

### البرمجة:

- الآلة الحاسبة بشكل عام هي عبارة عن صناديق إلكترونية مغلقة ولكي نستطيع استعمال هذه الآلة يجب احتواؤها على نظام تشغيل وعلى رزمة من البرامج عندها تصبح الآلة جاهزة للاستخدام والقيام بالمهام المطلوبة منها .

وهكذا فإن البرمجة هي أساس استعمال الآلة وعندما يكون البرنامج صحيحاً تقوم الآلة بإداء مهامها علي أكمل وجه وليس من الضروري أن يعبر البرنامج عن معادلة رياضية معقدة ولكن يجب أن يكون عبارة عن تسلسل منطقي لعدد من المعادلات والأوامر وعندما يفكر المبرمج بوضع برنامج معين لمسألته يجب أن يدرك الأمور التالية :

- أن يفهم المسألة فهماً صحيحاً وأن يحدد ما يريد منها وما يريد من نتائج .
- أن يحدد المعطيات المطلوب تلقينها للبرنامج
- أن يتأكد من إمكانية وضع برنامج للمسألة ومن مقدرة اللغة التي اختارها
- ان يحدد العمليات المطلوب اجراؤها

وهكذا للقيام بوضع برنامج معين يجب على المبرمج القيام بالعمليات التالية :

- تحليل المسألة
- وضع الخوارزم الخاص بالمسألة
- كتابة البرنامج
- تصحيح واختبار البرنامج
- توثيق البرنامج وتسجيل الملاحظات الخاصة بكيفية استعمال البرنامج كعمل تطبيقي.

### البرنامج :

هو مجموعة من الخطوات ( الأوامر ) المنطقية يكتبها المبرمج وذلك من أجل تأدية هدف معين وتنقسم البرامج إلي نوعين :

- برامج تطبيقية
- برامج النظم

### أولاً : البرامج التطبيقية :

☞ العمل والتجارة Business & Trade:

تعتبر استخدامات الحاسب في هذا المجال أكثر أهمية وشيوعاً بالنسبة للمستخدمين غير المتخصصين ففي الشركات والمؤسسات العامة والخاصة هناك كثير من الأعمال الأساسية يقوم عليها نشاط أي مكتب وقد تم تصميم وكتابة برامج كثيرة لمعالجة هذه المتطلبات وأهمها هي:-

☞ برامج الكتابة Word Processors:- ويطلق عليها البعض معالجة الكلمات أو النصوص وتستخدم أساساً في أعمال المكتبية مثل كتابة وحفظ وتوثيق المراسلات والتقارير والبحوث والجدول.

☞ برامج الجداول والحسابات Spread Sheets:- ويطلق عليها البعض "الجدول الإلكترونية" وتستخدم هذه النوعية في الأعمال المالية والمحاسبية.

☞ برامج الأرشيف Data Base:- وتسمى أيضاً برامج "قواعد البيانات" وتستخدم هذه النوعية من البرامج أساساً عند التعامل مع كميات كبيرة من البيانات مثل المخازن والمرتبات والعملاء والموردون وغيرهم.

☞ برامج العروض Visual Aids:- وتستخدم في الندوات Presentation والاجتماعات Meeting لعرض وجهات النظر بطريقة مؤثرة على الآخرين.

برامج الإنترنت Internet: وإتقان هذه النوعية من البرامج تستخدم أساسا في تسهيل الانتقال والعمل داخل الشبكة البينية الموسعة والتي أصبحت ضرورة علي المعلومات علي نطاق عالمي.

التصميم أو الرسم الهندسي (Computer Aided Design (CAD وهذه النوعية من البرامج تعمل على مساعدة المهندسين على عملية التصميم والرسم وذلك في مختلف المجالات الهندسية مثل تصميم الديكور وتصميم السيارات وتصميم المباني الخ.

التعليم Computer Based training: والتي يمكن من خلال الحاسب الشخصي تقديم عملية تعليمية مؤثرة وفعالة في كل المناهج.

الأرصاد الجوية Meteorology: وذلك بالتنبؤ بحالة الجو من حرارة ورطوبة وتسجيل الظواهر الجوية ومقارنة الظروف المناخية في نفس الأوقات من السنة عبر مدد زمنية.

الألعاب Games: برامج الألعاب بمختلف أنواعها مثل سباق السيارات ولعبة الشطرنج وكرة القدم وغيرها.

الإحصاء Statistics: إن قدرة الحاسب المتميزة على القيام بالعمليات الإحصائية تجعله الأداة الفعالة والمثالية للمشاركة في الأبحاث العلمية في فروع العلوم المختلفة، فيمكن للحاسب استخلاص النتائج من العلاقات بين القيم وإبرازها في صورة بيانية مما يجعل فهمها والتعامل معها والاستفادة منها أمرا سهلا.

برامج خدمية يكتبها المبرمج لحل بعض المسائل ومن امثلتها برامج ( الرواتب- الدراسة و الامتحانات- الميزانية- المكتبات - الشؤون الإدارية- المخازن )

### **ثانياً : برامج النظم :**

تكتب من قبل مبرمجي النظم المتخصصين لمساعدة مستخدمي الحاسب للاستفادة منه علي أحسن وجه ومن أهمها : لغات البرمجة - أنظمة التشغيل

### **أ- لغات البرمجة :**

هي وسيلة اتصال بين أجهزة الحاسب والشخص المشغل وتنقسم إلى :

#### **1- برنامج المصدر Source program :**

هي عبارة عن مجموعة من البرامج والتعليمات المكتوبة بشكل منطقي ومتسلسل بإحدى لغات البرمجة المتوفرة في الجهاز من قبل مبرمج أو مجموعة من المبرمجين وكل تعليمه من هذا البرنامج هي توجيه للحاسب لأداء عملية معينة من المسألة المعطاة.

#### **2- برنامج المترجمات Compilers :**

للحاسب الآلي لغة واحدة يفهمها هي لغة الآلة ولذا يجب ترجمة البرنامج المكتوب من قبل المبرمج إلى برنامج لغة الآلة ويتم ذلك باستخدام إحدى المترجمات التالية :

**المرجم Compiler :** كل لغة من اللغات الراقية لها مترجم خاص بها ومهمته :

- كشف وتسجيل الأخطاء التي تحدث في البرنامج
- ترجمة هذا البرنامج إلى لغة الآلة

**المفسر Interpreter :** يفحص ويترجم كل أمر من أوامر البرنامج الأساسي أولاً بأول ويبلغ المبرمج بالخطأ في نفس الوقت

**3- برنامج الهدف Object Program :**  
هو برنامج قابل للتنفيذ على الحاسب ولا يمكن التعديل فيه لأنه مكتوب بلغة الآلة .

### لغات الحاسب :

وتنقسم إلى :

- لغات المستوى الأدنى Low Level Languages
- لغات المستوى العالي High Level Languages
- اللغات الراقية

### لغات المستوى الأدنى Low Level Languages :

- 1- لغة الآلة Machin Language : هي أول لغة وضعت للحاسب وهي عبارة عن مجموعة صفوف من الأرقام الثنائية (0،1) .
- 2- لغة التجميع Assembly Language : تستخدم مجموعة من المصطلحات العلمية مثل ( ...\_Shift - Mov - ADD ) وتحتاج إلى مترجم يترجمها إلى لغة الآلة.

### لغات المستوى العالي High Level Languages :

- لغات انتشرت مع انتشار الحاسب وتستخدم في شتى المجالات وهي سهلة الاستعمال وحلت المشاكل التي واجهت المبرمجين بلغة التجميع ومن أهمها :
- 1- لغة فورتران Fortran Language : تستخدم في المسائل الرياضية والهندسية والعلمية .
  - 2- لغة كوبول Cobol Language : تستخدم في التقارير والتطبيقات الإدارية والتجارية.
  - 3- لغة البيسك Basic Language : أكثر اللغات انتشاراً لسهولة استعمالها وتستخدم لحل جميع أنواع المسائل العلمية والتجارية .
  - 4- لغة البسكال Pascal Language : لغة برمجية مرتبة وسهلة المتابعة ظهرت أواخر الخمسينات أطلق عليها اسم العالم Pascal.
  - 5- لغة السي C-Language : من اللغات الحديثة وحقت نجاحاً باهراً ومن أكثر اللغات رواجاً وتناسب جميع المبرمجين وغنية باللغات الهامة وتستخدم لحل جميع الأغراض ( علمية-هندسية -التجارية وغيرها ) .

6- لغة قواعد البيانات data Base Language : تستخدم مع البيانات والاحصاءات والمعلومات الكبيرة وتستخدم بكثرة في الاعمال الاداريه وكتابة التقارير .

### اللغات الراقية :

أبسط وأيسر اللغات ومن أمثلتها ( Delphi-Visual Basic –Visual C )

### **ب- أنظمة التشغيل Operating System**

وظيفتها التنظيم والتحكم في تشغيل وإدارة الحاسب آلي ومن أهمها ( Dos – Windows- Unix ) .

#### **تعريف نظام التشغيل Dos:**

نظام التشغيل عبارة عن برنامج جاهز Ready Made Program يتكون من مجموعة برامج تحتوي على تعليمات Instructions وأوامر Commands مكتوبة بإحدى لغات البرمجة القريبة للغة الآلة، وتمثل هذه البرامج الأداة التي يستطيع العنصر البشري من خلالها التعامل مع الحاسب بكل مكوناته المادية والبرامج بمختلف أنواعها، وبدون نظام تشغيل يعتبر الحاسب مجموعة من قطع الحديد والبلاستيك وبعبارة أخرى يعتبر حلقة الوصل بين المستخدم والجهاز.

الإصدارات Versions: ويقصد بها النسخ المختلفة من نظام التشغيل التي يتم إصدارها مع كل تطوير يطرأ على الحاسب أو إضافة التعديلات أو التحسينات على النظام، والإصدارات تأخذ أرقاماً للتعبير عنها وكلما زاد الرقم كان معنى ذلك أن الإصدار أحدث، وأول إصدار كان يحمل الرقم 1.00 بينما آخر إصدار كان رقمه 6.22 ونستطيع أن نلاحظ أن الرقم الذي يعبر عن الإصدار مكون من جزئين رقم صحيح وكسر، وظهور إصدار يزيد فيه الكسر فقط يعني حدوث بعض التعديلات في الإصدار السابق، بينما زيادة الرقم الصحيح تعني حدوث تغيير كبير عن الإصدار السابق. بعد ذلك توقفت شركة مايكروسوفت عن طرح إصدارات جديدة ليحل محل هذا النظام نظام تشغيلها المساعد النوافذ Windows ذو الواجهة الرسومية GUI.

**وظائف نظام التشغيل:-** يقوم نظام التشغيل بعدد من الوظائف - المهام الأساسية والتي تعطي الفرصة للمستخدم في أن يتعامل مع الحاسب والبرامج وهي على النحو التالي:

✓ الإشراف على الدخلات والمخرجات بمختلف أنواعها ويمكن أن تتخيله كشرطي مرور يقوم بعملية تنظيم المركبات في ميدان مزدحم، كذلك تنظيم العمل في الذاكرة وهي من المهام الأساسية لنظام التشغيل وذلك من خلال تحميل البرامج وإنشاء ملفات البيانات بمختلف أنواعها كذلك التعامل مع الأقراص من إدخال أي النسخ أو الكتابة عليها أو القراءة منها.

✓ الوسيط بين المستخدم والجهاز Interface:- حيث أن الحاسب لا يستطيع فهم أي لغة راقية فيما عدا اللغة الثنائية الخاصة به، فلا يستطيع توجيه أوامرك بصورة مباشرة إلا من خلال نظام التشغيل والذي يوفر لك معظم الوظائف الرئيسية التي تحتاج إليها عند التعامل مع الحاسب.

### الأنظمة العددية NUMBER SYSTEM

اتفق الناس على إستخدام النظام العشري منذ امد بعيد ، ورغم اختلاف الثقافات من بقعة إلى اخرى على الكرة الأرضية ، إلا أن جميع البشر متفقون على لغة الأعداد وإذا رأى أي شخص نال قسطا من التعليم الرقم 123 مثلا ، فانه يفهمه على أنه :  $100 + 20 + 3$  .

أى أن الخانة الأولى من اليمين للأحاد ، والثانية للعشرات ، والثالثة للمئات . وقد تختلف أشكال الرموز العشرة المستخدمة في **النظام العشري** من بلد إلى آخر فالبعض يستخدم الأرقام العربية ( 0 ، 1 ، 2 ، 3 ، 4 ، 5 ، 6 ، 7 ، 8 ، 9 ) والبعض الآخر يستخدم الأرقام الهندية او غيرها إلا أن الجميع يتفق على استخدام النظام العشري الذي أساسه العشرة ، لكن الحاسوب يختلف عن الإنسان في هذا الأمر . إذا أن انسب نظام له هو **النظام الثنائي** الذي يستخدم الرمز ( 0 ، 1 ) فقط والسبب في ذلك أنه لا يميز إلا حالتين مادتين يتناظران مع هذين الرمزتين .

ومع أن الأمر يبدو غريبا لأول وهلة ، فإن الصفر والواحد كافيان لتمثيل أي عدد مهما كان كبيرا ويسمى النظام العددي في هذه الحالة بالنظام الثنائي binary system .

### ❖ النظام الثماني والسادس عشري :

نستخدم هذين النظامين لا لتمثيل الأعداد داخل الحاسوب ولكن كوسيلة لتبسيط الأعداد الثنائية . **النظام الثماني octal system** : هو النظام العددي الذي يستخدم ثمانية رموز هي

$$\{ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 \}$$

بحيث : العدد في النظام العشري يقابله 10 في نظام الثماني ، والعدد 9 يقابله 11 هكذا

**النظام السادس عشري hexadecimal** فيستخدم 16 رمزا هي :

$$\{ 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F \}$$

لاحظ هنا ضرورة الاستعانة بالحروف الأبجدية كأرقام حيث :

A يعادل 10 في النظام العشري وB يعادل 11 وC يعادل 12 وهكذا

وبصورة عامة فإن العدد  $(a_n a_{n-1} \dots a_1 a_0)$  هو عدد ثمانية يكافئ :

$$a_n 8^n + a_{n-1} 8^{n-1} + \dots + a_1 8^1 + a_0 8^0$$

النظام	الاساس	الرموز المستخدمة
الثنائي	2	0,1
الثماني	8	0,1,2,3,4,5,6,7
العشري	10	0,1,2,3,4,5,6,7,8,9

0,1,2,3,4,5,6,7,8,9,,A,B,C,D,E,F	16	الستة عشر
----------------------------------	----	-----------

### ❖ التحويلات :

غالباً ما نحتاج إلى استخدام أنظمة أعداد مختلفة لذا من الأهمية أن يكون بمقدورنا التحويل من أي نظام إلى نظام آخر

### التحويل من أي نظام إلى النظام العشري :

#### ❖ التحويل من النظام الثنائي إلى النظام العشري :

قد يحدث التباس أحيانا بين اي عدد في النظام العشري والنظام الثنائي على سبيل المثال : هل العدد 101 هو بالنظام العشري أي مائة وواحد او هو بالنظام الثنائي أي خمسة ؟ لذلك نستخدم الدليل السفلي 2 للدلالة على النظام الثنائي والدليل 10 على النظام العشري .

إذا فإن (10) هو مائة وواحد ، أما (101) فهو يقبل خمسة فى النظام العشري وينطق ( واحد صفر واحد ) فى النظام الثنائي .

بصورة عامة فإن العدد الصحيح :  $(a_0 2^0 a_{n-1} \dots a_1 a_0)_2$

حيث  $a_j$  إما صفرًا أو واحدًا ، يمثل عدداً ثنائياً ، وهو يكافئ :

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

فمثلا : أوجد المكافئ العددي  $(1011)_2$  في النظام العشري

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

إذن فغن العدد 1011 في النظام الثنائي يقابله العدد 11 فى النظام العشري .

مثال 2 : أوجد المكافئ العددي  $(1101)_2$  في النظام العشري

$$\begin{aligned} (1101)_2 &= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 \\ &= 1 + 0 + 4 + 8 = (13)_{10} \end{aligned}$$

#### ❖ التحويل من النظام الثماني إلى النظام العشري :

مثال : أوجد المكافئ العددي  $(1002)_8$  في النظام العشري

$$(1002)_8 = 2 \times 8^0 + 0 \times 8^1 + 0 \times 8^2 + 1 \times 8^3 = 2 + 512 = 514$$

#### ❖ التحويل من النظام الستة عشر الى النظام العشري :

مثال : أوجد المكافئ العددي  $(AF)_{16}$  في النظام العشري

$$(AF)_{16} = Fx 16^0 + Ax16^1 = 15 \times 1 + 10 \times 16 = 15 + 160 = 175$$

مثال : اوجد المكافئ العددي  $(A15)_{16}$  في النظام العشري

$$(A15)_{16} = 5x 16^0 + 1x16^1 + Ax16^2 = 5 + 16 + 2560 = 2581$$

### التحويل من النظام العشري إلي أي نظام آخر :

لتحويل العدد  $X$  من النظام العشري إلي النظام ذي الأساس  $B$  نتبع الخطوات التالية :

1- اكتب العدد  $X$  في العمود الأول وأساس النظام الجديد  $B$  في العمود الثاني والباقي  $R$  من القسمة في العمود الثالث كما بالشكل

الباقي $R$	النظام الجديد $B$	العدد $X$

2- اقسم العدد  $X$  على  $B$  واكتب ناتج القسمة أسفل  $X$  في العمود الأول والباقي مقابل ناتج القسمة في العمود الثالث

3- إذا كان ناتج القسمة يساوي صفر توقف عن اجراء القسمة واكتب الجواب من عمود الباقي من اسفل إلى أعلي في سطر من اليسار لليمين وعلى ترتيب

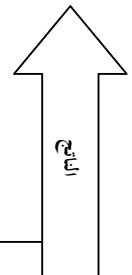
4- إذا كان ناتج القسمة لا يساوي صفرأ تابع اجراءت قسمة الناتج في العمود الأول على الأساس في العمود الثاني واكتب ناتج القسمة السابق والباقي في العمود الثالث

5- نكرر ذلك حتى يكون ناتج القسمة يساوي صفرأ .

مثال : أوجد نظير العدد  $(10)_{10}$  في النظام الثنائي ؟

الحل : تقوم بعملية القسمة التالية :

الباقي	الأساس	العدد
0	2	10
1	2	5



2	2	0
1	2	1

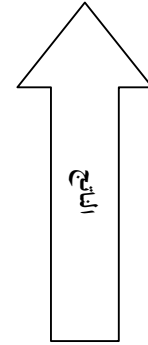
المقابل الثنائي إذن هو ( **1010** ) أي أنه يتكون من البواقي ابتداء من آخر باقي ، كما هو موضح أعلاه باتجاه السهم من أسفل الى أعلى .

**مثال :** المقابل الثنائي للعدد  $(615)_{10}$  هو :

$$(615)_{10} = (1001100111)_2$$

وهذا ينتج من عملية التحليل التالية :-

العدد	الأساس	الباقى
615	2	1
307	2	1
153	2	1
76	2	0
38	2	0
19	2	1
9	2	1
4	2	0
2	2	0
1	2	1

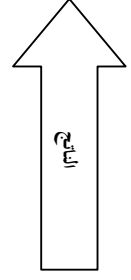


**مثال** اوجد نظير العدد  $(13)_{10}$  في النظام الثنائي ؟

**الحل** تقوم بعملية القسمة التالية :



العدد	الأساس	الباقى
13	2	1
6	2	0
3	2	1
1	2	1
0		

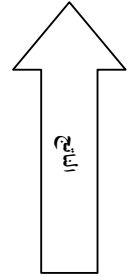


$$(13)_{10} = (1101)_2 \text{ الناتج}$$

مثال أوجد نظير العدد  $(35)_{10}$  في النظام الثنائى ؟

الحل تقوم بعملية القسمة التالية :

العدد	الأساس	الباقى
35	2	1
17	2	1
8	2	0
4	2	0
2	2	0
1	2	1
0	2	

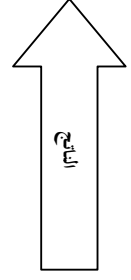


$$(35)_{10} = (100011)_2 \text{ الناتج}$$

مثال أوجد نظير العدد  $(41)_{10}$  في النظام الثماني ؟

الحل تقوم بعملية القسمة التالية :

العدد	الأساس	الباقى
41	8	1
5	8	5
0	8	0



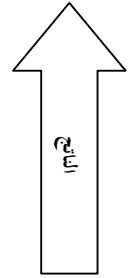
$$(41)_{10} = (51)_8 \text{ الناتج}$$

مثال أوجد نظير العدد  $(125)_{10}$  في النظام الثماني ؟

تقوم بعملية القسمة التالية :

الحل

العدد	الأساس	الباقى
125	8	5
15	8	7
1	8	1
0	8	



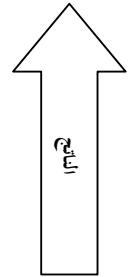
$$(125)_{10} = (175)_8 \text{ الناتج}$$

مثال أوجد نظير العدد  $(125)_{10}$  في النظام السادس عشر ؟

تقوم بعملية القسمة التالية :

الحل

العدد	الأساس	الباقى
125	16	D
7	16	7



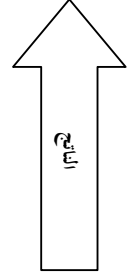
0	16	
---	----	--

**الناتج (D7) = (125)<sub>10</sub>**

مثال أوجد نظير العدد (415)<sub>10</sub> في النظام السادس عشر ؟

الحل تقوم بعملية القسمة التالية :

العدد	الأساس	الباقي
415	16	15 (F)
25	16	9
1	16	1
0	16	



**الناتج (19F) = (125)<sub>10</sub>**

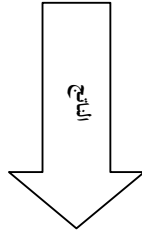
إذا كان العدد العشري كسريا وأقل من الواحد فإن عملية تحويله إلى النظام الثنائي تتم بالطريقة التالية :-

مثال :

أوجد المقابل الثنائي للعدد ( 0.125 )<sub>10</sub> .

الحل :

الجزء الصحيح			
	0.125	*2	
	0	0.25	*2
	0	0.5	*2
	1	1	



أى أن : ( 0.125 )<sub>10</sub> = ( 0.001 )<sub>2</sub>

لاحظ هنا أن اتجاه السهم إلى أسفل بعكس الاتجاه في حالة العدد الصحيح للتحقيق من الناتج نلاحظ أن :

$$( 0.001 ) = 2 = 1/8 = 0.125$$

مثال :

أوجد ( 0.1 ) في النظام الثنائي .

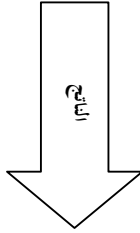
**الحل:**

نلاحظ هنا أن العملية غير منتهية حيث تكررت نفس الأعداد من جديد ، وبالتالي فإن:

$$( 0.1 )_{10} = ( 0.000110011 \dots )_2$$

الجزء الصحيح

	0	0.1	*2
	0	0.2	*2
	0	0.4	*2
	0	0.8	*2
	1	0.6	*2
	1	0.2	*2
	0	0.4	*2
	0	0.8	*2
	1	0.6	*2



**التحويل من النظام الثنائي إلى النظام الثماني :**

نستخدم في النظام الثماني الرموز التالية { 0 ، 1 ، 2 ، 3 ، 4 ، 5 ، 6 ، 7 } فان نظير هذه الأرقام في النظام الثنائي كما هي موضحة بالجدول التالي :

النظام الثماني	النظام الثنائي
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

وتتم عملية التحويل بطريقة استبدال كل رقم إلى نظيره كما هو موضح بالمثل التالي :

**مثال :** أوجد نظير العدد (25)<sub>8</sub> في النظام الثنائي

$$(25)_8 = (010 \ 101)_2$$

**مثال :** أوجد نظير العدد (41)<sub>8</sub> في النظام الثنائي

$$(41)_8 = (100 \ 001)_2$$

**مثال :** أوجد نظير العدد ( 2 ( 101 011 ) في النظام الثماني

$$(101 \ 011)_2 = (53)_8$$

**التحويل من النظام الثنائي إلى النظام السادس عشر :**  
 نستخدم في النظام الثماني الرموز التالية {F,E,D,C,B,A,9,8,7,6,5,4,3,2,1,0} فإن نظير هذه الأرقام في النظام الثنائي كما هي موضحة بالجدول التالي :

النظام السادس عشر	النظام الثنائي
0	0000
1	0010
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

وتتم عملية التحويل بطريقة استبدال كل رقم إلى نظيره كما هو موضح بالمثل التالي :

**مثال :** أوجد نظير العدد  $(25)_{16}$  في النظام الثنائي

$$(25)_{16} = (0010 \ 0101)_2$$

**مثال :** أوجد نظير العدد  $(A25)_{16}$  في النظام الثنائي

$$(A25)_{16} = (1010 \ 0010 \ 0101)_2$$

**مثال :** أوجد نظير العدد  $(AB5)_{16}$  في النظام الثنائي

$$(AB5)_{16} = (1010 \ 1011 \ 0101)_2$$

**مثال :** أوجد نظير العدد  $(3A5)_{16}$  في النظام الثنائي

$$(3A5)_{16} = (0011 \ 1010 \ 0101)_2$$

**مثال :** أوجد نظير العدد  $(20011 \ 1011 \ 0101)_{16}$  في النظام السادس عشر

$$(20011 \ 1011 \ 0101)_{16} = (3B5)_{16}$$

#### ❖ جمع الأعداد الثنائية :

تتم عملية جمع الأعداد الصحيحة الثنائية بطريقة مماثلة للنظام العشري ، مع ملاحظة ان في النظام الثنائي .

$$10=1+1 \quad 1=0+1 \quad 0=0+0$$

وبالتالي فإن عملية جمع عددين ثنائيين تجري كما في المثال التالي :

$$\begin{array}{r} 1001110 \\ + 11010 \\ \hline 1101000 \end{array}$$

ويمكن جمع الأعداد الكسرية أيضا بنفس الطريقة ، فمثلا نجمع 11.011 إلى 10.00 على النحو التالي :

$$\begin{array}{r} 3.375 \\ + 2.75 \\ \hline 6.125 \end{array} \quad \text{يكافى} \quad \begin{array}{r} 11.011 \\ + 10.11 \\ \hline 110.001 \end{array}$$

❖ ضرب الأعداد الثنائية :  
مثال :

$$\begin{array}{r} 10011 \\ \times 101 \\ \hline 10011 \\ 00000 \\ + 10011 \\ \hline 1011111 \end{array}$$

مثال :

$$\begin{array}{r} 1.01 \\ \times 10.1 \\ \hline 101 \\ 000 \\ + 101 \\ \hline 11.001 \end{array}$$

❖ طرح الأعداد الثنائية :

مثال :

اطرح العدد الثنائي 110 من 10011 بطريقة الاستعارة .

الحل :

$$\begin{array}{r} 10011 \\ - 110 \\ \hline 01101 \end{array}$$

تعريف :

مكمل الواحد *1's complement* للعدد الثنائي A هو العدد  $A^{\setminus}$  الذي يحقق :

$$A^{\setminus} + A = 11111111$$

( سنفترض في هذا البند لغرض التبسيط أن الكلمة ذات 8 بت )

مثال :

مكمل الواحد للعدد 00101001 هو 11010110

### تعريف :

مكمل الإثنين  $2s complement$  للعدد A هو  $1 + A$  حيث A مكمل الواحد للعدد A

### مثال :

10010101	=	العدد
01101010	=	مكمل الواحد
01101011	=	مكمل الاثنين

### مثال :

اطرح ( 110 ) بطريقة مكمل الاثنين .

0000	0110	العدد
1111	1001	مكمل الواحد
1111	1010	مكمل الاثنين
0001	0011	يضاف إلى

	1 0000	1101	الناتج
--	--------	------	--------

فائض يهمل

### مثال :

اطرح 110 من 10011 بطريقة مكمل الواحد

0000	0110	العدد
1111	1001	مكمل الواحد
0001	0011	يضاف إلى

	1 0000	1100	ينتج
			أضف إلى الناتج 1
	0000	1101	

### مثال :

اطرح 110110 من 1101 بنظام مكمل الاثنين .

0011	0110	المطروح
1100	1001	مكمل الواحد
1100	1010	مكمل الاثنين
0000	1101	يضاف إلى
1101	0111	الناتج

## المراحل الاساسية لحل المسائل :

حل مسألة هو ايجاد طريقة لأداء عمل معين بداية بالمعطيات حتى الوصول إلى المطلوب سواء باستخدام الحاسب أو أية اداة أخرى ولحل المسائل بالحاسب الآلي يجب أن يمر المبرمج بعدد من المراحل حتى يصل إلى الناتج المرجو وهي :

### 1- التحليل :

من أصعب المراحل التي تواجه المبرمج وهي تحليل وتعريف المسألة المراد حلها وجعلها أكثر وضوحاً وأكثر دقة بتحديد الآتي :

- تحديد عناصر المدخلات أو البيانات التي تستخدم في عملية الإدخال
- تحديد طريقة الوصول من المدخلات إلى المخرجات والتعرف على العمليات أو المعادلات التي تستخدم لحل المسألة
- تحديد طبيعة المخرجات او المتغيرات التي تستخدم في عملية إظهار النتائج وكيفية طباعتها

### 2-التصميم :

وضعها على شكل خطوات متسلسلة ومنطقية و مترابطة وتسمى بالخوارزميات .

### امثله :

مثال : برنامج لمعرفة متوسط درجات 5 طلاب:

### 3-الخوارزميات :

- 1- إبدأ
- 2- إقرأ درجات المواد
- 3- المتوسط = مجموع الدرجات / عدد الطلاب
- 4- اطبع المتوسط
- 5- توقف

مثال:برنامج لإيجاد حاصل جمع عددين :

- 1- ابدأ
- 2- اقرأ العددين A , B
- 3- اجمع  $S = A+B$
- 4- اطبع Sum



\* مثال :-

اكتب الخوارزمية لإدخال قيمتين  $Y, X$  ثم أوجد حاصل جمع ، طرح ، ضرب ، قسمة هاتين القيمتين .  
الحل :

قبل الشروع في كتابة الخوارزمية ، يجب تحليل المسألة أي تحديد عناصر المدخلات  $X, Y$  لهذه المسألة ، أيضا أنواع المخرجات وهي المتغيرات  $D, C, B, A$  .  
وفيما يلي بيان مسلسل للخطوات التي يجب أتباعها .

1- البداية

2- اقرأ قيمة  $X$

3- اقرأ قيمة  $Y$

4- احسب قيمة  $A$  من حاصل جمعهما  $A=X+Y$

5- احسب قيمة  $B$  من حاصل طرحهما  $B=X-Y$

6- احسب قيمة  $C$  من حاصل ضربهما  $C=X*Y$

7- احسب قيمة D من حاصل قسمتهما  $D=X/Y$

8- اطبع العددين  $Y, X$

9- اطبع النتائج  $D,C,B,A$

10- النهاية

لقد تمت كتابة خطوات هذه الخوارزمية خطوة تلي الأخرى من البداية حتى النهاية ، وإذا ما قمنا بإعطاء أي قيمتين وتتبعنا هذه الخطوات سوف نصل إلى النتائج المرجوة .

**\* مثال :-**

اكتب الخوارزمية لإيجاد متوسط ثلاثة الأوزان التالية :-

50 ، 60 ، 70

الحل :

1- البداية 5- اجعل  $D=A+B+C$

2- اجعل  $A = 70$  6- احسب المتوسط  $M=D/3$

3- اجعل  $B = 60$  7- اطبع  $M, C, B, A$

4- اجعل  $C = 50$  8- النهاية

**\* مثال :-**

أوجد قيمة المعادلة

$$X = \frac{A+B}{A-B}$$

بعد إدخال قيمتي  $B, A$

الحل :

1- البداية

2- أدخل قيمة كل من  $B,A$

3- إذا كانت  $B=A$  اذهب إلى الخطوة 9

4- احسب  $X=A+B$

5- احسب  $Y=A-B$

6- احسب  $Z=X/Y$

7- اطبع قيمة Z

8- اذهب إلى الخطوة 10

9- اطبع (( القيمة لانتهائية ))

10- النهاية

في هذه الخوارزمية تم استخدام القرار المنطقي وهو مقارنة كل من المتغيرين B,A فإذا كانا متساويي القيمة عندها يمنع حدوث القسمة لأن حاصل طرحهما في هذه الحالة يساوي صفراً ، وبالتالي طباعة الرسالة المناسبة والتي وضعت بين علامتي التنصيص وهي (( القيمة لا نهائية )) ثم وقف الخوارزمية ، أما إذا كانا غير ذلك ، عندها يحسب حاصل جمعهما ويسند إلى المتغير X أولاً ثم يحسب حاصل طرحهما ويوضع في Y ثانياً وأخيراً يوضع حاصل قسمة X و Y في المتغير Z وبالتالي يطبع هذا المتغير وتأتي نهاية الخوارزمية .

\* مثال :-

اكتب خطوات خوارزمية لحساب وطباعة مربعات الأعداد من 1 إلى 10 .

الحل :

1- البداية

2- اجعل  $N = 0$

3 - اجعل  $N = N + 1$  ( أي زد قيمة N بإضافة 1 إلى قيمتها الحالية )

4 - اجعل  $S = N * N$

5 - اطبع قيمتي S, N

6 - إذا كانت N أصغر من 10 اذهب إلى الخطوة 3

7 - النهاية

هنا المتغير N عبارة عن عداد يستخدم لأغراض التسلسل من 0 إلى 10 تلي ذلك الخطوة الثانية N  $N + 1 =$  وهي تعنى أضف 1 إلى قيمة المتغير N القديمة ثم أسند قيمة حاصل الجمع إلى نفس المتغير N ، أما الخطوة الرابعة فتعنى أسند مربع N في متغير S ثم أطبع المتغيرين ، أتت بعده الخطوة 6 وهي تعنى كرر الخطوات 3 ، 4 ، 5 في حالة أن قيمة N لازالت أقل من 10 .

\* مثال

حساب الحد المطلق للمتغير X ، اكتب خطوات خوارزمية لحساب قيمة Z حسب الحالات التالية :-

$$Z = \begin{cases} X \text{ if } x > 0 \\ 0 \text{ IF } X = 0 \\ -X \text{ IF } X < 0 \end{cases}$$

و ذلك بعد إدخال قيمة المتغير X .

الحل :

1 - البداية

2 - أدخل قيمة x

3 - إذا كان  $X < 0$  اذهب إلى الخطوة 6

4 - اجعل  $X = Z$

5 - اذهب إلى الخطوة 7

6 - اجعل  $Z = -X$

7 - أطبع قيمة Z

8 - النهاية

\* مثال

أكتب الخوارزمية لحساب معدل أعمار NUM من الطلبة في مدرسة تعليمية.

الحل :-

فيما يلي تحديد المتغيرات التي سوف نستخدمها و معنى كل واحدة منها و ذلك لحل هذه المسألة

و هي كما يلي :-

C : تمثل عددا لحصر عدد الطلبة

SUM : مجموع أعمار الطلبة

NUM : عدد الطلبة بالمدرسة

AGE : تمثل عمر كل طالب

AVG : متوسط أعمار الطلبة

و فيما يلي خطوات الخوارزمية المطلوبة : -

- 1 - البداية .
- 2 - أجعل  $C = 0$  .
- 3 - أجعل  $SUM = 0$  .
- 4 - أدخل عدد الطلبة NUM .
- 5 - أدخل العمر AGE .
- 6 - أطبع العمر AGE .
- 7 - أجعل  $SUM = SUM + AGE$  .
- 8 - أضف 1 إلى قيمة العداد C الحالية أي  $C = C + 1$  .
- 9 - إذا كان C لا تساوي NUM أذهب إلى الخطوة 5 .
- 10 - أحسب المتوسط AVG بقسمة المجموع SUM على العدد NUM .
- 11 - أطبع المتوسط AVG .
- 12 - النهاية .

### الشرح : -

حتى تكون النتيجة دائما صحيحة ، ينبغي تخصيص القيمة صفر لكل من المتغيرات المبدئية C, SUM على التوالي والتي يتم تنفيذها مرة واحدة في بداية هذه الخوارزمية ، وللحصول على عمر كل الطلبة AGE إضافتها في المجموع الكلي SUM وتعداد الطلبة ، يتم كل هذا داخل ما يسمى بالدورة ( LOOP ) والتي تضم الخطوات التالية :-

5-خال عمر الطالب AGE

6-طباعة العمر AGE

7-إضافة العمر AGE إلى المجموع SUM

8-زيادة قيمة العداد C الحالية بالرقم 1

ويتم الخروج من هذه الدورة عندما تصبح قيمة العداد C تساوي عدد الطلبة NUM ، عندها يتم حساب المتوسط AVG وبالتالي طباعته وإنهاء الخوارزمية .

**\* مثال :-**

أوجد ناتج تتبع الخوارزمية التالية ، إذا كانت قيم المتغير X كالاتي :

30 ، 90 ، 75 ، 21 ، 50 ، 88

1- البداية

2- اجعل  $S = 0$

3- اجعل  $M = 0$

4- اقرأ قيمة X

5- إذا كانت X اكبر من أو تساوي 75 اذهب إلى الخطوة 8

6- اطبع قيمة X

7- اجعل  $S = S + X$

8- اجعل  $M = M + 1$

9- إذا كانت M أقل من أو تساوي 5 فاذهب إلى الخطوة 4

10- اطبع قيمة S

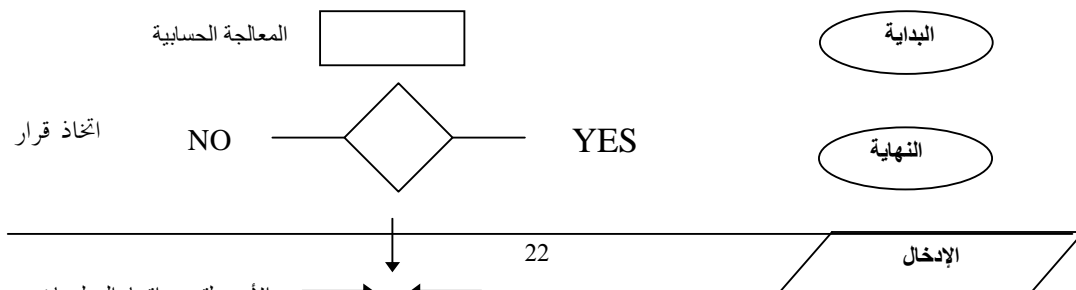
11- النهاية

**3- مخطط سير العمليات:**

يستخرج من الخوارزميات وهو عبارة عن خريطة تستخدم فيها رموز وأشكال هندسية ، وكل شكل من الأشكال يعبر عن نوع الأوامر المستخدمة لحل المسألة وهو يعتبر مرجعاً من مراجع حل المسائل يمكن الرجوع إليها عند غياب مصمم البرنامج لإجراء التعديلات وذلك بمجرد النظر للمخطط .

خرائط سريان *Flow chart* :

تحول هذه الكلمات إلى أشكال تسمى ( خرائط سريان ) Flow chart :

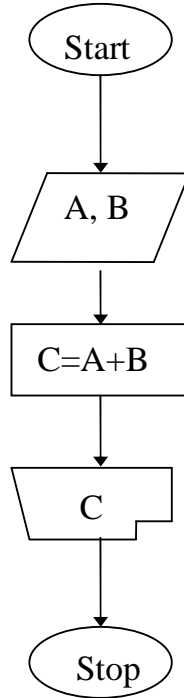


pdfMachine - is a pdf writer that produces quality PDF files with ease!

Get yours now!

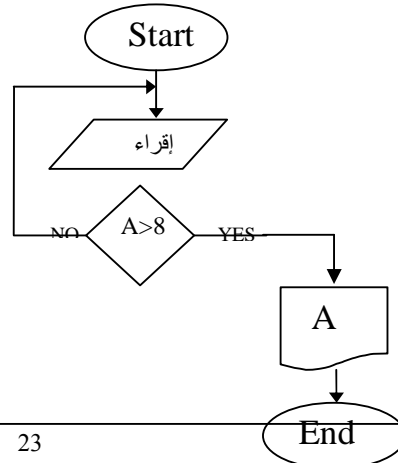
"Thank you very much! I can use Acrobat Distiller or the Acrobat PDFWriter but I consider your product a lot easier to use and much preferable to Adobe's" A.Sarras - USA

مثال / ارسم مخطط سير العمليات ( خريطة السريان ) لبرنامج يقوم بجمع عددين.



مثال : ارسم مخطط سير العمليات ( خريطة السريان ) لبرنامج يقوم بطباعة عدد أكبر من 8

- 1- أبدأ
- 2- اقرأ
- 3- إذا كانت  $A > 8$  أذهب إلى الخطوة 4 وإلا فأذهب إلى خطوة 2
- 4- اطبع A
- 5- توقف



\* مثال (1)

المطلوب رسم مخطط سير العمليات يقرأ العجلة الثابتة A والزمن T ثم يحسب ويطلع المسافة D

حيث

$$D = \frac{1}{2} AT^2$$

و السرعة النهائية V حيث

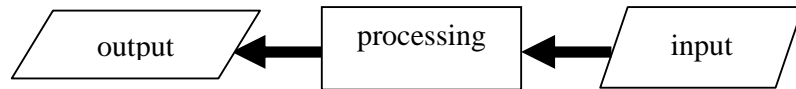
$$V = AT$$

الحل :



الشرح :

في هذا المثال يوجد عدد من الأشكال المختلفة رسمت تحت بعضها من أعلى إلى أسفل الواحد تلو الأخر دون استخدام أية قرارات منطقية .  
في هذه الأشكال تمثل دورة معالجة البيانات Data processing المتعارف عليها والتي تمت مناقشتها في الفصل الأول من هذا الكتاب وهي كالآتي :-



حيث الخطوة (2) تدل على إدخال ( Input ) للمتغيرين  $T, A$  والخطوتان ( 3 ، 4 ) تعتبران معالجة ( Processing ) أو تخصيصا لكل معادلة إلى المتغير المناسب لها بينما الخطوة (5) تعتبر رمز إخراج ( Output ) لكل من المسافة  $D$  والسرعة  $V$

أيضا قد يتم ضم الخطوتين ( 3 , 4 ) المرسومتين لحساب المساحة  $A$  والمحيط  $B$  في شكل واحد كالآتي :-

\* مثال (2)

ارسم مخطط سير العمليات لقراءة أي عدد مع طباعة الجملة " THE NUMBER IS NEGATIVE " إذا كان العدد سالبا ، ويطبع الجملة

THE " إذا كان موجبا ، وإذا كان صفرا يطبع الجملة " " NUMBER IS POSITIVE  
" NUMBER IS ZERO  
الحل :

بعد قراءة العدد ، ينبغي مقارنته مع القيمة 0 حتى يمكن معرفته سالبا كان أم موجبا أو  
يساوي 0 ، وفيما يلي مخطط سير العمليات :-

\* مثال (3)

ارسم مخطط سير العمليات لا يجاد العدد الكبير من بين ثلاثة أعداد

الحل :

في الأمثلة السابقة كنا تستخدم طريقة حل متتابعة أي مجموعة أشكال أو رموز الواحدة تلو الأخرى  
من البداية إلى النهاية وبدون أي قرارات أو تفرعات ، وفي هذا المخطط ولإيجاد أكبر عدد من ثلاثة  
أعداد وجب علينا مقارنة هذه الأعداد بعضها ببعض ، عليه لابد من استخدام أشكال القرارات التي  
سوف نوضحها في المخطط التالي :-

هنا يتم إسناد العدد الأول A إلى المتغير MAX الذي يحتفظ فيه العدد الأكبر ثم استخدام قرارين منطقيين الأول والذي به السؤال هل  $B > MAX$  وهو يعني إذا كان جواب B أكبر من MAX بنعم عندها يتم تغيير قيمة المتغير MAX بإسناد العدد B إليه . أما القرار المنطقي الثاني الذي يحتوي على السؤال هل  $C > MAX$  أي مقارنة العدد الثالث C مع MAX وتسند قيمة C إلى MAX في حالة C أكبر من MAX ، وألا تبقى قيمة MAX كما هي أخيرا تتم طباعة المتغير MAX وبالتالي نهاية المخطط .

\* مثال (4)

صمم مخطط سير العمليات لقراءة ثلاثة إعداد حقيقية A , B , C ثم أوجد الجذور الحقيقية للمعادلة من الدرجة الثانية في الصورة

$$AX^2 + BX + C = 0$$

حيث صيغة الجذر الأول RI هي :

$$R_1 = \frac{-B + \sqrt{4B^2 - AC}}{2A}$$

و صيغة الجذر الثاني R2 هي :

$$R_2 = \frac{-B - \sqrt{4B^2 - AC}}{2A}$$

\* مثال ( 5 )

المطلوب تصميم مخطط سير العمليات لقراءة رقم الطالب و درجته المتحصل عليها في عدد من المواد ، ثم إيجاد و طباعة متوسط درجات الطالب مع رقمه ، و أخيرا طباعة أكبر معدل و إيقاف المخطط عندما يكون رقم الطالب يساوى صفر .

الحل :

لتسهيل و فهم حل هذا المثال ، ينبغي تعريف بعض المتغيرات التي سوف تستخدم في المخطط .

MAX : لحفظ أكبر متوسط في الفصل

SUM : مجموع درجات الطالب

C : يمثل عدداً لحساب عدد الدرجات لكل طالب ( ثلاثة امتحانات في هذا المخطط )

ID : رقم قيد الطالب

G : درجة الطالب

AVG : المتوسط

و فيما يلي مخطط سير العمليات الذي يفني بالغرض : -

**pdfMachine - is a pdf writer that produces quality PDF files with ease!**

**Get yours now!**

"Thank you very much! I can use Acrobat Distiller or the Acrobat PDFWriter but I consider your product a lot easier to use and much preferable to Adobe's" A.Sarras - USA

\* مثال ( 6 )

المطلوب تتبع مخطط سير العمليات في المثال السابق و إيجاد الناتج في حالة إدخال البيانات التالية  
- :

متوسط ورقم الطالب الثاني كالآتي : -

222 26

و يستمر تتبع المخطط حتى الوصول إلى سطر البيانات الأخير الذي فيه رقم الطالب يساوى 0 يصبح

ID = 0

و عليه فان بقية البيانات الموجودة في هذا السطر أو الذي يليه ليس لها تأثير ، و في نفس الوقت  
يتجه التنفيذ إلى السؤال هل  $AVG > MAX$  . و تكون الإجابة بلا في هذه الحالة ، و هنا لا يحدث تغيير  
في قيمة المتغير MAX و هي 60 .

و هكذا يستمر تتبع هذا المخطط بنفس الطريقة حتى الوصول إلى سطر البيانات الأخير الذي فيه  
رقم الطالب يساوى 0 ، حيث يتم طباعة أكبر متوسط و بالتالي نهاية المخطط .

ملاحظة : -

يمكن استخدام هذا المخطط لأي عدد من الطلبة ، و الشرط الوحيد لإنهائه هو إدخال القيمة صفر  
إلى رقم قيد الطالب وإلا فلن ينتهي هذا المخطط .

يمكن الحصول على نتيجة تتبع المخطط السابق بالطريقة المختصرة حيث توضع كل المتغيرات و تتغير  
من حين إلى آخر بحسب الظروف و هي كالآتي : -



هل AVG > MAX ؟	AVG	هل C<3 ؟	G	C	SUM	هل ID = 0 ؟	ID	MAX
				0	0	لا	111	0
		نعم	60	1	60			
		نعم	30	2	90			
نعم	61	لا	93	3	183			61
				0	0	لا	222	
		نعم	16	1	16			
		نعم	45	2	61			
لا	26	لا	23	3	84			
				0	0	لا	333	
		نعم	69	1	69			
		نعم	70	2	139			
نعم	66	لا	65	3	204			66
						نعم	0	

\* مثال ( 7 )

ارسم مخطط سير العمليات لقراءة رقم الدواء NUM و السعر P  
لصيدلية بها عدد من الأدوية ، ثم أوجد الأتي :-

- عدد جميع الأدوية ( M ) بالصيدلية .
- عدد الأدوية ( K ) التي يفوق سعرها 10 دينارات .
- متوسط سعر الأدوية ( AVG ) التي يكون سعرها أقل من أو يساوي 10 دينارات .
- طباعة رقم الدواء NUM و سعره P للأدوية التي أسعارها أعلى من 25 دينارا .
- أوقف المخطط إذا كان رقم الدواء NUM يساوي صفر
- الحل :

حتى يكون حل هذه المسألة واضحاً و سهل المتابعة ، ينبغي كتابة خطوات هذا الحل ( الخوارزمية ) التي قد تكون كالآتي :

- 1 - البداية
- 2 - أجعل المتغيرات SUM ، K ، N ، M تساوي صفرا
- 3 - أقرأ قيمة كل من رقم الدواء NUM و السعر P
- 4 - إذا كان  $NUM = 0$  فإذهب إلى الخطوة 13
- 5 - أضف القيمة 1 إلى العداد M
- 6 - إذا كان  $P < 25$  فاطبع NUM ، P
- 7 - إذا كان  $10 < P$  فإذهب إلى الخطوة 11
- 8 - اجعل  $SUM = SUM + P$  ( حيث SUM تمثل مجموعة أسعار الأدوية )
- 9 - أضف القيمة 1 إلى N ( حيث N عدد الأدوية التي يقل سعرها عن 10 دينارات )
- 10 - اذهب إلى الخطوة 3
- 11 - أضف القيمة 1 إلى K
- 12 - أذهب إلى الخطوة 3
- 13 - إذا كان N لا تساوي 0 أجعل  $AVG = SUM / N$
- 14 - أطلع قيمة كل من K , AVG
- 15 - النهاية

و فيما يلي مخطط سير العمليات لهذه الخوارزمية

في هذا المثال خصصت القيمة الابتدائية صفر لبغض المتغيرات ، يلي ذلك قراءة رقم الدواء NUM و السعر P أيضا تم أستخدم عدد 4 أشكال أو رموز كقرارات منطقية و هي :-

الأول : ليدل على نهاية المخطط في حالة NUM تساوى صفر .

الثاني : ليدل على طباعة رقم الدواء NUM و السعر P الذي يزيد على 25 دينارا .

الثالث : لإيجاد عدد الأدوية K التي يزيد سعرها عن 10 دينارات في حالة الإيجاب ، و مجموع الأسعار SUM و عددهم N في حالة النفي للحصول على المتوسط AVG .

الرابع : حساب المتوسط في حالة N لا تساوى صفر .

مع افتراض أن هناك أكثر من دواء سعره أعلى من 10 دينارات .

#### 4- البرمجة :

الحاسب الآلي لا يفهم الخوارزميات أو مخطط سير العمليات لذا يجب أن تكتب كمجموعة من الأوامر بإحدى اللغات البرمجية ويتم إدخالها للحاسب ثم تتحول هذه الأوامر إلى برنامج صالح للاستخدام

#### 5- اختبار البرنامج :

تنفيذ واختبار البرنامج عن طريق عينة من البيانات للتأكد من صحة البرنامج قبل تنفيذه على بيانات حقيقية حتى يعطي البرنامج النتائج الصحيحة .

### العناصر المكونة لأي لغة.

- 1- الرموز: (حروف - أرقام - أشكال خاصة (+، -، !، \*، .....))
- 1- الكلمات المحجوزة.
- 2- الأوامر (Instructions).

أ - أوامر إدخال البيانات.

ب - أوامر إخراج البيانات.

ج - أوامر تحكم في سريان البرنامج.

د - أوامر تعريفية.

4- قواعد اللغة: وهي مجموعة من القوانين والقواعد والقيود التي يجب أن تنتج عن البرمجة.

### أساسيات لغة البيسك.

#### (Fundamentals of Basic Language)

#### - عناصر لغة البيسك:

تتكون لغة البيسك من مجموعة من الحروف والثوابت والمتغيرات والتعبيرات تسمى عناصر البيسك.

#### 1- عناصر حروف البيسك :

يعتبر الحرف هو أصغر عنصر في لغة البيسك وتتضمن لغة البيسك ثلاث مجموعات رئيسية من الحروف هي :

. الأرقام ( 0- 9 )

. الحروف الأبجدية ( A - Z )

. الحروف الخاصة (+، -، !، #، .....)

## 2- ثوابت البيسك:

هي القيم الثابتة التي لا تتغير أثناء تشغيل البرنامج .  
وهي نوعان:

أ- ثوابت عددية . ب- ثوابت حرفية.

أ- الثوابت العددية:

- الأعداد الصحيحة.

- الأعداد الحقيقية ( الكسور العشرية - الصور الأسية ) .

ب - الثوابت الحرفية: ي تتابع من الحروف الأبجدية محصورة بين علامتي تنصيص .

## 3- المتغيرات:

هي أسماء البيانات وهي الأوعية التي تحفظ فيها البيانات في الذاكرة وهي

أ - المتغيرات الحرفية "ALI"

ب- المتغيرات العددية:

1 - المتغيرات الصحيحة:

هي موضع التخزين بذاكرة الكمبيوتر المخصصة لتخزين بيانات عددية صحيحة متغيرة وتتميز المتغيرات الصحيحة بوضع علامة ( % ) في نهاية المتغير .

A% , B% , C%

2- المتغيرات الحقيقية:

هي موضع التخزين بذاكرة الكمبيوتر المخصصة لتخزين بيانات عددية حقيقية متغيرة .

أ- متغيرات حقيقية أحادية الدقة وتتضمن 7 أرقام A!, B!, C!, .....

ب- متغيرات حقيقية متضاعفة الدقة تتضمن 16 رقم . A#, B#, C#, .....

## 4- تعبيرات البيسك:

تنقسم إلى قسمين: تعبيرات عددية - تعبيرات حرفية

- تكوين التعبيرات العددية .

يتم بناء التعبيرات العددية باستخدام الثوابت العددية والمتغيرات العددية المتصلة فيها بينهما بالأقواس والمعاملات الحسابية .

### المعاملات الحسابية:

التعبير الجبري	التعبير البيسك	العمليات الحسابية
A + B	A + B	الجمع
A - B	A - B	الطرح
A * B	A . B	الضرب
$\frac{A}{B}$	A / B	القسمة
A <sup>b</sup>	A ^ B	الأس

أولوية العمليات الحسابية:

1- ما بداخل الأقواس

2- الأس

3- الضرب والقسمة طبقاً لأولوية ظهورها.

4- الجمع والطرح طبقاً لأولوية ظهورها.

مثلاً:

$$\frac{((A+B) * (A-B))}{C}$$

### أوامر لغة البيسك :

1- أمر الملاحظة: REM statement.

REM This program is to compute phase angle.

2- أمر التخصيص: LET variable = constant

```
10 LET X=5
```

```
20 LET N$ = "ALI"
```

يتم تخزين قيمة الثابت ( العددي - الحرفي) الموجود إلى يمين علامة = في المتغير الموجود إلى يسار علامة =

### معالجة المدخلات والمخرجات

(Input & output processing)

1- أمر الطباعة: Print

يستخدم أمر الطباعة في إظهار بيانات المخرجات على شاشة العرض أو نسخ ورقية على الطباعة.

```
Let A=5
Let A$ = "Ali"
print "....."
print A, A$
print A; A#
print
print "Thank You"
```

2- استخدام دالة المسافات ( TAB(x) :

تستخدم دالة المسافة مع جملة الطباعة لتعيين موضع الطباعة.

```
Print Tab (10); "Garyounis"
Print Tab (25); "University"
Print Tab (10); "Garyounis"
print Tab (25); "University"
```

أمر الاستخدام - الطباعة (print using Statement) :

```
Let A$ = "# # ## #, # # # , # # # . # # "
```

```
Let x = 1234.56
```

```
Print using A$ ; x, x, x
```

```
Print using "# # # #"; x
```

```
Print using "# , # # #" : x
```

```
Print using "# , # # # . # #" ; x
```

Run

```
1234 1,235 1,234.56
```

```
1235
```

```
1.235
```

```
1,235.56
```

```
Let w =15673088.209
```

```
Let A=17.667
```

```
Let B=-5,38
```

```
Let C= 40
```

```
Print using "# # . # #"; A, B, C
```

```
Print using "# # # # # # # , . # #"; W #
```

Run :

```
17.67 -5.38 40.00
```

```
15,673, 688.21
```

```
Print using "$ # # # # . # #"; 21.58
```

```
Run : $ 21 . 58
```

```
Print using "$$ # # # # # # #"; 21.58
```

```
Run : $ 21.58
```

```
Print using "$$ # # # # # , # # ; 2192.963
```

```
$ 2,192.96
```



### - إدخال البيانات أثناء تنفيذ البرنامج INPUT:

يستخدم H أمر الإدخال في استقبال البيانات المغذاة بواسطة لوحة المفاتيح أثناء تشغيل البرامج وتخزينها في الذاكرة.

**Input variable, variable,....**

- (1) Input x, y,z  
Print x+ y + z
- (2) Print “ Enter Your name “  
Input N\$  
Print “Your name is” ; N\$
- (3) Input “ what is your name “ ; N\$  
Print N\$

### - تخزين البيانات داخل البرنامج READ / DATA :

تتميز هذه النوعية من البيانات بأنها ثابتة ولا تحتاج إلى تعديلها أو تحديثها ولا تتغير من تنفيذ إلى آخر.

**Read** variables, .....

**Data** Constant, .....

- (1) Read A,B,C  
Print A+B+C  
Data 11,2,31

```
(2) Data Ahmed, Ali
      Read Name1 $, Name2 $
      Print Name1$ ; Name2 $
```

### - أمر إعادة التخزين (Restore Statement):

يستخدم الأمر إعادة التخزين في إعادة التحكم إلى بداية قائمة البيانات.

```
READ A, B, C,
Restore
Read X, Y
Data -1, 4, 16
Print A; B; C ; x ; Y
```

### عملية الانتقال والتفرع

(Transferring and Branching Operations)

### \* أوامر الانتقال غير المشروط:

للتحكم في موضع معين إلى موضع آخر بالبرنامج دون أي شرط . والأمر المستخدم في تنفيذ، الانتقال غير المشروط هو أمر GOTO

**Goto Line number.**

```
print "ALi"
Goto 40
print "same"
40 print " Hello"
```

## أوامر استخدام القرار للتحكم فى البرنامج::

### -1 التفرع المشروط للمسار :IF - Then

IF Conditional-Expression (Then Goto \ Then \ Goto Line Number)

إذا ..... حينئذ

Input x

If X= 1 then 40

End

40 print “ yes “

### \* إذا - حينئذ - وإلا

IF Conditional-Expression Then (Statement 1 / Line No.) Else (Statement 2 / Line No.)

input x

If x= 5 then print “ x = 5” Else print “ x<> 5”

### المعاملات المنطقية:

العامل المنطقي و ( AND )

العامل المنطقي أو (OR)

العامل المنطقي النفي(Not)

10 Read x

IF x>2 And x <8 then print x Else End

Goto 10

Data 1,-7 8 , 15, 3, 6

### \* أمر اذهب - إلى المتعدد ( On Goto )

يمكن التفرع فى لغة البيسك من موضوع معين بالبرنامج إلى مواضع مختلفة ومتعددة طبقاً لمجموعة محددة من الشروط باستخدام أمر اذهب - إلى المتعدد .

```
On numeric Expression Goto Ln1,Ln2, .....
```

```
Read x
```

```
On x Goto 40 , 30
```

```
30 Print x^2 : End
```

```
40 Goto 10
```

```
Data 1,2
```

### الحلقات المتكررة والبرامج الفرعية (Loops and Subprograms)

#### -الحلقات باستخدام For and Next statement

**For** variable = Numeric -Expression 1 To Numeric-Expression 2 Step ( Numeric Expressions)

**Next** variable

المعنى. بدء آمن القيمة الابتدائية للمتغير وحتى القيمة النهائية بخطوة مقدارها .....

```
(1) For I = 1 to 20 step 2
```

```
Print I;
```

```
Next I
```

```
(2) For I = 10 to 1 step -1
```

```
print I;
```

```
next I
```

#### - الحلقات باستخدام (While and wend statement) :

**while** Condition

**Wend**

```
I=1
```

```
while I<=100  
S = S + I  
I= I+ 1  
wend  
print "The sum is"; S  
End
```

### البرامج الفرعية (Subprograms)

**Gosub Line number**

**Return**

اذهب إلى تنفيذ البرنامج الفرعي بالسطر رقم ..... وفى نهاية كل برنامج فرعي يجب استخدام العبارة Return

```
Input x  
Gosub 100  
Print y,w  
End  
100 Y = x ^ 2  
w= x^3 + y  
Return
```

### البرامج الفرعية المشروطة المتعددة (ON - Gosub):

تستخدم هذه العبارة عندما يكون مطلوباً من البرنامج الذهاب إلى برامج فرعية مختلفة حسب قيمة تعبير ما .

**On Numeric Expression Gosub Line numbers**

بحسب قيمة التعبير اذهب إلى البرامج الفرعية أرقام ..... و ..... و .....

```
10 Input x  
On X GoSub 100, 200, 40.  
Goto 10
```

```
40   End
100  Input A,B
      Print A+B
      Return
200  Input A,B
      Print A*B
      Return
```

### معالجة المتغيرات ذات الأبعاد

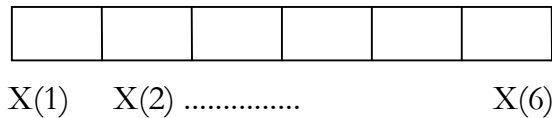
#### (Dimensional Variables Processing)

- **المصفوفات ذات البعد الواحد One-Dimensional Arrays :** تعتبر مصفوفة البعد الواحد قائمة List وتسمى أيضا متجه من البيانات من نفس النوع (بيانات عددية او حرفية )
- يستخدم أمر Dim في الإعلان عن مصفوفة ذات البعد الواحد ( العددية أو الحرفية ) ويأخذ أمر البعد الشكل الآتي:

Ln Dim variable (Limit ),.....

Dim x(6)

Array X



- إدخال / إخراج المصفوفات باستخدام أمر القراءة :

Dim A (10)

For I = 1 to 10

Read A (I)

Next I

Data 3,5,8,7,6,18,91,19,10

**باستخدام أمر الإدخال :**

```
Dim A (15) , B (15)
For I = 1 to 15
Input A(I) ,B(I)
Next I
```

### إخراج المصفوفات باستخدام أمر الطباعة:

```
Dim A(5)
print "Days", "Sales"
For I= 1 to 5
Read A(I)
Data 101,200, 50.5, 35-5, 16
Next I
For I =1 to 5
Print I, A( I)
Next I
```

### - المصفوفات ذات البعدين Two-Dimensional Array :

- تتكون من مجموعة افقية الصفوف Rows ومجموعة رأسية من الأعمدة Coulmns ويستخدم أمر البعد Dim Statement في الإعلان عن مصفوفة البعدين ( العددية او الحرفية ) ويأخذ أمر البعد الشكل الآتي:

**Dim variable 1 (n1,m1 ), variabl2 (n2,m2 ),.....**

```
Dim A(5,3)
For I=1 to 5
```

```
For J=1 to 3  
Input A( I,J )  
Next I  
Next J  
End
```

- تطبيقات على المصفوفات ذات البعد الواحد والبعدين .

### الدوال (Functions)

#### - الدوال العددية :

1- القيمة العددية المطلقة لـ  $x$

$$Y = \text{ABS} (x)$$

Print ABS (-5)

2- جيب تمام الزاوية وجيب الزاوية

$$Y = \text{Cos} (x)$$

$$Y = \text{Sin} (X)$$

3- الدالة الأسية

$$\text{Exp} ( X )$$

4- اللوغاريتم الطبيعي لـ  $x$

$$\text{Log} (X) , X > 0$$

5- دالة الأعداد العشوائية

$$Y = \text{RND} \ 0 < Y < 1$$

6- الجذر التربيعي لـ  $x$

$$Y = \text{SQR} ( X )$$

7- باقي القسمة

MOD



$$\text{SGN}(x) = \begin{cases} 1 & x > 1 \\ 0 & X = 0 \\ -1 & X < 1 \end{cases}$$

9- دالة تقرب الأعداد الحقيقية (x) CINT

10 لحذف الأرقام العشرية من العدد X . (x) FIX

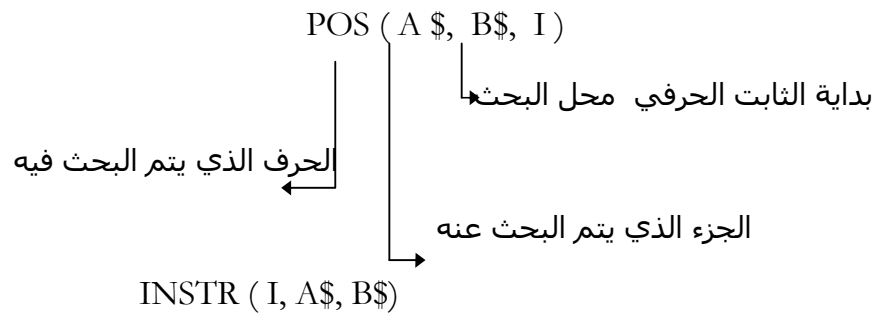
11- أكبر عدد صحيح أقل من أو يساوي X Int (x)

- تطبيقات على الدوال العديدية .

### -الدوال الحرفية :

الوظيفة	الدالة
للبحث عن حرف أو أكثر داخل ثابت حرفي	INSTR , POS
لاستخراج جزء من ثابت حرفي	MID \$ - RIGHT\$- SEG \$ - LEFT\$
لتكوين ثابت حرفي بأي طول مكون من لبنة متكررة	STRING \$
لتكوين ثابت حرفي يحتوي على مسافات خالية	SPACE \$
لتحديد عدد اللبنة المكونة لثابت حرفي ( طوله )	LEN

البحث عن حرف في كلمة INSTR / POS :



A\$ = "ABCDERGHJKLMNO"

B\$ = "DEFG"

X% = POS ( A \$, B\$, 1%)

Y% = INSTR ( 1%, A\$, B\$ )

Print x%, Y%

End

الجزء المطلوب بدءاً من  
الحرف الرابع

4 4

\*- طبع جزء من الكلمة SEG \$ :

SEG \$ ( A\$, I, J)

آخر حرف ← بداية الجزء → الكلمة المراد طباعة جزء منها

10 Print Seg\$ ( "ABCDEFG", 3,5)

RUN CDE

\* استخلاص حروف من وسط لمجموعة حروف MID\$ :

MID \$ ( A\$, I, J)

طول الجزء المطلوب طباعته → بداية الجزء المراد طباعته ← طول الكلمة المراد طباعة جزء منها

10 print MID\$ ( "ABCDEFG", 3,5 )

Run CDEFG

\* LEFT \$ تعطي الجزء الأيسر لمجموعة الحروف :

LEFT \$ ( A\$, I)

← عدد اللينات المطلوب استخراجها من جهة اليسار

```
10 print Left $ ("ABCDEFG", 3)
```

ABC

RIGHT \$ تعطي الجزء الأيمن لمجموعة حروف:

```
RIGHT $ (A$, I)
```

← ترتيب اول حرف فى الجزء المطلوب طباعته من جهة اليسار

```
(1) Print Right $ ("ABCDEFG", 2)
```

FG

```
(2) y$ = Right $ (ENVELOPES, 6)
```

```
Print y $
```

ELOPES

تكوين ثابت حرفي من لبنة متكررة \$ STRING :

```
STRING $ (I1, I2)
```

عدد صحيح يمثل الكود الأسكي للبنية المطلوبة ← طول الثابت الحرفي

```
Print String $ (5, 65)
```

A A A A A

\* دالة المسافة الخالية ( I ) :SPACE\$

← عدد المسافات الخالية

```
print "ABC"; space$ (4) "Computer"
```

ABC 4 ← computer

\*- دالة تعيين عدد الحروف : LEN (A\$)

← الكلمة المراد حساب عدد حروفها

```
(1) LEN ( Hello )
```

RUN.

5

(2) Input X\$

Print LEN(X\$)

RUN.

? Hello There

11

### \*- الدوال المبتكرة:

**DEF FN name ( variable ) = numeric exprescion**

DEF FN Y (x) = 3 \* x ^ 2 - 2 \* x + 4

E= 3 : F=2

D = FN Y(E+f)

M= FN Y ( 4 )

W= FN Y(E)

Print D; M ; W

Run

69

44

25

### أوامر التخزين وتحميل البرنامج

• الأمر Save :

يستخدم لتخزين البرنامج المخزن اما على القرص المرن او القرص الصلب

### Save "File name"

• الامر Load :

يستخدم لتحميل البرنامج المخزن على القرص إلى الذاكرة الرئيسية

### Load "File name"

### الأمر Locate

تنقسم الشاشة إلى 24 سطراً و 80 عموداً فإذا أراد المبرمج تحديد النقطة (x,y) أي السطر رقم X والعمود رقم Y فإنه يستعمل الأمر :

### Locate X,Y

مثال :

Locate 10,18

Print "Mohamed"

البرنامج سوف يقوم بطباعة الكلمة Mohamed في السطر 10 ابتداء من العمود 18

الرسوم البيانية :

### أمر الشاشة Screen:

### Screen n

حيث n نمط الشاشة وهناك 3 أنماط للشاشة هي :

- Screen 0 لا يسمح بإظهار الرسوم البيانية على الشاشة
- Screen 1 من 200 صف من البيكسل وكل منها بعض 320 بيكسل يسمح بإظهار الرسوم البيانية على الشاشة و تتركب الشاشة في هذا النمط
- Screen 2 من 200 صف من البيكسل وكل منها بعض 640 بيكسل يسمح بإظهار الرسوم البيانية على الشاشة و تتركب الشاشة في هذا النمط

### أمر اللون Color:

### Color Background , Palette

رقم الخلفية ويأخذ قيمة من 0 إلى 15

## Background Palette

مجموعة الألوان المراد وضعها على الخلفية لتلوين الصورة الأمامية

مثال :

Color 7,0

رسم النقط :

$Pset(x,y),color$

حيث  $(x,y)$  تمثل إحداثي النقطة .

مثال :

Pset (250,1),1

رسم الخطوط :

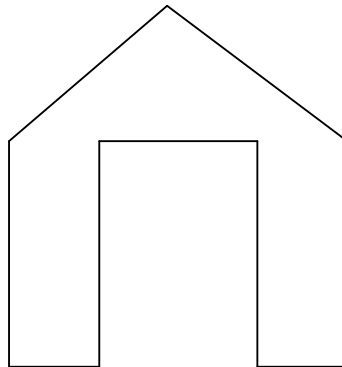
**Line  $(x1,y1) - (x2,y2),color$**

ارسم الخط من النقطة  $(x1,y1)$  إلى النقطة  $(x2,y2)$  .

مثال :

**Line  $(0,0)-(219,199),1$**

تمرين : قم برسم الشكل التالي



## رسم الدائرة :

**Circle (x,y) R,C,R1,R2**

حيث

- R نصف القطر
- C اللون
- R1 بداية نصف القطر
- R2 نهاية نصف القطر

امثلة :

Circle (0,0),10



Circle (0,0),10,0,1



Circle (0,0),10,1,0



Circle (0,0),10,-1,0



## معالجة ملفات البيانات

(Data Files Processing)

### ماهي الملفات ؟

للوصل المغناطيسي ( الأقراص ) فائدتان :

- حفظ البرامج واسترجاعها عند الحاجة إليها .
  - هي حفظ البيانات التي يمكن استخدامها بواسطة البرامج المختلفة عند الحاجة إليها .
- ويطلق اسم الملف على مجموعة ما من البيانات المخزنة في الوصل المغناطيسي .

### ماهي ملفات البيانات ؟

ملفات البيانات هو تجميع لمجموعات البيانات المرتبطة والمخزنة في وسط تخزين ثانوي ( القرص المرن ) وتكون مفصولة عن البرنامج الذي يستخدمها .  
وتسجل البيانات على أوساط التخزين الثانوية في شكل ملفات بيانات يحتوي على مجموعة من سجلات البيانات ويتكون كل سجل من هذه السجلات من مفردات البيانات .

### مزايا استخدام ملفات البيانات :

- يمكن استخدام ملفات البيانات عن طريق أكثر من برنامج
  - يمكن لملفات البيانات تخزين مخرجات البرنامج لاستخدامها كمدخل لبرنامج اخر
  - يمكن إنشاء ملفات البيانات وتحديثها بواسطة البرنامج بسهولة.
  - يمكن تداول العديد من ملفات البيانات بواسطة برنامج واحد
- ويتضمن نظام البيسك مجموعة من الأوامر التي تتيح للمستخدم إجراء العمليات التالية :

- إنشاء ملفات البيانات
- تعريف ملفات البيانات
- فتح ملفات البيانات
- قراءة البيانات من الملف
- كتابة البيانات على الملف
- اختبار نهاية الملف
- غلق ملفات البيانات

### تنظيم الملفات :

هو طريقة ترتيب الملفات على أوساط التخزين الثانوي وهناك نوعان رئيسيان لتنظيم الملفات هما :

- 1- **الملفات التتابعية** : تتم معالجة السجلات في الملفات التتابعية بالترتيب المخزن به في الملف فلكي تتم معالجة السجل رقم 10 يلزم المرور على السجلات السابقة له من 1 حتى 9 .
- 2- **الملفات العشوائية** : لاتوجد علاقة تتابع لمعالجة السجلات في الملفات العشوائية بالترتيب الذي تخزن به السجلات في الملفات بمعنى إذا أردنا معالجة السجل رقم 10 فلا يستلزم المرور على السجلات السابقة له في الترتيب بل يتم تداوله مباشرة.

### اسماء الملفات :

يتكون اسم الملف من حرف الى 8 أحرف متبوعة بنقطة وامتداد مكون من 3 أحرف يعبر عن نوع الملف على سبيل المثال :

- (.Txt) لتمثيل ملفات النصوص
- (.Dat) لتمثيل ملفات البيانات



- (.Sav) لتمثيل النسخ الاحتياطي.
- (.Lis) لتمثيل الملفات التي تحتوي على التقارير.
- (.Bas) لتمثيل برامج البيسك.
- (.TBL) لتمثيل الملفات التي تحتوي على جداول.

### أ- الملف التساعي (Creating Sequential File)

#### **1- أمر فتح وغلق الملفات التساعية:**

يقوم أمر فتح الملف بتسمية الملفات وتحديد نوع Mode الملف وتخصيص رقم الملف.

#### **Open "mode" , #filename, "filespec"**

أمر فتح الملفات

Mode:

- I لإدخال البيانات
- O لإخراج البيانات

Filename : رقم الملف من 1-15

Filespec : اسم الملف ووصفه

#### **Close #filename, #filename,....., #filename**

أمر غلق الملفات

Open "O",#1," Student.dat"

أمثله:

Open "I",#4," Degree.dat "

Open "O",#6," Age.dat"

Close #1

Close #4,#6

**مثال:**

10 Open "O", # 1 " Data "

20 Input "name"; N\$

```
30 IF N# = "ENd" Then close : End
40 Input "Grade"; G
50 Go to 20
```

## 2- أمر الطباعة ( print # ):

يقوم أمر الطباعة بكتابة البيانات على الملف. بمعنى أنه يتم نقل البيانات من منطقة التخزين الوسيطة بذاكرة الحاسب الآلي إلى وسط التخزين الثانوي للملف .

***print # File-number, List of expressions***

اطبع في الملف رقم ..... التعبيرات الآتية .

مثال :

```
Print #1 ,Tab(10);"The answer is";A
```

## 3- استرجاع المتغيرات :

***Input*** تتم قراءة البيانات من الملف التتابعي باستخدام الأمر

***Input # File-number, variadle-list***

استرجاع المتغيرات الآتية من الملف رقم .....

مثال :

```
Input #3 ,, Hours ,Rate
```

## 4- غلق الملف:

***IF EOF ( file-number ) then close***

إذا وصلت لنهاية الملف المفتوح بالقناة رقم .... اغلق الملف .

**مثال :**

```
10  Open "I", # 1, "Data"  
20  Print Tab(4); "Name" ; Tab(25); "Grade" .  
30  IF EOF (1) Then close : End  
40  Input # 1, N$ , G  
50  Print N$ ; Tab (27) ; G  
60  Goto 30
```

**ب- الملف العشوائي (Random File)**

Open "Filename" , As #filename, record length Hمر فتح الملفات

Open "Invlist" As #1 ,55

Close #filename, #filename,....., #filename Hمر غلق الملفات

أمثله:

```
Close #9  
Close #2,#3
```

**يوجد اختلافان جوهريان لفتح الملف التتابعي وفتح الملف العشوائي :**

- في الملفات التتابعية لابد من تحديد نوع المعالجة ( إدخال أم إخراج ) بينما في الملفات العشوائية دائما لعمليتي الإدخال والإخراج
- في الملفات العشوائية يكون من الضروري الإشارة إلى طول السجل ( عدد الحروف بالسجل ) في أمر الفتح وبتراوح طول السجل من 1- 32767 حرفاً.

**أمر الجلب والوضع GET , PUT**

• يستخدم أمر **GET** لقراءة ونقل سجل من الملف العشوائي

**GET # File number, Record number**

مثال :

Get #1 ,75  
Get #2 , num

• يستخدم أمر **PUT** في كتابة السجل من منطقة التخزين على الملف العشوائي

**GET # File number, Record number**

مثال :

Put #1 ,27  
Put #2 , K

• **دالة الموقع LOF** تعطي دالة طول الملف عدد الحروف بالملف وتستخدم لتعيين عدد السجلات بالملف وذلك بقسمة إجمالي عدد حروف الملف على طول السجل

**LOF( File number)**

مثال :

Let Length =LOF(1) / 64

## مشروع

1- اكتب برنامجاً يقوم بإدخال رقم واسم عدد من الطلبة مع المبلغ المطلوب من الطالب مقابل تسجيله وإخراج تقرير يضم الطلبة الذين لم يقوموا بدفع اشتراكاتهم في تسجيل المواد بالجامعة في الملف الأول وأرقام الطلبة والمواد التي تم تسجيلها في ملف اخر .

2- اكتب برنامجاً ينتج عنه ملف بيانات يضم بيانات تخص مصرف الدم بحيث تتم قراءة البيانات عن طريق دالة فرعية وتخزينها في ملف رئيسي خاص بالبيانات التالية :

- رقم الشخص ID
- اسم الشخص Name
- العنوان Address

- الهاتف Telephone
- العمر Age
- فصيلة الدم Blood Type

المطلوب :

- طباعة رقم الشخص و الاسم و رقم الهاتف للأشخاص الذين لديهم فصيلة الدم A موجبه .
- تخزين الاسم والعنوان والعمر للأشخاص الذين لديهم فصيلة الدم O سالبه أو A سالبة مع عددهم في ملف آخر .
- إنشاء ملف ثالث تخزن فيه كل التعديلات الموجودة في الملف الرئيسي .
- إنتاج قائمة بكل الأسماء وأرقام الهواتف للأشخاص الذين تكون أعمارهم أقل من 18 سنة ولديهم فصيلة B .

## تمارين عامة

1 - صحح الخطأ في جمل البيسك الآتية

- 1:- PRINT TEB(50) "OK"
- 2:- IF I > 50 GOTO 40
- 3:- PRINT (AIL),
- 4:- DATE A,B,C
- 5:-  $X_1 + X_2 + X_3 = S$
- 6:- LET C\$ = (A + B) / 2
- 7:- DAM (15)

## 2 - أوجد نتيجة البرامج الآتية:

(ب)

```
10 READ A,B
20 DATA 5,4,7,8,3,8
30 IF A = B THEN 70
40 PRINT A,B, A*B
50 PRINT
60 GOTO 10
```

```
10 READ A$,X,Y
20 DATA "GOOD LUCK",7,5
30 IF X > Y ^2 THEN PRINT A$ ELSE PRINT X
  - Y ^2
```

(أ)

3 - باستخدام WEND WHILE قم بكتابة برنامج يطبع جدول ضرب العدد (7) بهذا الشكل :

```
  1*7
  2*7
-----
 10*7
```

4 - اكتب برنامجاً يحجز مصفوفة ذات بعدين و لتكن  $X(3,3)$  ثم يقوم بإدخال أو قراءة عناصر هذه المصفوفة و لكن عند الطباعة يطبع العناصر التي فوق القطر الرئيسي فقط و يملأ بقية المواقع بالرمز (\*).

```
  *  1  3
  *  *  2
  *  *  *
```

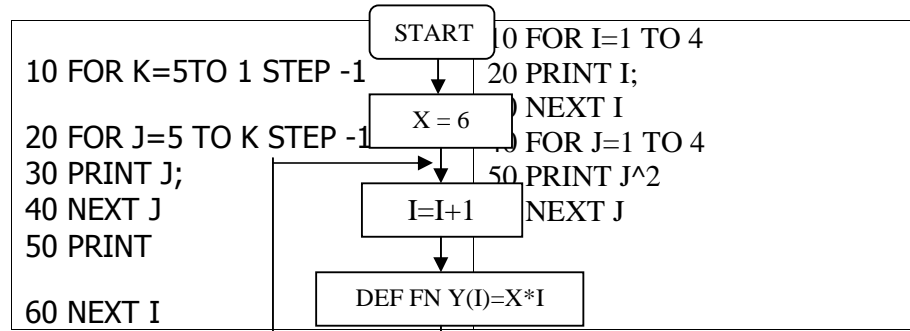
5 - اكتب برنامجاً يقوم بحجز مصفوفة ذات بعد واحد ، واحدة للاسم و أخرى للدرجة لـ 10 طلبة ، ثم يقوم بالبحث عن الدرجة التي أعلى من 85 إذا وجدها يطبع اسم الطالب و يطبع أمام اسم الطالب الجملة " V.Good"

6 - اكتب برنامجاً يقوم بحجز و تخزين المصفوفة  $A(2,2)$  ثم يكون المصفوفة B و هي عبارة عن مقلوب المصفوفة A ؟

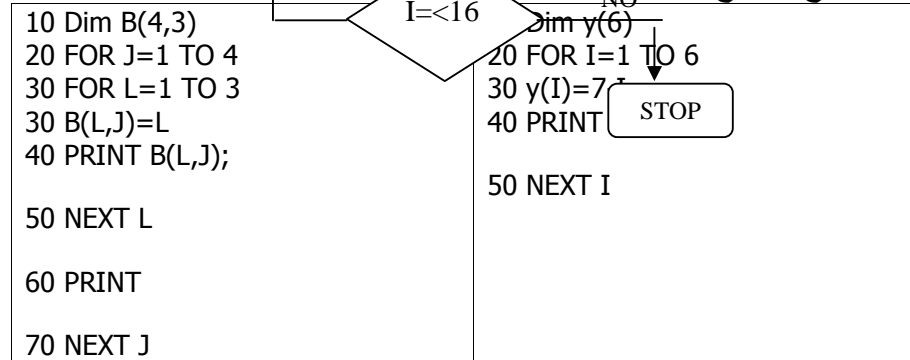
7 : اكتب برنامجاً لقراءة المصفوفة A,B ثم إيجاد المصفوفة C حيث  $C=A+9B$

8 : اكتب برنامج يقوم بتخزين القيم التالية في المصفوفة وطباعتها بصورة عكسية  
4,5,6,7,8,9

9 : تتبع ناتج البرامج التالية :



10 : اكتب برنامجاً يقوم بطباعة من رتبة 3x3 بدون استخدام تعليمة Input شوائية أعداده فردية محصورة بين 10-90



11 : أوجد ناتج البرامج التالية :

12 : اطبع ناتج خريطة السريان ثم اكتب البرنامج بلغة البيسك :

13 : اكتب برنامجاً لترتيب مجموعة من الأعداد ترتيباً تنازلياً

14 : اكتب برنامجاً يقوم بإدخال اسم الطالب ورقمه الدراسي ودرجات الامتحان الأول والثاني والثالث وحساب معدله وفقاً للآتي :  
المعدل = الدرجة / 4

F	اقل من 1
D	1-1.5
C	1.5-2
CC	2-2.5
B	2.5-3
A	3-3.5
AA	3.5-4

16 : ارسم خريطة السريان وذلك للبرنامج التالي:

- البرنامج يقوم بحساب مجموع أربعة أعداد إذا كانت كلمة المرور 44
- البرنامج يقوم بطباعة الأعداد الفردية المحصورة بين 15- 123 إذا كانت كلمة المرور 55

مع مراعاة بأن لا يسمح لمستخدم البرنامج بمحاولة إدخال كلمة المرور أكثر من أربع مرات وفي حالة اخفاقه يتم إغلاق البرنامج بالكامل مع عرض الرسالة التالية " كلمة المرور غير صحيحة "





24- اكتب برنامجاً باستقبال 10 أرقام ثم يقوم بطباعة اكبر رقم وأصغر .

**25- أوجد نتحة البرامج التالية :**

(أ)-

```
10 LET Y = 0
20 LET X = 1
30 FOR I = 1 TO 10 STEP 2
40 PRINT X
50 X = X + 2
60 IF I = 3 THEN X = 0
70 NEXT I
80 FOR J = 1 TO 3
90 LET Y = X + Y
100 PRINT Y
110 NEXT J
120 END
```

(ب)-

```
10 FOR I = 15 TO 1 STEP -2
20 PRINT I
30 NEXT I
40 PRINT I + (I^2) + (I^3)
50 END
```

(ج)-

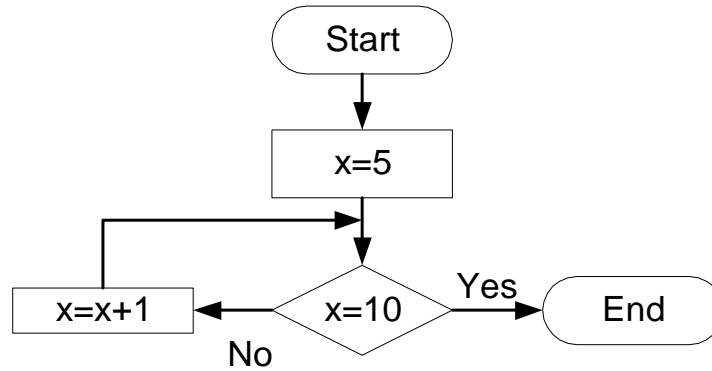
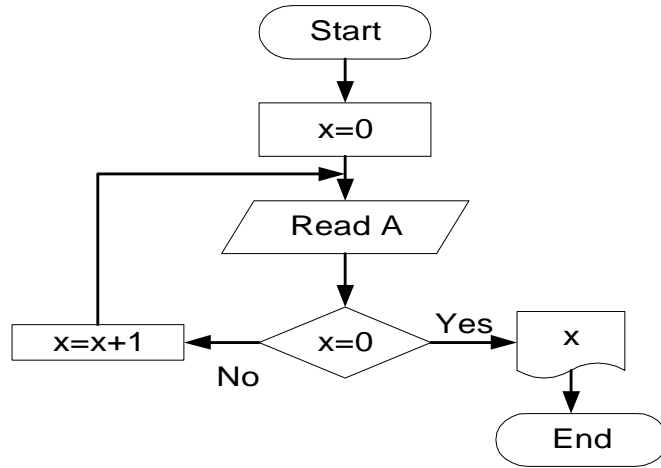
```
10 LET A = 10
20 WHILE A < 5
30 PRINT A
40 WEND
50 END
```

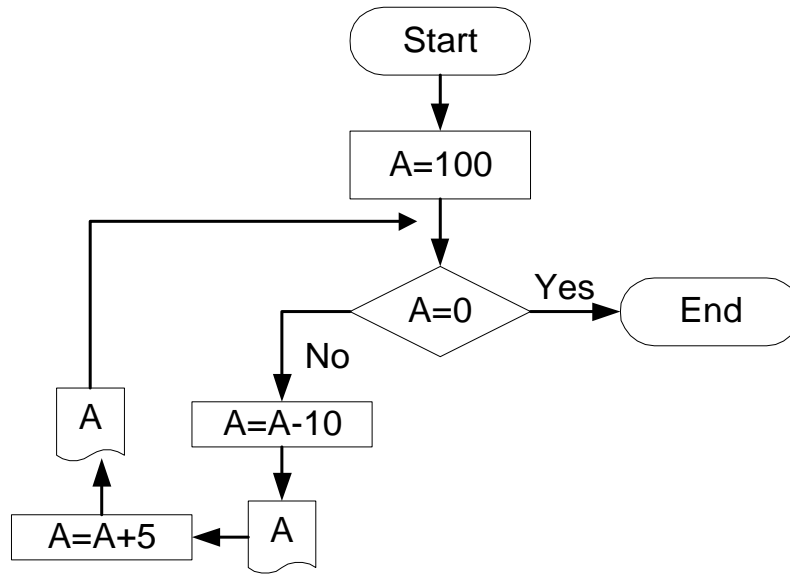
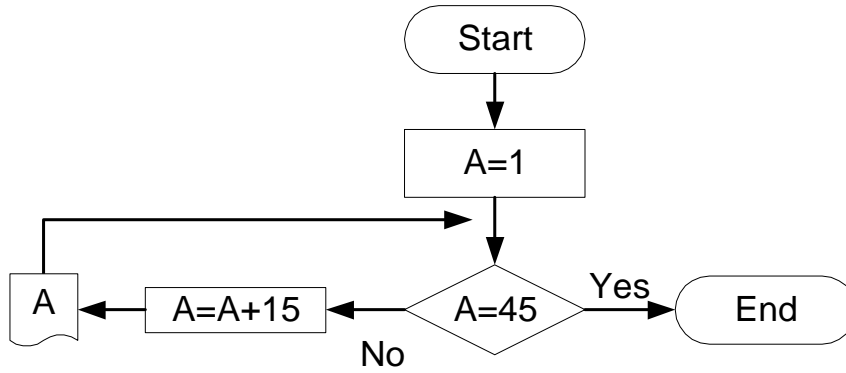
(د)-

```
10 INPUT N
20 FOR I = 1 TO N
30 READ X
40 IF X < 3 : RESTORE
50 PRINT "X=" ; X
60 NEXT I
70 DATA 9,5,7,4,3,12,6,2,14,5,1
80 END
```

\*أوجد نتيجة البرنامج مرتين في المرة الأولى إذا اعتبرنا أن قيمة  $4 = N$  و المرة الثانية  $10 = N$

26-أوجد نتيجة الخوارزميات التالية :





27- ماهو ناتج العمليات التالية في لغة البيسك :

- $(5+6)*4$

- $5+6*4$
- $5+8/4$
- $8/4+5$
- $(7+8)/(1+2)$
- $7+8/1+2$
- $(20+10*(4+2))/(5+15)$

28- ماهو ناتج البرنامج التالي في لغة البيسك عندما  $X = 100, 50, 150$  :

Input X

If  $x < 100$  then go sub 70

If  $x > 100$  then go sub 90

If  $x = 100$  then go sub 110

Print y

End

70  $y = x/10$

return

90  $y = 15*X / 100$

return

110  $y = 15*X + 100$

return

28- أوجد ناتج البرنامج التالي :

```
INPUT "Enter Filename: "; n$
OPEN n$ FOR OUTPUT AS #1
PRINT #1, "This is saved to the file".
```

```
CLOSE
OPEN n$ FOR INPUT AS #1
INPUT #1, a$
PRINT "Read from file: "; a$
CLOSE
```

29- أوجد ناتج البرنامج التالي :

```
INPUT X
INPUT M
FOR I = 1 TO M
Z = I * X
PRINT X; "*"; I; " ="; Z
NEXT I
```