1 ثم اطبع النتيجة.
- قارن السلسلة s1 مع السلسلة s3 ثم اطبع النتيجة.
- قارن السلسلة s3 مع السلسلة s1 ثم اطبع النتيجة.
- قارن 6 محارف من السلسلة s3 مع السلسلة s1 ثم اطبع النتيجة.
- قارن 7 محارف من السلسلة s3 مع السلسلة s1 ثم اطبع النتيجة.
- قارن 7 محارف من السلسلة s1 مع السلسلة s3 ثم اطبع النتيجة.

البرنامج :

```
// Using strcmp and strncmp.
#include <iostream.h>
#include <iomanip.h>
#include <string.h> // prototypes for strcmp & strncmp
void main( ) {
    char *s1 = "Happy New Year";
    char *s2 = "Happy New Year";
    char *s3 = "Happy Holidays" ;
    cout << "s1 = " << s1 << "\ns2 = "
        << s2 << "\ns3 = " << s3
        << "\n\nstrcmp(s1, s2) = "
        << setw( 2 ) << strcmp( s1, s2 )
        << "\n\nstrcmp(s1, s3) = " << setw( 2 )
        << strcmp( s1, s3 ) << "\n\nstrcmp(s3, s1) = "
        << setw( 2 ) << strcmp( s3, s1 );
    cout << "\n\nstrncmp(s1, s3, 6) = "
```

```

    << setw( 2 ) << strncmp( s1, s3, 6 )
    << "\nstrncmp(s1, s3, 7) = "
    << setw( 2 ) << strncmp( s1, s3, 7 )
    << "\nstrncmp(s3, s1, 7) = "
    << setw( 2 ) << strncmp( s3, s1, 7 ) << endl;
} // End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

s1 = Happy New Year

s2 = Happy New Year

s3 = Happy Holidays

strcmp(s1, s2) = 0

strcmp(s1, s3) = 1

strcmp(s3, s1) = -1

strncmp(s1, s3, 6) = 0

strncmp(s1, s3, 7) = 1

strncmp(s3, s1, 7) = -1

Press any key to continue

١١-٣-٤ - تقسيم سلاسل المحارف إلى وحدات token

يمكن تقسيم سلاسل المحارف إلى وحدات باستخدام التابع (S1 , S2) strtok .

أ- التابع (S1 , S2) strtok

يقوم التابع بتقسيم السلسلة S1 إلى وحدات tokens وتتألف كل وحدة من عدة محارف يفصل بينها محارف فواصل ، يتسبب الاستدعاء المتكرر للتابع strtok في تقطيع السلسلة إلى عدة وحدات ، ويأخذ الاستدعاء الأول للتابع strtok وسيطين هما السلسلة المطلوب تقطيعها إلى وحدات ، وسلسلة أخرى تتضمن المحارف التي تفصل بين الوحدات ، ويعيد التابع مؤشراً يدل على بداية الوحدة التي أوجدها بعد كل استدعاء ، ويعيد NULL في حال عدم وجود أية وحدة ، ويوضح البرنامج في المثال (6-11) عمل التابع (S1 , S2) strtok .

مثال (6-11)

اكتب برنامجاً بلغة C++ يقوم بتعريف سلسلة كما يلي :

char sentence[] = " This is a sentence with 7 tokens "

وبتعريف مؤشر char *tokenPtr أيضاً ، والمطلوب :

- تقسيم السلسلة sentence إلى وحدات وطباعتها ، حيث الفاصل بين الوحدات هو الفراغ space .

```
// Using strtok.
```

```
#include <iostream.h>
```

```
#include <string.h> // prototype for strtok
```

```

void main()
{
    char sentence[ ] = " This is a sentence with 7 tokens " ;
    char *tokenPtr;
    cout << "The string to be tokenized is:\n" << sentence
        << "\n\nThe tokens are:\n\n";
    // Begin tokenization of sentence
    tokenPtr = strtok( sentence, " " );
    // continue tokenizing sentence until tokenPtr becomes NULL
    while ( tokenPtr != NULL ) {
        cout << tokenPtr << '\n';
        // get next token
        tokenPtr = strtok( NULL, " " );
    } // End while
    cout << "\nAfter strtok, sentence = " << sentence << endl;
} // End main

```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

RESULT

The string to be tokenized is:

This is a sentence with 7 tokens

The tokens are:

This

is

a

sentence

with

7

tokens

After strtok, sentence = This

Press any key to continue

١١-٣-٥ - حساب أطوال سلاسل المحارف

يمكن حساب أطوال سلاسل المحارف باستخدام التابع `strlen( S )` .

أ- التابع `strlen( S )`

يحسب التابع عدد محارف السلسلة التي تسبق المحرف الصفري الذي

يحدد نهايتها ، والشكل العام لاستدعاء هذا التابع كما يلي :

`strlen( S )`

ويوضح البرنامج في المثال (7-11) التالي عمل التابع `strlen( S )` .

مثال (7-11)

اكتب برنامجاً بلغة ++C يقوم بإسناد ثلاث سلاسل محرفية إلى ثلاثة مؤشرات كما يلي :

```
char *string1 = " abcdefghijklmnopqrstuvwxyz " ;  
char *string2 = "Computer Engineering and Automation " ;  
char *string3 = "Damascus " ;
```

والمطلوب :

- حساب أطوال هذه السلاسل باستخدام التابع (S) strlen .

// Using strlen.

```
#include <iostream.h>
```

```
#include <string.h> // prototype for strlen
```

```
void main() {
```

```
char *string1 = " abcdefghijklmnopqrstuvwxyz " ;
```

```
char *string2 = " Computer Engineering and Automation " ;
```

```
char *string3 = " Damascus " ;
```

```
cout << "The length of \" << string1
```

```
<< \" is \" << strlen( string1 )
```

```
<< \"\n\"The length of \" << string2
```

```
<< \" is \" << strlen( string2 )
```

```
<< \"\n\"The length of \" << string3
```

```
<< "\" is " << strlen( string3 ) << endl;  
} // End main
```

وعند تنفيذ البرنامج سنحصل على الخرج التالي:

Result

The length of "abcdefghijklmnopqrstuvwxyz" is 26

The length of " Computer Engineering and
Automation " is 35

The length of " Damascus " is 8



المراجع الأجنبية

References

1. OBJECT ORIENTED PROGRAMMING WITH C++ 4E books 13 Digit ISBN 9780070669079 ,Publication Year 2010 , Authors E.BALAGURUSAMY , Book Edition 4 , Book Binding Type SOFTCOVER , Number of pages N/A
2. DAVID LEVINE.“ TEACH YOURSELF C++” .Second Edition, Greg Wiegand Sams Publishing 201 W. 103rd Street Indianapolis, IN 46290, 772 p.
3. DEITEL P.J ,DEITEL H.M.[2007]. “ C HOW TO PROGRAM”. Fifth Edition,2007 by Pearson Education, Inc.Upper Saddle River, New Jersey 07458,1110 p.
4. DEITEL P.J ,DEITEL H.M.[2008]. “ C++ HOW TO PROGRAM”.Sixth Edition,2008 by Pearson Education , Inc.Upper Saddle River, New Jersey 07458,1429 p.
5. DOUGLAS C.SCHMIDT . ” AN OVERVIEW OF C++ ” Vanderbilt University, www.cs.wustl.edu, September 8, 2003.
6. FRANCOIS FLEURET . “ C++ LECTURE NOTES ”. University of Chicago, 2010.
7. GREG PERRY . “C++ By EXAMPLE” . 1992 by Que Printed in the United States of America 761 p.
8. HORSTMANN C.S.[1999] “COMPUTING CONCEPTS WITH C++ ESSENTIALS”. Second Edition,1999 by John Wiley & Sons, Inc.327

9. Herbert Schildt C++: The Complete Reference, Fourth Edition, Copyright © 2007 by The McGraw-Hill Companies.
10. The C++ Programming Language ,2012,Third Edition and Special Edition Addison-Wesley, ISBN 0-201-88954-4 and 0-201-70073-5
11. SHARAM HEKMAT ."C++ PROGRAMMING ". 1998 Pragmatics Software Pty. Ltd.
12. Hennessy, John L. Patterson, David A.. Arpaci-dusseau, Andrea C. Arpaci-Dusseau, Remzi H.. Asanovic, Krste,**Computer Architecture : A Quantitative Approach**, Elsevier Science Ltd , 2006.
13. Barry B. Brey .The Intel Microprocessors ,7e ,Pearson Education ,2006 .
14. John L. Hennessy, David Goldberg , **Computer Architecture: A Quantitative Approach** , Morgan Kaufmann Pub,2006 .
15. James Goodman, **A programmer's view of computer architecture: With examples from the MIPS RISC architecture** , Saunders College Pub,2004 .
16. Heuring, Vincent P. Jordan, Harry F. Murdocca, Miles, **Computer Systems Design and Architecture**, Addison-Wesley, 1996



المراجع العربية

١. التفكير والإبداع في لغة ++C - المهندس إحسان أحمد الرفاعي، المهندس ضياء زهير الأدرنلي، المهندس ياسر عبد الكريم صبح- شعاع للنشر والعلوم- سورية- حلب- 2005.
٢. الدليل الشامل للاحتراف في لغة ++C - عبد الحافظ ، يمان البني . دار البراق - سورية- 2004 .
٣. كيف تبرمج بلغة ++C - الدكتور صلاح الدوه جي- شعاع للنشر والعلوم - سورية- حلب- 2004.
٤. البرمجة (١) - البرمجة غرضية التوجه - الدكتور مأمون يونس & الدكتور أكرم مذكور -جامعة دمشق - سوريا - 2014.
٥. المرجع الأساسي لمستخدمي C/C++ - المهندس عزب محمد عزب، مجدي محمد أبو العطا- العربية لعلوم الحاسب- مصر - القاهرة- 1998.
٦. المرجع الأساسي في برمجة ++C ، د.عوض منصور وآخرون -الأردن - مؤسسة البشير الأردن - 2000 .
٧. الوحدات المحيطية للحاسوب - الدكتور مأمون يونس & الدكتور رزق غانم - جامعة دمشق - سوريا - ٢٠١١ .
٨. البنى المتقدمة للحاسوب - الدكتور مأمون يونس & الدكتور رزق غانم - الدكتور جمال الياسين- جامعة دمشق - سوريا - ٢٠١١ .
٩. الشبكات الحاسوبية وتراسل المعطيات - الدكتور جمال الياسين- جامعة دمشق - سوريا - ٢٠١٢ .



المصطلحات العلمية

إنكليزي-عربي

English-Arabic





A	
Absolute value	القيمة المطلقة
Access	برمجيات قواعد المعطيات
Access Time	زمن النفاذ
Accumulator	المراكم
Active Directory Server	مخدم خدمات الدليل الفعال
Adder	الجامع
Address	عنوان
Address	عنوان
Address Bus	مسرى العنوان
Address Bus	مسرى العنوان
Address Unit	وحدة العنوان
Algorithm	الخوارزمية
Algorithm of the Problem	خوارزمية المسألة
Allocate	يُخصص
Allocation error	خطأ التخصيص
American National Standard Institute(ANSI)	المعهد الوطني الأمريكي للمعايير القياسية
American Standard Code for Information Interchange(ASCII)	الترميز المعياري الأمريكي لتبادل المعلومات
Analog Signals	إشارات تماثلية

Analogue Computer	الحاسوب التماثلي
Application Programming Interface	واجهة برمجة التطبيقات
Application Server	مخدم تطبيقات
Application Software	البرمجيات التطبيقية
Area	مساحة
Arg –List	قائمة الوسطاء
Argument	وسيط
Arithmetic and Logic Unit(ALU)	وحدة الحساب والمنطق
Arithmetic Expression	تعبير حسابي
Arithmetic Operators	المؤثرات الحسابية
Arithmetic Statement	تعليلة حسابية
Array	مصفوفة
Array Definition	تعريف المصفوفة
Array index	دليل المصفوفة
Array Name	اسم المصفوفة
Array Of Pointers	مصفوفة المؤشرات
Arrow Keys	مفاتيح الأسهم
Artificial Intelligence	الذكاء الصناعي
Assemblers	المجمعات
Assembly Language	لغة التجميع
Assignment	إسناد
Assignment Statement	تعليلة الإسناد

Attenuation	التضاؤل
AutoCAD	برمجيات التصميم
Average	معدل
B	
Bar Code Reader	جهاز قارئ الأعمدة
Bar Code Reader	جهاز قارئ الأعمدة
Base	أساس (قاعدة)
BASIC	لغة بيزك
Basic Input Output System	نظام الإدخال والإخراج الأساسي
Batch Processing	المعالجة الدفعية
Binary Coded Decimal(BCD)	نظام العد العشري المرمز ثنائيا
Binary Number System	نظام العد الثنائي
Binary Representation for the Decimal Numbers	التمثيل الثنائي للأعداد العشرية
Binary Representation for the Octal Numbers	التمثيل الثنائي لنظام العد الثماني
Binary search	البحث الثنائي
BIOS	نظام الإدخال والإخراج الخاص بالحاسوب
Bit	خانة ثنائية
Bit Manipulation	التعامل على مستوى الخانة الثنائية
Block of statements	مجموعة من التعليمات
Blocks	مجموعة كتل

Body of the function	جسم التابع
Boolean Expression	تعبير منطقي
Branched Flowchart	المخطط التدفقي التفرعي
Bridge	الجسر
Broadcast Radio Transmissions	الأمواج الراديوية
Bubble Sort	الفرز الفقاعي
Byte	بايت
C	
Cache Memory	ذاكرة الكاش
Calling	استدعاء
Calling Function	استدعاء التابع
Carrier Signal	الإشارة الحاملة
Central Processing Unit (CPU)	وحدة المعالجة المركزية
Character	محرّف (حرف أو رقم أو رمز)
Character Type	نوع المحرف
Characters and Strings	المحارف وسلاسل المحارف
Check Disk	أداة تفحص الأقراص
Circle	دائرة
Classes	الصفوف
Clean Disk	أداة تنظيف القرص
Client / Server	شبكات المخدم/العميل

Clock Speed	تردد ساعة الحاسوب
Close Condition	الشرط المغلق
Coaxial Cables	الكابلات المحورية
Combined Programming Language CPL	لغة البرمجة المجمعة
Comments	تعليقات
Communication Lines	خطوط الاتصال
Communication Server	مخدم الاتصالات
Communication Switches	مبدلات الشبكة
Compact Disks (CD- ROM)	الأقراص الليزرية
Company	شركة
Complex	عقدي
Composition	تركيب
Compound Assigination Operators	مؤثرات الإسناد المركبة
Compute	يحسب
Computer	حاسوب
Computer Classifications	تصنيف الحواسيب
Computer Network	شبكات الحواسيب
Computer Networks Protocol	بروتوكولات شبكات الحواسيب
Computer Performance	أداء الحاسوب
Computer Peripheral	الوحدات المحيطة بالحاسوب
Computer Viruses	فيروسات الحاسوب
Condition	شرط

Conditional Operators	المؤثرات الشرطية
Constant	ثابت
Constant member function	تابع عضو ثابت
Continuous Sine Wave	إشارة جيبية مستمرة
Control and Looping Structures	بنى التحكم والتكرار
Control Unit	وحدة التحكم
Cost	التكلفة
Counter	عداد
Counters	العدادات
Counting Algorithm	خوارزمية العد
Cmpiler	مترجم
CPU speed	سرعة وحدة المعالجة
D	
Daisy Wheel	طابعة العجلة
Data	المعطيات (البيانات)
Data Bus	مسرى المعطيات
Data Bus	مسرى المعطيات
Data Members	المعطيات الأعضاء
Data Processing	معالجة المعطيات
Data Structures	بنى المعطيات
Data type	نوع البيانات
Database Server	مخدم قواعد معطيات

Decimal Number System	نظام العد العشري
Decision	اتخاذ القرار
Declaration	تصريح
Declaration of Variable	التصريح عن المتغير
Declarator	المصرح
Declaring Function	التصريح عن التابع
Decrement	النقصان
Default Arguments	الوسطاء الافتراضية
Defining Function	تعريف التابع
Digital Camera	الكاميرا الرقمية
Digital Camera	الكاميرا الرقمية
Digital Computer	الحاسوب الرقمي
Digital Signals	الإشارات الرقمية
Digital Video (Versatile)Disc(DVD)	أقراص الفيديو الرقمية
Direct Memory Access	النفوذ المباشر إلى الذاكرة
Directories	الأدلة
Disk Drivers	مشغلات الأقراص
Disk Operation System	نظام تشغيل الأقراص
Distance Education	التعلم عن بعد
Distance Education	التعلم عن بعد
Distribution Extranet	شبكات اكسترانت التوزيع
Dot Matrix Printer	الطابعة النقطية

Downloaded Program	برنامج التحميل
Draw	يرسم
Drived Data Types	أنواع البيانات المشتقة
Drum Scanner	الماسحات الاسطوانية
Dynamic RAM(DRAM)	ذاكرة ديناميكية
E	
e-commerce	التجارة الإلكترونية
Electronic Board	السطح الإلكترونية
Electronic Board	السطح الإلكترونية
Empty	خالي
EPROM (Erasable PROM)	الذاكرة المبرمجة القابلة للمحي
Escape Characters	محارف الهروب
Example	مثال
Excel	برمجيات معالجة الجداول الإلكترونية
Expansion Cards and Slots	شقوق وبطاقات التوسع
Expression	تعبير
Extended Binary Coded Decimal Interchange Code(EBCDIC)	الترميز الموسع للأرقام العشرية الممثلة بالنظام الثنائي لتبادل المعلومات
Extension	الامتداد
External	خارجي
External Storage Units	وحدات التخزين الخارجية

F	
Factor	مُعامل
Factorial	العاملِي (!)
False	خطأ
Fax	أجهزة الفاكس
Fax Server	مخدم فاكس
Fiber Optic Cables	كابلات الألياف الضوئية
File	ملف
File Server	مخدم ملفات
File system	نظام الملفات
File Transfer Protocol	خدمة نقل الملفات
Final value	القيمة النهائية
Fire wire	المنفذ الناري
First	الأول
Flags	مؤشرات
Flash Disk	ذاكرة الفلاش ديسك
Flash Memory	الذاكرة غير المتطايرة (الدائمة)
Flat bed	الماسح الضوئي
Flat bed	الماسحات المسطحة
Flat Panel Display	شاشات العرض المسطحة

Floppy Disks	الأقراص المرنة
Flow line	اتجاه سير العمليات
Flowchart	المخطط التدفقي
Flowchart Method	طريقة المخططات التدفقية
FORTRAN	لغة فورتران
Fraction Part	الجزء الكسري
Function	تابع
Function	تابع (دالة)
name Function	اسم التابع
Function Keys	مفاتيح الوظائف
Function Name	اسم التابع
Function Type	نوع التابع
H	
Hard Disk	القرص الصلب
Hardware	المكونات المادية
File Header	ملف رأسي
Hexadecimal Number System	نظام العد السداسي عشر
High Level Progamming Languages	اللغات البرمجية العالية المستوى
Hub	الموزع

Hybrid Computer	الحاسوب الهجين
Hypertext Transfer Protocol	بروتوكول نقل النص الفائق
G	
Graphics Acceleration Card	بطاقة الشاشة
Gateway	البوابة
GBR	سجلات الأغراض العامة
General Purpose Computers	حواسيب الأغراض العامة
Global	شمولي
Global Variable	متغير شمولي
Gopher	خدمة الغوفر
Graphic Card	بطاقة الرسوم
Graphical User Interface	المواجهة الرسومية
Graphics	رسومات
I	
I/O Functions	توابع الإدخال والإخراج
IC	الدارة المتكاملة
Identifiers	المعرفات
Increment	الزيادة
Increment and Decrement Operators	مؤثرات الزيادة والنقصان
Independent	مستقل

Information Management Programs	برمجيات إدارة المعلومات
Infrared Data Association	منفذ معطيات الأشعة الحمراء IrDA
Initial value	القيمة الابتدائية
Initial Value	القيمة الابتدائية
Initialization	تهيئة
Ink Jet Printer	الطابعة النافثة للحبر
Inline Functions	التوابع الفورية (السطرية)
Inner Loop	الحلقة الداخلية
Data Input	إدخال البيانات
Input Statement	تعليلة الإدخال
Input Units	وحدات الإدخال
Input/Output Management Programs	برمجيات وحدات الإدخال والإخراج
Insertion	إدراج
Insertion Operator	عملية الإدراج
Installing Requirements	متطلبات التركيب
Institute of Electrical And Electronic Engineers(IEEE)	معهد مهندسي الكهرباء والإلكترونيات
Instruction	التعليمات
Instruction Unit	وحدة التعليمات
Integer	صحيح
Integer Constant	ثابت صحيح
Integer Variable	متغير صحيح
Integer Part	الجزء الصحيح

Integrated Circuits	الدارات المتكاملة
Interface	واجهة تخاطب
Interface Unit	وحدة الترابط مع الحاسوب
Internal	داخلي
International Standard Organization(ISO)	المنظمة الدولية للمواصفات والمقاييس
Internet Explorer	متصفح الانترنت
Interpreters	المفسرات
J	
Joy stick	عصا التحكم بالألعاب
K	
Kernel	النواة
Keyword	كلمة محجوزة
L	
LAN Card	بطاقة الشبكة
Language Method	الطريقة اللغوية
Laptop	الحاسوب المحمول
Laser Printer	الطابعة الليزرية
Last	الأخير
Least significant Digit(LSD)	خانة الدلالة الدنيا

Length	طول
Lexical analysis	التحليل المعجمي
Light Pen	القلم الضوئي
Light Pen	القلم الضوئي
Line Command Interface	المواجهة الخطية
Linear search	البحث الخطي
Linking	ربط
Liquid Crystal Display (LCD)	شاشة السائل البلوري
List of Variable	قائمة المتغيرات
Loader	محمل
Loading	تحميل
Local Area Networks(LAN)	الشبكات المحلية
Local Variable	متغير محلي
Logic Error	خطأ منطقي
Logical Operators	المؤثرات المنطقية
Logical Constant	ثابت منطقي
Logical Expression	تعبير منطقي
Logical Variable	متغير منطقي
Loop	حلقة تكرار
Loop Flowchart	المخطط التدفقي الحلقي
Low Level Progamming	اللغات البرمجية المتدنية المستوى

M	
Machine Language	لغة الآلة
Mail Server	مخدم البريد الإلكتروني
Main Memory	الذاكرة الرئيسي
Main Memory Unit	وحدة الذاكرة الرئيسية
Main Program	البرنامج الرئيسي
Maine Frame	الحواسيب الكبيرة
Mainframe	الحواسيب الكبيرة
Manipulator	معالج
Mapping Function	تابع الإسقاط
Math Library Functions	مكتبة الدوال الرياضية
Mathematics	الرياضيات
Matrixes	مصفوفات
Medium Scale Integration	الدارات المتكاملة المتوسطة كثافة التكامل
Member	عضو
Member functions	التتابع الأعضاء
Management Memory Programs	برمجيات إدارة الذاكرة الرئيسية
Menu Interfac	المواجهة بالقوائم
Metropolitan Area Networks(MAN)	شبكات المدينة
Microcomputer	الحاسوب الميكروي
Microelectronics	الإلكترونيات الصغيرة

Microphone	جهاز اللاقط
Microphone	جهاز الميكروفون
Microprocessor	معالج ميكروي
Microprocessors	المعالجات الميكروية (الصغيرة)
Microwave System	الأمواج الميكروية
Mid	الأوسط
Minicomputer	الحواسيب المتوسطة
Minicomputer	الحواسيب المتوسطة
Minutes	دقائق
Modem	المودم
Card Modem / Fax	بطاقة الفاكس/الموديم
Modulation and Demodulation	التعديل وفك التعديل
Modules	وحدات (أجزاء)
Monitor	جهاز العرض (شاشة)
Month	شهر
Most Significant Digit(MSD)	خانة الدلالة العليا
Mother Board	اللوحة الأم
Mother Board	اللوحة الأم
Mouse	الفأرة
Multimedia	الوسائط المتعددة
Multimedia	الوسائط المتعددة
Multiprocessing Systems	نظم المعالجة المتعددة

Multiprogramming Systems	نظم البرمجة المتعددة
Multi-Tasking	أنظمة متعددة المهام
Multi-user	أنظمة متعددة المستخدمين
N	
Name	الاسم
Names mangling	تشويه الأسماء
Naming of variables	تسمية المتغيرات
Negation	النفي
Negative	سالب
Nested Loop Flowchart	المخططات التدفقية الحلقية المتداخلة
Nested loops	الحلقات المتداخلة
Network Interface Card	بطاقة الشبكة
Networks Topology	طبولوجيا الشبكات
Number	عدد
Number Keys	مفاتيح الأرقام
Number Systems	أنظمة العد
Numerical Constant	الثابت العددي
Numerical Variable	المتغير العددي
O	
Object	غرض (هدف)

Object Oriented Programming	البرمجة غرضية التوجه
Object program	البرنامج الهدف
Octal Number System	نظام العد الثماني
One-Dimensional Array	مصفوفة أحادية البعد
OOP	البرمجة غرضية التوجه
Operator	عملية (مؤثر)
Optical	ضوئي
Outer Loop	الحلقة الخارجية
Outlook Express	برمجيات التراسل الإلكتروني
Output Data	إخراج البيانات
Output Statement	تعلية الإخراج
Output Units	وحدات الإخراج
Output Units	وحدات الإخراج
Overloading Functions	توابع التحميل الزائد
P	
Portable) Computer (Laptop	الحاسوب المحمول
Package Software	البرمجيات التطبيقية الجاهزة
Page	صفحة
parallel Port	البوابة التفرعية
Parameter	وسيط
parameter list	قائمة الوسطاء

Parameterized Types	الأنماط ذات الوسطاء
Pass	يمرر
Pass by Pointer	التمرير بالمؤشر
Pass by Reference	التمرير بالمرجع
Pass by Value	التمرير بالقيمة
Peer Extranet	شبكات اكسترانت الند
Peer-to-Peer	شبكات الند للند
Personal Digital Assistants (PDA)	المساعدات الرقمية الشخصية
Photo Shop	برمجيات معالجة الصور
Plotters	الراسمات
Pointer	مؤشر
Pointers	المؤشرات
Pointers of Functions	مؤشرات التوابع
Pop character	يُدخل محرفاً (حرفاً)
Language Portable	لغة متنقلة
Ports	المنافذ
positive	موجب
Power Point	برمجيات عرض الشرائح
Power Supply	وحدة التغذية الكهربائية
Print	يطبع
Print decimal value	يطبع قيمة بالنظام العشري
Print Head	رأس الطابعة

Print hexadecimal value	يطبع قيمة بالنظام السداسي عشر
Print octal value	يطبع قيمة بالنظام الثماني
Printer	الطابعة
Printer Server	مخدم طباعة
Process	يعالج
Processes Management Programs	برمجيات إدارة المعالجات
Processing	معالجة
Program	البرنامج
PROM(Programmable ROM)	الذاكرة القابلة للبرمجة
Programming	البرمجة
Provider Service	وكيل محلي
Pseudo code Method	طريقة شبه البرنامج
Q	
Quadratic equation	المعادلة التربيعية
Queues	صفوف الانتظار
Quotients	نواتج القسمة
R	
Random Access Memory	ذاكرة الوصول العشوائي
Read Only Memory (ROM)	ذاكرة القراءة فقط
Real	حقيقي

Real Constant	ثابت حقيقي
Rectangle	مستطيل
Recursive Functions	التتابع العودية (التعاودية)
Reference	مرجع
Refresh	إعادة إنعاش
Registers	المسجلات
Relational Operations	العمليات العلائقية (المنطقية)
Remainders	البواقي (بواقي القسمة)
Repeaters	مقويات الإشارة
Replacement Algorithm	خوارزمية الاستبدال
Result	نتيجة
Return	يُعيد
Return-Type	النوع المُعاد
Root	جذر
Router	الموجه
S	
Satellite	الأقمار الصناعية
SBR	سجلات الأغراض الخاصة
Scanner	الماسح الضوئي
Screen	الشاشة
SD memory	كرت الذاكرة الآمن

Second	ثانية
Secondary Storage Units	وحدات التخزين الثانوية
Serial Port	البوابة التسلسلية
Server	مخدم
Shell	الغلاف
Shielded Twisted Pair(STP)	الكابلات المجدولة المغلفة
Show	يعرض
Silicon	السياليكون
Simple sequential Flowchart	المخطط التدفقي التتابعي البسيط
Single-Tasking	أنظمة وحيدة المهام
Single-user	أنظمة وحيدة المستخدم
Small Computer System Interface	واجهة ترابط نظام الحاسوب الصغير
Integration (SSI) Small Scale	الدارات المتكاملة صغيرة كثافة التكامل
Software	البرمجيات
Solution	حل
Sound Card	بطاقة الصوت
Sound Cart	كرت الصوت
Sound System	النظام الصوتي
source program	البرنامج المصدر
Speakers	السماعات الصوتية
Special Keys	مفاتيح الحاسوب الخاصة
Special Purpose Computers	حواسيب الأغراض الخاصة

Stacks	المكدسات
Standard	معياري (قياسي)
Template Library Standard	مكتبة القوالب المعيارية
Start	البداية
Starting address	عنوان البداية
Statement	تعليلة
Static	ساكن
(SRAM) Static RAM	ذاكرة ساكنة
Stationary Scanners	ماسحات مثبتة
Stop	النهاية
String Constant	الثابت من نوع سلاسل المحارف
String Variables	المتغيرات من نوع سلاسل المحارف
Structure	بنية (سجل)
Structured Language	اللغة البنيوية
Structured Languages	اللغات البنيوية
Structured Programming	البرمجة البنيوية (الهيكلية)
Subdirectories	الأدلة الفرعية
Summers Algorithm	خوارزمية التجميع
Super Computer	الحواسيب الفائقة
Supercomputer	الحاسوب العملاق
Switches	مفاتيح الكترونية
Symbols	رموز
Syntax analysis	التحليل النحوي واللغوي

System Board	لوحة النظام
System File Protection	حماية ملفات النظام
T	
Tables	جداول
Tape Driver	الشريط الممغنط
Task Manager	إدارة المهام
Telecommunication Cards	بطاقات الاتصال
Temporary Files	ملفات مؤقتة
Temporary Internet Files	ملفات الانترنت المؤقتة
Test Data	المعطيات الاختبارية
Thick Coaxial Cable	الكبل المحوري السميك
Thin Coaxial Cable	الكبل المحوري الرفيع
Time Bombs	القنابل الموقوتة
Time Sharing System	نظام المشاركة بالزمن
Timing Unit	وحدة التوقيت والتتابع الزمني
Touch Screen	شاشة اللمس
Touch Screen	شاشة اللمس
Tow-Dimensional Array	مصفوفة ثنائية البعد
Tracks	المسالك
Transmission Speed	سرعة نقل المعلومات
Trojan Horses	أحصنة طروادة

TV Tuner	بطاقة التلفزيون
Typing Keys	مفاتيح الأحرف والرموز
U	
Ultraviolet Light	الأشعة فوق البنفسجية
Universal Serial Bus	المسرى التسلسلي العام
Unshielded Twisted Pair (UTP)	الكابلات المجدولة غير المغلفة
Using Pointer To Objects	استخدام المؤشر إلى الأغراض
Utilities Programs	البرامج الخدمية
V	
Vacuum Tubes	الصمامات المفرغة
Variable	متغير (متحول)
Video Display unit	وحدة العرض المرئي
Virtual Memory	الذاكرة الافتراضية
Virtual Memory	الذاكرة الافتراضية
W	
Wide Area Networks(WAN)	الشبكات الواسعة
Wire Communications	وسائط الاتصال السلكية
Word	برمجيات معالجة النصوص

Word size	عرض الكلمة
Work Stations	محطات العمل
World Wide Web(WWW)	الشبكة العنكبوتية العالمية
Worms	الفيروسات الدودية
Write Time	زمن الكتابة
Y	



اللجنة العلمية :

الدكتور غسان فلوح

الدكتور محمد مازن المحاييري

الدكتور رزق غانم

المدقق اللغوي :

الدكتور دياب راشد

حقوق الطبع والنشر محفوظة لمديرية الكتب والمطبوعات الجامعية

