

1- مقدمة :

إن أهم مرحلة في حل مسألة ما باستخدام الحاسوب هي المرحلة المتعلقة بإيجاد خطة الحل , يجب أن تكون هذه الخطة قابلة للتنفيذ من قبل الآلة , وقابلة للتوصيف على وجه لا يدعو إلى اللبس أو التأويل , يطلق اسم الخوارزمية على هذه الخطة .

2- تعريف الخوارزمية :

مجموعة الخطوات المتسلسلة والمحدودة التي تؤدي إلى حل مسألة معينة والوصول إلى نتائج محددة اعتباراً من معطيات ابتدائية.

3- أنواع الخوارزميات :

- ✓ خوارزميات حسابية : تهتم بالمسائل الرياضية  
مثالها ( حل معادلة من الدرجة الأولى )
- ✓ خوارزميات غير حسابية : لا تهتم بالمسائل الرياضية ولكنها تحتاج إلى حل منطقي  
مثالها ( طريقة التدقيق الإملائي لنص ما ، اتخاذ قرار بالذهاب إلى مكان ما وتحديد الطريق الأمثل للوصول إليه )

4- طرق التعبير عن الخوارزمية :

- ✓ الطريقة الكلامية : كتابة الخوارزميات على شكل خطوات باستخدام اللغة المتداولة كاللغة العربية أو الإنكليزية .
- ✓ الطريقة الرمزية : كتابة الخوارزميات باستخدام الرموز .
- ✓ الطريقة التدفقية : كتابة الخوارزميات باستخدام المخططات البيانية (المخططات التدفقية).

مثال توضحي :

اكتب الخوارزمية التي تعطي نتيجة حل التعبير الرياضي الآتي باستخدام اللغة المتداولة (الطريقة الكلامية) :

$$Y=(x^2+7)/x(x+2)$$

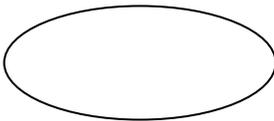
علماً بأن x معلومة

الحل :

- يمكن التعبير عن الخوارزمية باللغة المتداولة(العربية) على الشكل الآتي:
- الخطوة الأولى : أدخل قيمة المتحول x .
- الخطوة الثانية: احسب المقام
- الخطوة الثالثة: إذا كان المقام مساوياً للصفر اطبع " المسألة ليس لها حل "
- الخطوة الرابعة: احسب البسط
- الخطوة الخامسة: احسب قيمة y .
- الخطوة السادسة: اطبع قيمة y .
- الخطوة السابعة: توقف .

5- المخطط التدفقي ( أو الهندسي ) :

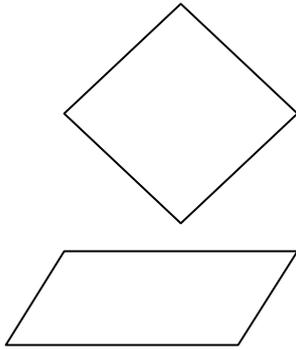
لتحديد بداية الخوارزمية ونهايتها :



عمليات المعالجة:

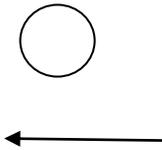


العمليات التي ترتبط باختبار تحقق شرط ما وتتطلب قرارًا منطقيًا :



عمليات الإدخال والإخراج :

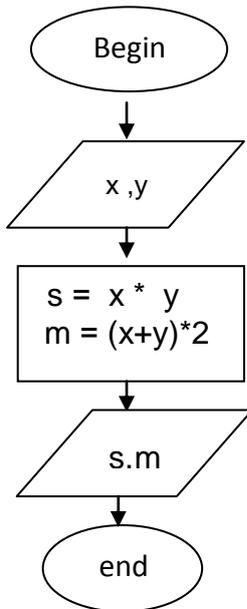
عمليات الربط في حال تعدد الصفحات :



اتجاه تنفيذ الخوارزمية :

-6 أمثلة :

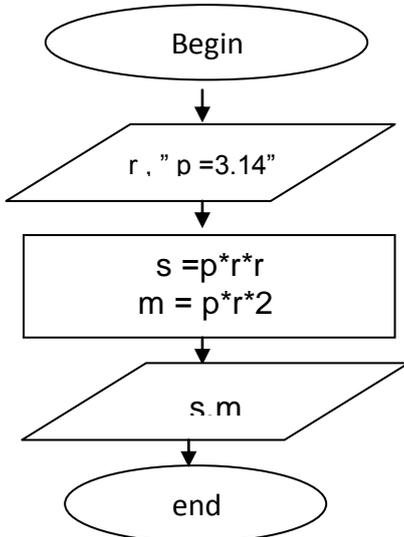
اكتب الخوارزمية الكلامية والرمزية والمخطط التدفقي لإيجاد مساحة ومحيط المستطيل ؟



<u>❖ الخوارزمية الرمزية :</u>	<u>❖ الخوارزمية الكلامية :</u>
<u>المدخلات :</u> x و y	<u>المدخلات :</u> الطول والعرض
<u>المعالجة :</u> $y * x = (s)$ $2 * (x + y) = (m)$	<u>المعالجة :</u> المساحة = (s) = الطول * العرض المحيط (m) = (الطول + العرض) * 2
<u>المخرجات :</u> m, s	<u>المخرجات :</u> المساحة والمحيط

### تمارين:

التمرين الأول: على نمط المثال السابق اكتب الخوارزمية الكلامية والرمزية والمخطط التدفقي لإيجاد مساحة ومحيط الدائرة ؟



<u>❖ الخوارزمية الرمزية :</u>	<u>❖ الخوارزمية الكلامية :</u>
<u>المدخلات :</u> r و p=3.14	<u>المدخلات :</u> نصف القطر و π
<u>المعالجة :</u> $r * r * p = (s)$ $p * 2 * r = (m)$	<u>المعالجة :</u> المساحة = (s) = π * نصف القطر تربيع المحيط (m) = نصف القطر * 2 * π
<u>المخرجات :</u> m, s	<u>المخرجات :</u> المساحة والمحيط للدائرة

**التمرين الثاني :** اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال  $x$  (عدد) وإيجاد قيمة  $y = (x-2)/x$   
**الخوارزمية الرمزية :**

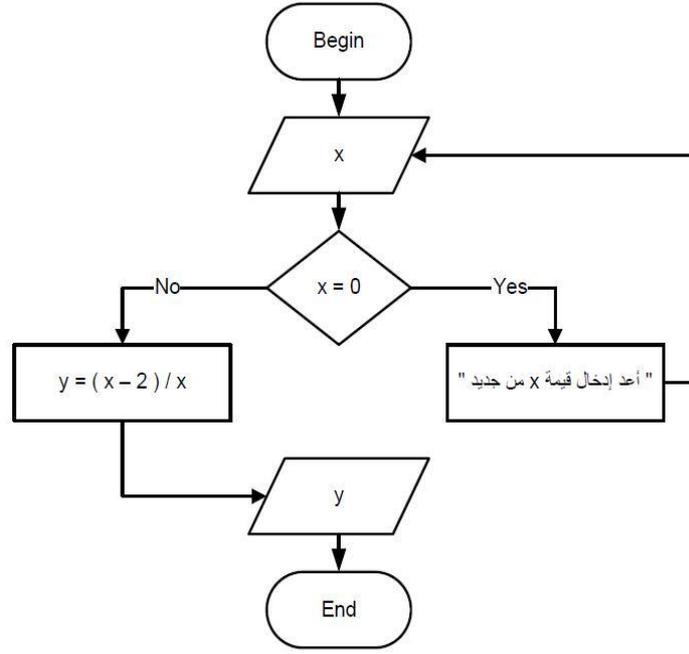
المدخلات :  $x$

المعالجة :

إذا كانت  $(x = 0)$  عندئذ "أعد إدخال قيمة  $x$  من جديد لأنه لا يمكن القسمة على 0"

وإلا فاحسب :  $y = (x - 2) / x$

المخرجات :  $y$



$$y = x / (x - 3)$$

**التمرين الثالث :** اكتب الخوارزمية الرمزية والمخطط التدفقي لإيجاد

**الخوارزمية الرمزية :**

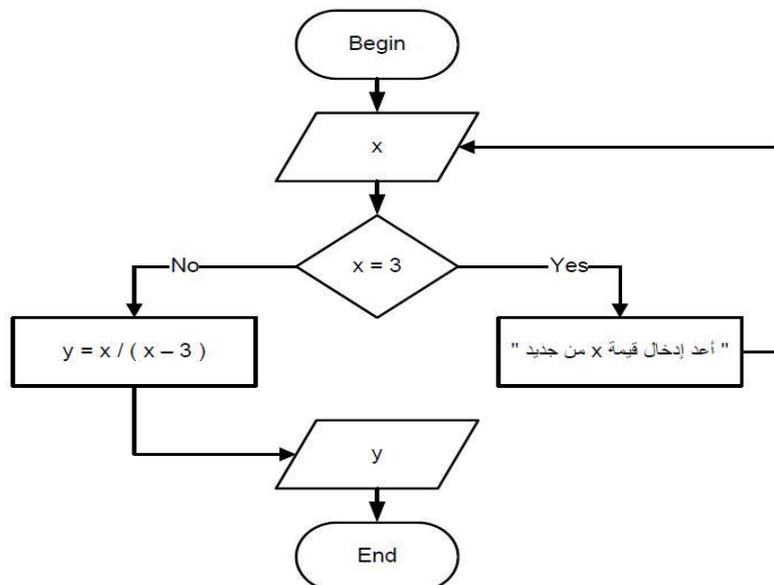
المدخلات :  $x$

المعالجة :

إذا كانت  $(x = 3)$  عندئذ "أعد إدخال قيمة  $x$  من جديد لأنه لا يمكن القسمة على 0"

وإلا فاحسب :  $y = (x - 3) / x$

المخرجات :  $y$



**التمرين الرابع :** اكتب الخوارزمية الرمزية والمخطط التدفقي لحل المعادلة  $ax + b = 0$

مناقشنا جميع الحالات الممكنة لـ  $a, b$

**الخوارزمية الرمزية :**

المدخلات :  $a, b$

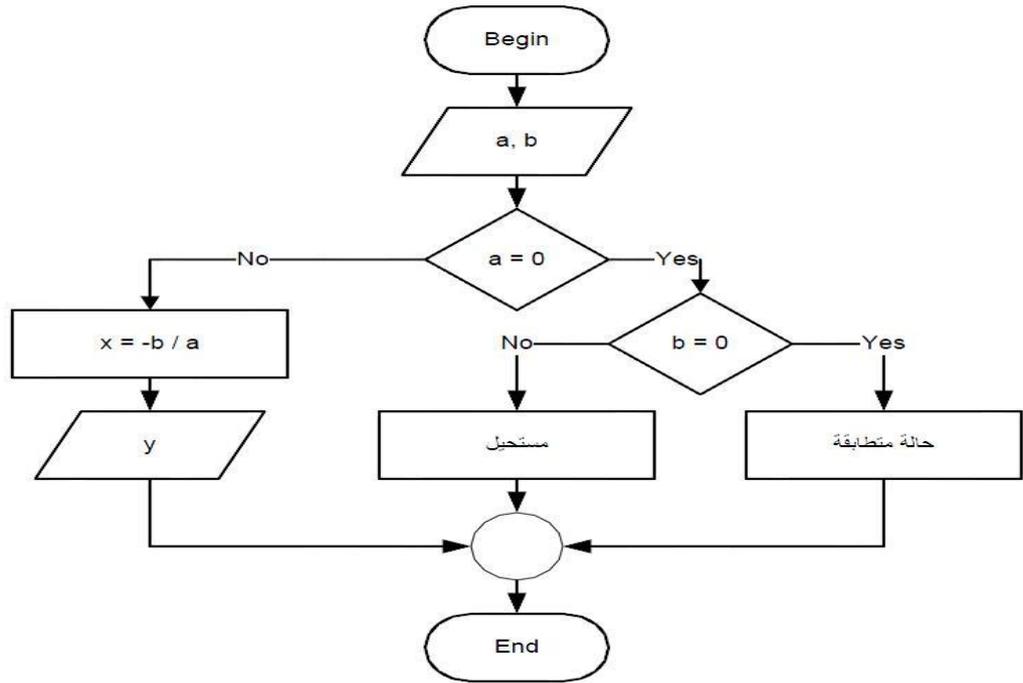
المعالجة :

إذا كانت  $(a=0, b < 0)$  اطبع : " مستحيل الحل "

وإلا إذا كانت  $(a=0, b=0)$  اطبع : " حالة متطابقة "

وإلا  $(a < 0)$  نجد :  $x = -b/a$

المخرجات :  $x$



**التمرين الخامس :** اكتب الخوارزمية الرمزية والمخطط التدفقي لإيجاد قيمة  $y$  المعطاة بالشكل التالي :

$$y = \begin{cases} 2/(x-2) & x > 2 \\ -4/(5-x) & x \leq -2 \end{cases}$$

**الخوارزمية الرمزية :**

المدخلات :  $x$

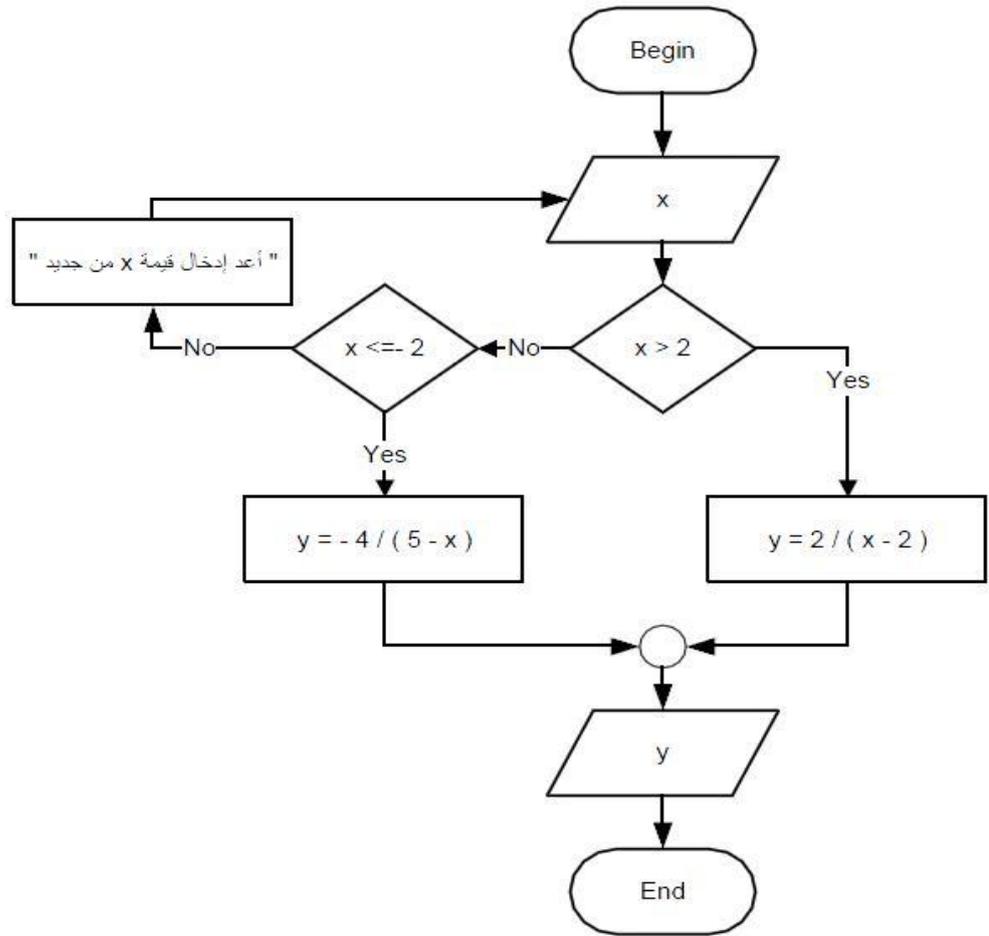
المعالجة :

إذا كانت  $(x > 2)$  عندئذ :  $y = 2 / (x - 2)$

وإلا إذا كانت  $(x \leq -2)$  عندئذ :  $y = -4 / (5 - x)$

وإلا أعد إدخال  $x$

المخرجات :  $y$



**التمرين السادس :**

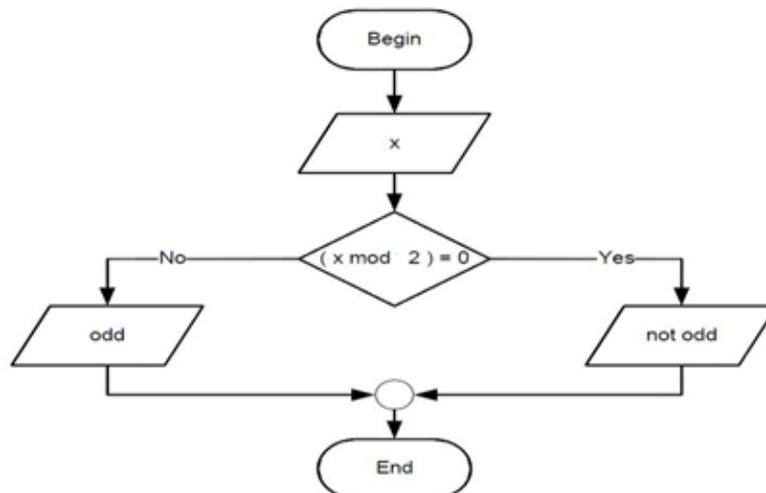
اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عدد صحيح (x) موجب وطباعة إذا كان فردياً أم زوجياً ؟  
الخوارزمية الرمزية :

المدخلات : x  
المعالجة والمخرجات :

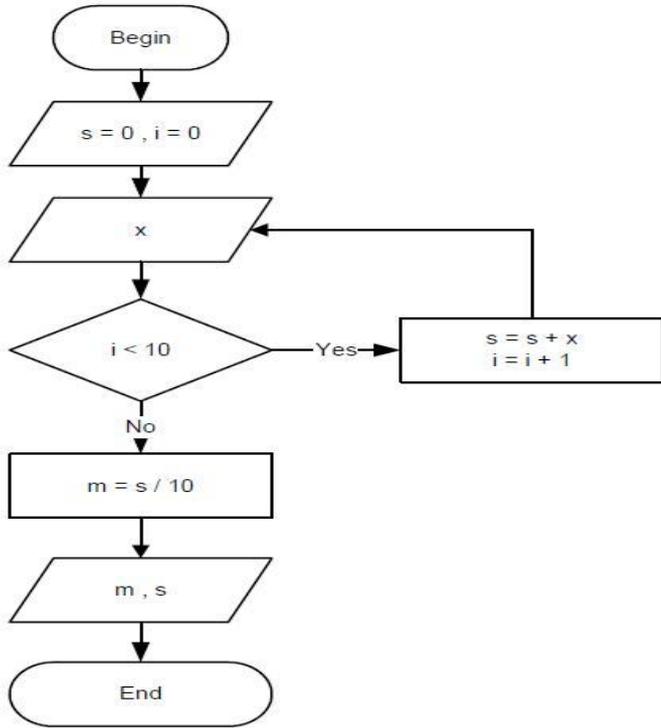
إذا كان باقي قسمة العدد على 2 يساوي صفر (  $x \bmod 2 = 0$  ) فإن

اطبع : " العدد زوجياً " not odd

وإلا اطبع : " العدد فردي odd "



**التمرين السابع :** اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عشرة أعداد مختلفة وإيجاد المتوسط الحسابي والمجموع .



**الخوارزمية الرمزية :**

\* المدخلات :  $x, i=0, s=0$

\* المعالجة :

العداد (  $i=i+1$  ) ، المجموع (  $s=s+x$  )

إذا كان  $i < 10$  عندئذ " أعد إدخال العدد "

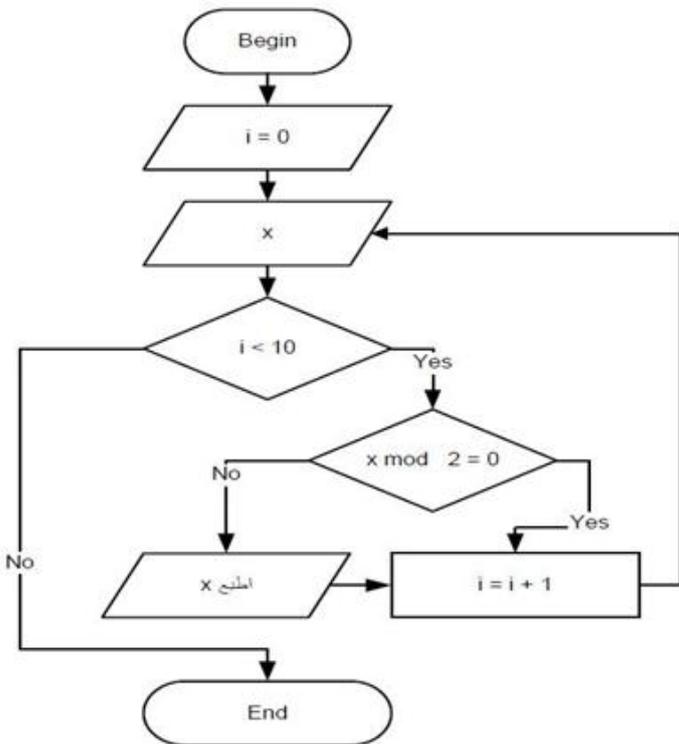
وإلا  $i \geq 10$  عندئذ : " توقف عن إدخال " واحسب :

$m=s/10$

\* المخرجات :

المجموع (  $s$  ) ، المتوسط (  $m$  )

**التمرين الثامن :** اكتب الخوارزمية الرمزية والمخطط التدفقي لإدخال عشرة أعداد وطباعة الفردي منها فقط ؟



**الخوارزمية الرمزية :**

\* المدخلات :  $x, i=0$

\* المعالجة و المخرجات :

العداد (  $i=i+1$  )

إذا كان  $i < 10$  عندئذ " وإذا كان (  $x \bmod 2 = 0$  )

عندئذ  $i=i+1$  و " أعد إدخال  $x$  "

وإلا اطبع قيمة  $x$  الحالية ثم أدخل قيمة جديدة لـ  $x$

وشغل العداد  $i=i+1$

وإلا اخرج من البرنامج

## أساسيات البرمجة بلغة C++

أولاً ..أنواع اللغات :

يمكن تقسيم اللغات المستخدمة في البرمجة الى ثلاثة أنواع :

- 1- لغة الآلة
- 2- لغة المجمع
- 3- اللغات عالية المستوى .

1- لغة الآلة : هي اللغة التي يستطيع الحاسب أن يفهمها مباشرة وهي معرفة من قبل البنية الصلبة للحاسب ، تتألف بشكل عام من سلاسل من الأعداد (مجموعات من الأصفار والواحدات ) التي تعطي الأوامر للحاسب من اجل تنفيذ تعليماته الأولية كل تعليمة على حده .

2- لغة المجمع : هي لغة تستخدم مصطلحات قريبة من اللغة الانكليزية للتعبير عن العمليات الأولية للحاسب ، وقد تم تطوير مترجمات البرامج تسمى بالمجمعات assemblers تحويل البرامج من لغة المجمع إلى لغة الآلة .

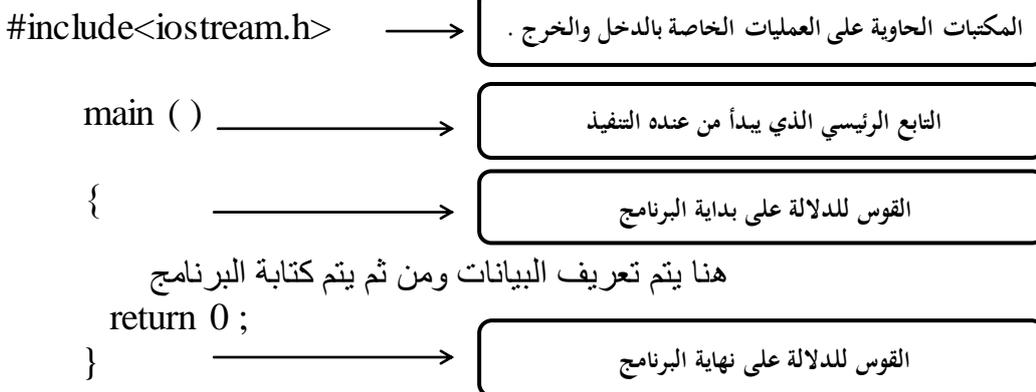
3- اللغات عالية المستوى : هي اللغات التي ظهرت لتسريع عملية البرمجة وذلك باستخدام تعليمات تقوم بالعديد من المهام الجوهرية مثل لغة C++ .

تعتبر لغة C++ من أشهر اللغات التي تتمتع بطابع القوة والمرونة لإنتاج أسرع برامج وأداء أفضل . وعلى الرغم من وجود العديد من لغات البرمجة الأخرى إلا أنها تفتقر شمولية لغة C++ وقوتها . فاللغة C++ تتميز بقابليتها على معالجة التطبيقات الكبيرة والمعقدة، والقوة في صيانة البرامج المكتوبة بها مما يوفر وقتاً في تصميم البرامج وتطويرها .

ملاحظة : تدعى البرامج التي تقوم بتحويل النصوص من البرامج مكتوبة بلغات عالية المستوى الى لغة الآلة بالمترجمات

ثانياً .. كيفية كتابة برنامج بلغة ( C++ ) :

الشكل العام للبرنامج هو :



1 - طباعة نص مؤلف من سطر :

```
// First Program
#include <iostream.h>

// كل الكتابات التي تلي هذه الإشارة ( // ) تسمى تعليق لا يتم تنفيذه
// توجيه ما قبل الترجمة حيث يتم ضم محتوى الملف الرئيسي ذو ( .h )
// الامتداد الحاوي على العمليات الخاصة بالدخل و الخرج لنص البرنامج

main()
{
    // التابع الرئيسي الذي يبدأ من عنده التنفيذ
    // بداية البرنامج
    cout << " welcome to c++ " ; // تعليمة الطباعة
    return 0 ; // إحدى طرق الخروج من التابع
} // نهاية البرنامج
```

2 – برنامج جمع عددين صحيحين:

```
#include <iostream.h>
main()
{
    // تعريف المتحولات
    int x1 , x2 , x3 ;
    // تعليمة الطباعة
    cout << " enter first number " ;
    // تعليمة قراءة متحول
    cin >> x1 ;
    //
    cout << " enter second number " ;
    //
    cin >> x2 ;
    // إجراء عملية الجمع والإسناد إلى المتحول الجديد x3
    x3=x1+x2 ;
    // تعليمة الطباعة المتعددة
    cout << "sum is " << x3 ;
    return 0 ;
}
```

ثالثاً.. المتحولات (المتغيرات) **Variables**:

كل اسم من أسماء المتحولات مثل x1, x2, x3, .... يتم وضعه في الذاكرة ويعرف باسم name ونمط type وحجم size وقيمة value . وبالتالي فإن المتحول x1 يملك الاسم x1 والنمط int والحجم 4بايت والقيمة هي حسب القيمة المقروءة .

5	x1
10	x2
15	x3

مواضع المتحولات في الذاكرة  
مع ذكر الاسم والقيمة

✓ أنواع المتحولات :

<i>char</i>	➤ المتحول المحرفي
<i>short int , int , long int</i>	➤ المتحولات الصحيحة
<i>float , double , long double</i>	➤ المتحولات الحقيقية
<i>bool</i>	➤ المتحول المنطقي
<i>const</i>	➤ الثابت (ويضاف لنوع المتحول)

رابعاً .. قواعد تسمية المتحولات :

بشكل عام يتكون اسم المتحول من مجموعة من الحروف (z ----> a , Z ----> A) والأرقام ويخضع اسم المتحول للقواعد التالية :

- لا يمكن أن يبدأ برقم .
- لا يحتوي على سلسلة فارغة .
- يجب أن لا يكون من الأسماء المحجوزة في لغة C++ مثل: true,false,break,puplic,private,static,new,.....,etc

يكتب تعريف المتحول وفق الصيغة :

*Data-Type name = initial-value ;*  
 قيمة ابتدائية اسم المتحول نوع المعطيات

إن ذكر اسم المتحول ونوع المعطيات الذي ينتمي إليه هو أمر اجباري في كل تعريف لأي متحول ،أما اسناد قيمة ابتدائية له عند تعريفه فهو أمر اختياري يمكن عدم ذكره في عملية التصريح عن المتحول .  
 ملاحظة : من الجدير بالذكر أن لغة C++ تفرق بين الحروف الأبجدية الصغيرة والكبيرة

يبين الجدول التالي أنواع المتحولات ومجالاتها وحجمها :

نوع المتحول	الحجم	المجال
char	1 byte	-128 to 127
int	4 bytes	-2147483648 to 2147483647
short int	2 bytes	-32768 to 32767
long int	4 bytes	-2147483648 to 2147483647
bool	1 byte	True or false
float	4 bytes	-38+E3.45 to 38-E3.4
double	8 bytes	-308+E1.7 to 308-E1.7
long double	8 bytes	-308+E1.7 to 308-E1.7

خامساً.. المعاملات في لغة C++ :

A.المعاملات الحسابية الأحادية :

الوصف	الاستخدام	المعامل
زيادة قيمة المتحول s بمقدار 1 ، استخدام قيمة s قبل زيادته	s++	++
زيادة قيمة المتحول s بمقدار 1 ، استخدام قيمة s بعد زيادته	++s	++
انقاص قيمة المتحول s بمقدار 1 ، استخدام قيمة s قبل انقاصه	s--	--
انقاص قيمة المتحول s بمقدار 1 ، استخدام قيمة s بعد انقاصه	--s	--

**B. المعاملات الحسابية الثنائية :**

الوصف	الاستخدام	العملية
جمع المتحول x مع y	x+y	+
طرح المتحول y من x	x-y	-
ضرب المتحول x بالمتحول y	x*y	*
قسمة المتحول x على المتحول y	x/y	/
باقي قسمة المتحول x على المتحول y	x%y	%

**C. المعاملات العلائقية :**

تستخدم المعاملات العلائقية لمقارنة قيمتين وتحديد العلاقة بينهما والتي تكون النتيجة إما True أو False (يتم تمثيل True بالقيمة 1 و False بالقيمة 0)

الوصف	الاستخدام	العملية
إعادة True في حال كانت x أكبر تماماً من y و False إذا لم يكن x أكبر تماماً من y	x>y	>
إعادة True في حال كانت x أكبر أو تساوي y و False في الحالة غير ذلك.	x>=y	>=
إعادة True في حال كانت x أصغر تماماً من y و False إذا لم يكن x أصغر تماماً من y	x<y	<
إعادة True في حال كانت x أصغر أو تساوي y و False في الحالة غير ذلك.	x<=y	<=
إعادة True في حال كانت x تساوي y و False في حال عدم المساواة .	x==y	==
إعادة True في حال كانت x لا تساوي y و False في حال المساواة .	x!=y	!=

**D. معاملات الإسناد :**

نستخدم معامل الإسناد لإسناد القيمة الموجودة على يمين المعامل إلى المتحول الموجود على يسار المعامل .  
x=5 أسدنا القيمة 5 إلى المتحول x .

**ملاحظة :**

يمكننا في لغة C++ إجراء عمليات حسابية مع عمليات الإسناد مثل الجمع والضرب والطرح والقسمة كالتالي :

المعامل	شكل الاستخدام	مساوي لـ
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

### E. العمليات المنطقية :

يتم استخدام العمليات المنطقية مع العمليات التي يكون لها إحدى قيم True أو False وهذه العمليات هي :

الرمز	العملية
&&	AND
	OR
!	NOT

### ✓ جداول الحقيقة لأدوات الربط المنطقية:

- **النفي "Not"**: لتكن A عبارة منطقية إن تطبيق أداة الربط نفي على هذه العبارة سيعطينا عبارة منطقية جديدة رمزها **!A**.  
جدول الحقيقة للنفي:

A	!A
0	1
1	0

### • **الوصل "And"**: $A \text{ And } B = A \&\& B$

جدول الحقيقة للوصل:

A	B	A&&B
0	0	0
0	1	0
1	0	0
1	1	1

إن العملية A AND B دوماً خاطئة (أي قيمة الحقيقة لها تساوي الصفر) إلا في حالة واحدة فهي صحيحة (أي قيمة الحقيقة لها تساوي الواحد) وذلك في حال كلا العبارتين A,B صحيحتين.

### • **الفصل "OR"**: $A \text{ OR } B = A || B$

جدول الحقيقة للوصل:

A	B	A  B
0	0	0
0	1	1
1	0	1
1	1	1

إن العملية A OR B دوماً صحيحة (أي قيمة الحقيقة لها تساوي الواحد) إلا في حال كلا العبارتين A,B خاطئتين.

### سادساً.. أولوية العمليات الحسابية :

عندما نطبق العمليات الحسابية في العبارات الحسابية فإنها تطبق حسب ترتيب معين تبعاً لقواعد الأولوية بين العمليات وهي :

- ( ) الأقواس :يتم تنفيذ الأقواس الداخلية أولاً أما إذا كان لدينا مجموعة من الأقواس جانب بعضها البعض وعلى نفس المستوى عندها يبدأ الحساب من اليسار إلى اليمين .
- %,/,\*: إذا وجدت على نفس المستوى فإنها تنفذ من اليسار الى اليمين .
- +,- : تنفذ في النهاية، إذا وجدت على نفس المستوى فإنها أيضاً تنفذ من اليسار إلى اليمين .

سابعاً.. سلاسل الهروب :

المعنى	سلسلة الهروب
سطر جديد أي وضع المؤشر في بداية السطر التالي	\n
تحريك المؤشر مسافة جدولية أفقية ( tab )	\t
تستخدم لطباعة علامة الاقتباس "	\"

```
# include <iostream.h>
main ( )
{
cout << "welcome to c++ \n Hello World " ;
return 0;
}
```

حرف الهروب

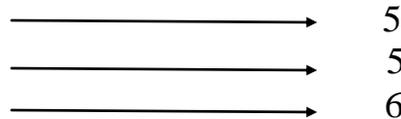
والخرج يكون :

welcome to c++  
Hello World

ثامناً.. أمثلة على العمليات :

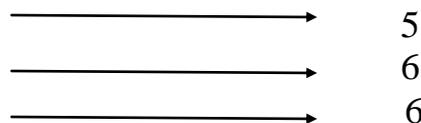
✓ المعاملات الحسابية الأحادية (معاملات الزيادة والنقصان) :

```
# include <iostream.h>
main ( )
{ int a=5;
cout << a << "\n";
cout << a++ << "\n";
cout << a << "\n";
return 0;
}
```



-----\*\*\*\*\*-----

```
# include <iostream.h>
main ( )
{ int a=5;
cout << a << "\n";
cout << ++a << "\n";
cout << a << "\n";
return 0;
}
```



```
# include <iostream.h>
main ( )
{ int a=3,b;
b= a++ *2;
cout<<b<<"\n";
cout<<a<<"\n";
return 0;
}
```

→ 6  
→ 4

-----\*\*\*\*\*-----

```
# include <iostream.h>
main ( )
{ int a=3,b=5,c;
c= - a *2+ b++;
cout<<c<<"\n";
cout<<b<<"\n";
cout<<a<<"\n";
return 0;
}
```

→ 9  
→ 6  
→ 2

### ✓ أولوية العمليات :

- $(5+2)*3+(2*6)/4-1$   
 $=7 * 3 + 12 / 4 - 1$   
 $=21 + 3 - 1 = 23$
- $4 * 8 - 5 + 2 * 6 / 3$   
 $=32 - 5 + 12 / 3$   
 $=32 - 5 + 4 = 31$

### ✓ المعاملات المنطقية :

- $\text{cout} \ll ((4 > 3) \&\& (3 == 2));$  -----> 1 && 0 ----> 0  
 النتيجة المطبوعة النهائية
- $\text{cout} \ll (!(4 < 3) \parallel (3 == 2));$  -----> !(0) || 0----> 1 || 0  
 وبالتالي النتيجة النهائية المطبوعة 1 .

### تمارين :

1. اكتب برنامج يأخذ كدخول ثلاث أعداد صحيحة من لوحة المفاتيح ثم يطبع مجموعها ومتوسطها وناتج جداولها.

```
# include <iostream.h>
main( )
{
int a,b,c ;
cin >> a >> b >> c;
cout << " sum is " << a+b+c << " \n" ;
cout << " average is " << (a+b+c)/3 << " \n";
cout << " product is " << a*b*c;
return 0;}
}
```

2. اكتب برنامج يقرأ نصف قطر دائرة ثم يطبع قيمة قطر الدائرة ، محيطها ، مساحتها.

```
#include <iostream.h>
main( )
{
float r ; // تعريف متحول حقيقي
float const p=3.14; // تعريف متحول حقيقي واسناد قيمة ابتدائية له
cin >> r ;
cout << r * 2 << " \n" ;
cout << 2*p*r << " \n";
cout << p*r*r;
return 0;}
```

3. اكتب برنامج يطبع شكل مستطيل باستخدام سلاسل الهروب .

```
#include <iostream.h>
main()
{
cout << " ***** \n" << " * \t" << " * \n" ;
cout << " * \t" << " * \n";
cout << " * \t" << " * \n";
cout << " * \t" << " * \n";
cout << " * \t" << " * \n";
cout << " ***** \n";
return 0;
}
```

4. اكتب برنامج يقوم بحساب مساحة ومحيط مستطيل (قيم الطول والعرض يتم ادخالها من لوحة المفاتيح).

```
#include<iostream.h>
main()
{float area,length,width,cir; // أسماء للمتحويلات
cout<<"Enter the length: "; //الطول: length
cin>>length;
cout<<"Enter the width: "; //العرض: width
cin>>width;
area=length*width; //المساحة
cir= 2*( length+width); //المحيط
cout<<"\n Area= " << area << endl; // تعني انتقل الى سطر جديد
cout<<" circumference= " << cir << endl;
return 0;}
```

## بنى التحكم (الشروط)

عادة يتم تنفيذ العبارات حسب تسلسل ورودها في البرنامج ويسمى هذا بالتنفيذ التتابعي ( Sequential Execution). لكننا سنتعرض لبعض عبارات C++ والتي تجعل التنفيذ ينتقل لعبارة أخرى قد لا تكون التالية في تسلسل البرنامج، ويسمى هذا بنقل التحكم Transfer of control. تنقسم بنى التحكم في C++ إلى قسمين: بنى التحكم الشرطية والنوع الثاني وهو بنى التحكم التكرارية .

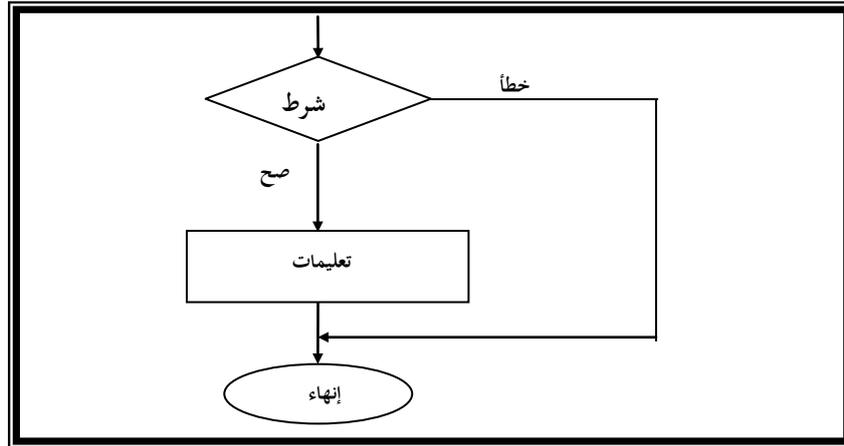
### 1. بنية الاختيار if:

تقوم بنية الاختيار if بتنفيذ فعل معين عندما يكون الشرط المرافق لها محققاً وإلا يتم تجاهله (أي في حال عدم تحقق الشرط لا يتم تنفيذ أي شيء) ولها الشكل العام التالي :

```
if(الشرط condition)
{
statements;
}
```

تعلية أو مجموعة تعليمات في حال كانت تعلية واحدة ليس هناك ضرورة للأقواس أما في حال كانوا أكثر من تعلية فهنا يجب وضع هذه الأقواس { } عندئذ تسمى كتلة تعليمات ( block )

ويبين الشكل التالي طريقة عمل العبارة if :



**مثال 1:** علامة النجاح في إحدى الامتحانات تساوي 60 عندها يكون الكود الخاص لمعرفة الطالب إذا كان ناجح هو كالتالي :

```
#include <iostream.h>
main()
{ int grade;
  cin>>grade;
  if ( grad >= 60 )
  { cout << " passed " ;}
  return 0;}
```

مثال 2 : اكتب برنامج يطلب من المستخدم إدخال عددين صحيحين ثم يطبع العدد الأكبر بينهما متبوعاً بالجملة is larger أما إذا كان العددين متساويين عندها تطبع الكلمة Equals .

```
#include<iostream.h>
main()
{int x,y;
cin>>x>>y;
if(x>y) {cout<<x<<"is larger";}
if(y>x) {cout<<y<<"is larger";}
if(x==y) {cout<<"Equals";}
return 0;
}
```

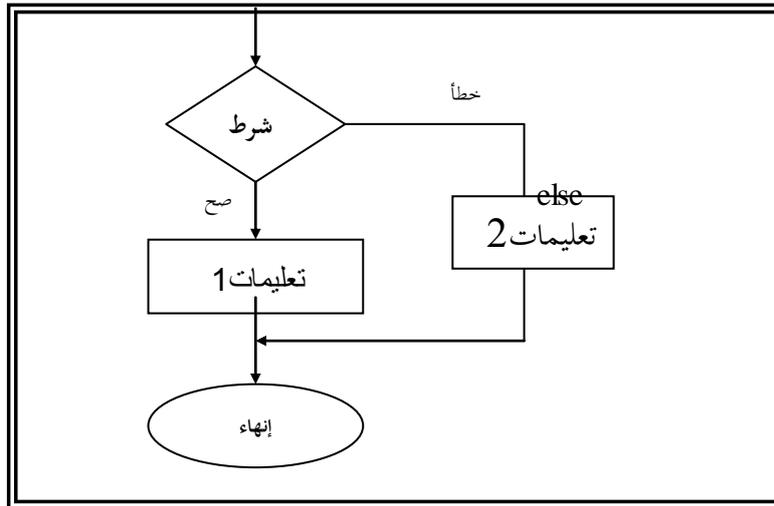
## 2. العبارة if...else :

في العبارة if البسيطة يتم تنفيذ فعل معين إذا كان الشرط صحيحاً، لكن إذا لم يكن كذلك لا يحدث شيء على الإطلاق. لكن لنفترض أننا نريد حدوث شيء في الحالتين إذا كان الشرط صحيحاً وآخر إذا لم يكن كذلك، لتحقيق ذلك نستخدم العبارة if... else أي أن البنية if...else تسمح بتنفيذ مجموعة من الأفعال الممكن تنفيذها إذا كان الشرط المرافق لها محققاً وتنفيذ مجموعة من الأفعال الأخرى في حال عدم تحقق الشرط المرافق لها ولها الشكل التالي :

```
if(الشرط condition)
{Statements 1;
}
else
{Statements 2;
}
```

أي إذا تحقق الشرط فإنه يتم تنفيذ 1 Statements  
أما إذا لم يتحقق يتم تنفيذ 2 Statements

ويبين الشكل التالي طريقة عمل البنية if..else :



**مثال 3:** قراءة علامة طالب (أي ادخالها من لوحة المفاتيح) وطباعة passed في حال نجاحه وطباعة failed في حال رسوبه (علامة النجاح 60).

```
#include<iostream.h>
main( )
{ float mark;
  cin>>mark;
  if(mark>=60)
    {cout<<"passed";}
  else
    {cout<<"failed";}
  return 0;
}
```

**مثال 4:** برنامج لمعرفة العدد اذا كان زوجي أم فردي .

```
#include<iostream.h>
main()
{ int x;
  cin>>x;
  if(x%2==0) // شرط العدد الزوجي
    {cout<<"even";}
  else
    {cout<<"odd";}
  return 0;
}
```

#### ملاحظة:

هناك طريقة أخرى للتعبير عن البنية if/else في لغة C++ وذلك باستخدام ما يسمى بالمعامل الشرطي (-:?) والذي له الشكل التالي:

تعلية 2 : تعلية 1 ? شرط

أي في حال تحقق الشرط يتم تنفيذ تعلية 1 وإلا يتم تنفيذ تعلية 2

تستخدم هذه البنية عادة في دالة الإخراج وفي تعلية الاسناد.

**مثال 5:** طباعة كلمة pass في حال كانت علامة طالب أكثر من 60 و طباعة fail في حال كانت العلامة أقل من 60.

```
# include <iostream.h>
main( )
{ int a;
  cin>>a;
  a>=60 ? cout<<"pass":cout<<"fail";
  return 0 ;
}
```

مثال 6 : طباعة العدد الأكبر بين عددين مدخلين من لوحة المفاتيح .

```
#include <iostream.h>
main( )
{int x,y,max;
  cin>>x>>y;
  max= x>y ? x:y; // max اسناد القيمة المعادة من اختبار الشرط الى المتحول
  cout<<max<<"is greater \n";
  return 0 ;}
```

### 3.العبارات if ... else المتداخلة:

يمكن وضع العبارات if /else ضمن بعضها البعض ، من أجل القيام بفحص عدة حالات ولها الشكل :

```
if(الشرط 1)
{ Statements 1; //1 تحقق الشرط
}
else if(الشرط 2)
{ Statements 2; //2 تحقق الشرط 1 و تحقق الشرط 2
}
else
{ Statements 3; // عدم تحقق جميع الشروط السابقة
}
```

مثال 7 : اكتب برنامج يطلب من المستخدم إدخال عددين صحيحين ثم يطبع العدد الأكبر بينهما متبوعاً بالجملة is larger أما اذا كان العددين متساويين عندها تطبع الكلمة Equals (باستخدام البنية if/else المتداخلة)

```
#include<iostream.h>
main( )
{int x,y;
  cin>>x>>y;
  if(x>y) {cout<<x<<"is larger";}
  else if(y>x) {cout<<y<<"is larger";}
  else {cout<<"Equals";}
  return 0;}
```

### 4. العبارة switch :

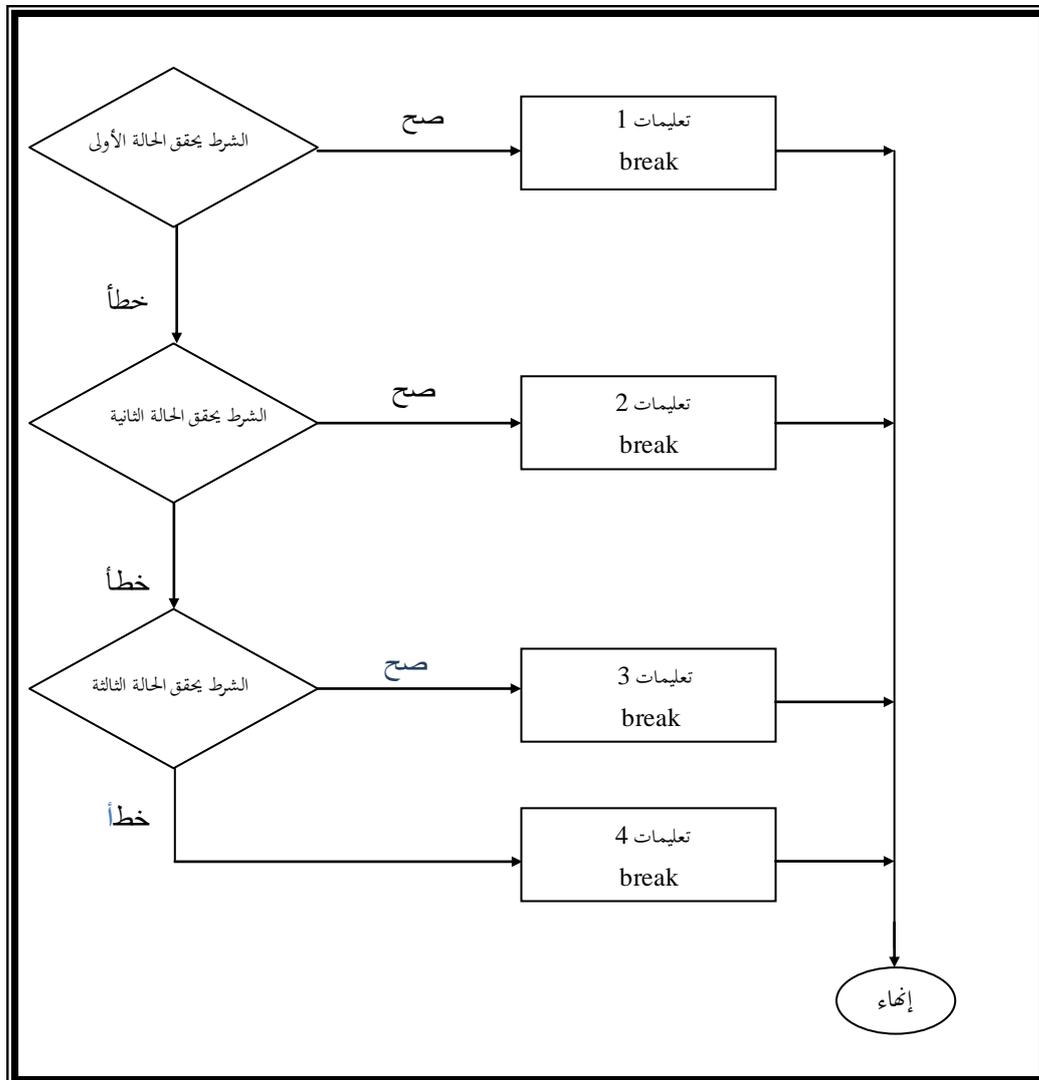
يمكن أن تصادفنا حالة خاصة في إحدى البرامج تحتوي على سلسلة من القرارات التي تتعلق بنتائج متعددة لفحص قيمة متحول أو تعبير ما ،ويمكن ان تؤدي كل نتيجة إلى القيام بفعل مختلف عن الآخر .لذلك توفر لغة C++ البنية switch من أجل التعامل مع حالات اتخاذ القرار المتعلقة بعدة خيارات ،ولها الشكل العام التالي :

```
switch (Variable name) // المتحول قد يكون عدد أو حرف
{
  case constant1 : statement1; break;
  case constant2 : statement2; break;
  .
  .
  case constant n : statement n; break;
  default : last statement;
}
```

تتألف العبارة switch من الكلمة الأساسية switch يليها اسم متغير بين قوسين ، تفحص العبارة switch المتغير وتوجه البرنامج نحو أقسام مختلفة وفقاً لقيم ذلك المتغير.

يتضمن جسم العبارة switch عدداً من الوسوم وهي أسماء تليها نقطتان. تتألف هذه الوسوم من الكلمة الأساسية case ثم ثابت ثم نقطتين.

عندما تكون قيمة متغير العبارة switch مساوية للثابت المذكور في أحد وسوم case ينتقل التنفيذ إلى العبارات التي تلي ذلك الوسم وتؤدي العبارة break إلى منع تنفيذ بقية العبارة switch (أي الخروج الآمن والقسري من حالة case الحالية وتضمن عدم الدخول إلى حالات case الأخرى) ، وإذا لم تتطابق قيمة متغير العبارة switch مع أي وسم ينتقل التنفيذ إلى الوسم الافتراضي default (تعليلة default اختيارية للبنية switch فإذا تم تجاهلها فإن البنية ستعمل بالشكل المعتاد ، فإذا لم يكن هناك أي تطابق بين أي من ثوابت أقسام case وقيمة صيغة التحكم فإن التحكم سينتقل إلى خارج البنية .



مثال 6: طباعة اسم اليوم من أيام الاسبوع عند إدخال رقمه (عند إدخال 1 يطبع Saturday وعند إدخال 2 يطبع Sunday وهكذا.....).

```
#include<iostream.h>
main( )
{int x;
cin>>x;
switch(x)
    {case 1:cout<<"Saturday";break;
    case 2:cout<<"Sunday";break;
    case 3:cout<<"Monday";break;
    case 4:cout<<"Tuesday";break;
    case 5:cout<<"Wednday";break;
    case 6:cout<<"Thursday";break;
    case 7:cout<<"Friday";break;
    default :cout<<"Wrong Number";
    }
return 0;
}
```

### تمارين:

1. برنامج اختبار عدد يتم إدخاله من لوحة المفاتيح إذا كان موجب أم سالب .

```
# include <iostream.h>
main( )
{int a;
cin>>a;
if(a>=0)
    {cout<<"positive";}
else
    {cout<<"negative";}
return 0 ;
}
```

2. اكتب برنامج لاختبار عدد إذا كان قاسم لآخر (يتم إدخال العددين من لوحة المفاتيح).

```
#include<iostream.h>
main( )
{int x,y;
cin>>x>>y;
if(x%y==0 && y!=00)
    {cout<<x<<"is divisible by "<<y;}
else
    {cout<<x<<"is not divisible by "<<y;}
return 0;
}
```

3. برنامج يأخذ كدخل ثلاثة أعداد صحيحة غير متساوية ومن ثم يطبع أصغر هذه الأعداد .

```
#include <iostream.h>
main()
{int a , b ,c ;
cin >> a >> b >> c ;
if ( a < b && a<c)
    {cout << "min is " << a ;}
else if( b < a && b<c)
    {cout << "min is " << b ;}
else if(c < a && c < b)
    {cout << "min is " << c ;}
    else {cout<<" Two numbers are equally or more " ;} // عددان متساويان أو أكثر
return 0 ;}
```

4. برنامج حل معادلة من الدرجة الأولى لها الشكل :  $ax+b=0$  حيث  $a, b$  أعداد ثابتة .

```
#include <iostream.h>
main( )
{ int a,b;
  cin>>a>>b;
  cout<<a<<"X+"<<b<<"=0"<<"\n";
  if(a==0 && b!=0)
    {cout<<"Unsolvable";} // مستحيلة الحل
  else if(a==0 && b==0)
    {cout<<"State match";} // حالة تطابق
  else
    {cout<<"X="<<-1*b/a;}
return 0 ;}
```

5. اكتب برنامج لحساب المعادلة :

$$y = \begin{cases} x^2 + 1 & : x > 0 \\ x + 5 & : x = 0 \\ 2x^3 - 1 & : x < 0 \end{cases}$$

```
#include<iostream.h>
main( )
{int x,y;
cout<<"Enter the x: ";
cin>> x;
if(x >0)
  {y=x*x+1;}
else if(x == 0)
  {y=x+5;}
  else
    {y=2*x*x*x-1;}
cout<< y;
return 0;
}
```

6. اكتب برنامج لحساب المعادلة :

$$y = \begin{cases} x^2 - 1 & : 10 < x < 100 \\ x^3 - 1 & : x > 100 \end{cases}$$

```
# include <iostream.h>
main( )
{
int x,y;
cout<<"Enter the x: ";
cin>> x;
if(x >10&& x<100)
{
    y=x*x-1;
    cout<< y;
}
else if(x>100)
{
    y=x*x*x-1;
    cout<< y;
}
return 0;
}
```

7. اكتب برنامج لإدخال نتيجة طالب وطباعة تقديره .

```
# include <iostream.h>
main( )
{
int grade;
cout<<"Enter the grade: ";
cin>> grade;
if(grade >=90)
{cout<<" : Excellent "<<endl;} // ممتاز
else if(grade >=80)
{cout<<" : Very Good "<<endl;} // جيد جداً
else if(grade >=65)
{cout<<" : Good "<<endl;} // جيد
else if(grade >=50)
{cout<<" : Accepted "<<endl;} // مقبول
else if(grade < 50)
{cout<<" : is failing "<<endl;} // راسب
return 0;
}
```

8. برنامج لإنجاز العمليات الحسابية (+,-,\*,/) لعددتين a,b على أن يتم إدخال رمز العملية والعددتين من لوحة المفاتيح باستخدام البنية switch.

```
# include <iostream.h>
main( )
{
int x,y;
char op;
cout<<"Enter The Numbers .:";
cin>>x>>y;
cout<<"Enter The Operation : ";
cin>>op;
switch(op)
{case '+': cout<<x+y<<"\n";break;
case '-': cout<<x-y<<"\n";break;
case '*': cout<<x*y<<"\n";break;
case '/': cout<<x/y<<"\n";break;
default:cout<<"Error Operation\n";
}
return 0 ;
}
```

## الحلقات في لغة C++

توفر C++ عدداً من أساليب التكرار (حلقات) التي تستخدم لتكرار أجزاء من البرنامج قدر ما تدعو الحاجة، لتحديد عدد مرات تكرار الحلقة تفحص كل حلقات C++ ما إذا كان تعبير ما يساوي صحيح (true) أو خطأ (false) يبلغها هذا ما إذا كان عليها التكرار مرة إضافية أخرى أو التوقف فوراً. هنالك ثلاثة أنواع من الحلقات في C++:

### 1. الحلقة for

تأخذ الحلقة for الشكل العام التالي:

```
for( expression1; expression2; expression3)
{ statement; }
```

حيث يمثل:

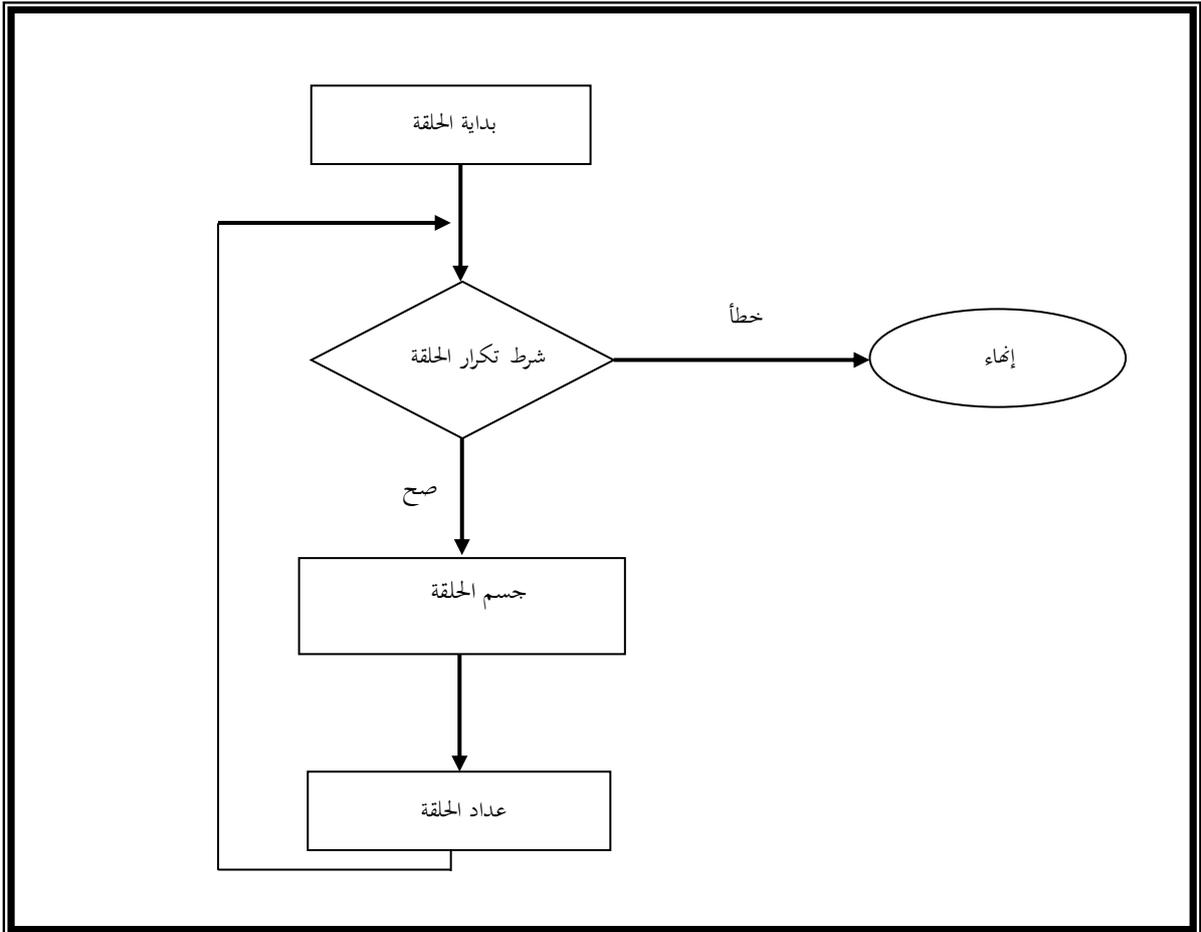
expression1 : بداية الحلقة.

expression2 : شرط الحلقة الذي يفحص قيمة عداد الحلقة ويحدد ما إذا كان يجب تكرار الحلقة مرة أخرى أم لا.

expression3 : يمثل زيادة الحلقة الذي يقوم بزيادة أو إنقاص قيمة عداد الحلقة.

في الحلقة for يكون عدد مرات تنفيذ الحلقة مذكوراً عادة في بدايتها.

الشكل التالي يبين كيفية عمل الحلقة for.



طريقة عمل الحلقة for

**مثال 1:**

المثال التالي يقوم بطباعة قيم المتغير i من 1 إلى 10 .

```
#include <iostream.h>
main( )
{
for ( int i = 1; i<= 10; i++)
{ cout<<i <<"\n";}
return 0;
}
```

نتيجة التنفيذ:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

تحتوي الأقواس التي تلي الكلمة الأساسية for على ثلاثة تعابير مختلفة تفصلها فاصلة منقوطة. تعمل هذه التعابير الثلاثة في أغلب الأوقات على متغير يدعى عداد الحلقة، وهو المتغير i في المثال السابق. هذه التعابير هي:

بداية الحلقة: يمهد قيمة عداد الحلقة عادة `int i = 1;`  
 شرط الحلقة: يفحص قيمة العداد ليرى ما إذا كان يجب تكرار الحلقة مرة أخرى أو إيقافها `i <= 10;`  
 زيادة الحلقة: يقوم عادة بزيادة (أو إنقاص) قيمة عداد الحلقة `i++`.  
 المثال التالي يقوم بإنقاص عداد الحلقة بـ 1 كلما تكررت الحلقة:

**مثال 2:**

```
#include <iostream.h>
main ( )
{
for ( int j=10; j>0; j--)
{ cout<<j<<" ";}
return 0;}
```

نتيجة التنفيذ: 10 9 8 7 6 5 4 3 2 1

ويمكن أيضاً زيادة أو إنقاص عداد الحلقة بقيمة أخرى .  
**مثال 3:** البرنامج التالي يوضح ذلك :

```
#include <iostream.h>
main ( )
{
for (int j=10; j<100; j+=10)
{ cout<<j<<" ";}
return 0;}
```

نتيجة التنفيذ: 10 20 30 40 50 60 70 80 90

**حلقات for المتداخلة:**

تتكون الحلقات المتداخلة من حلقة خارجية وحلقة أخرى داخلية أو أكثر وفي كل مرة تتكرر الحلقة الخارجية يتم تكرار الحلقة الداخلية من بداية العداد إلى نهايته .  
تأخذ الحلقات for المتداخلة الشكل العام التالي :

```
for (.....)
  for (.....)
    { statements; }
```

مثال 4:

```
#include <iostream.h>
main( )
{ int i,j;
  for (i=1 ; i<5;i++)
  {
    for (j=1 ; j<4;j++)
      {cout << i<<j<<" \t";}
    cout<<"\n";
  }
  return 0; }
```

نلاحظ هنا أن الحلقة الداخلية تتكرر 3 مرات لكل قيمة من قيم i (عداد الحلقة الخارجية).  
نتيجة التنفيذ:

11	12	13
21	22	23
31	32	33
41	42	43

يمكننا وضع أي نوع من الحلقات ضمن أي نوع آخر، ويمكن مداخل الحلقات في حلقات متداخلة في حلقات أخرى وهكذا.

**2. الحلقة while**

تأخذ الحلقة while الشكل العام التالي:

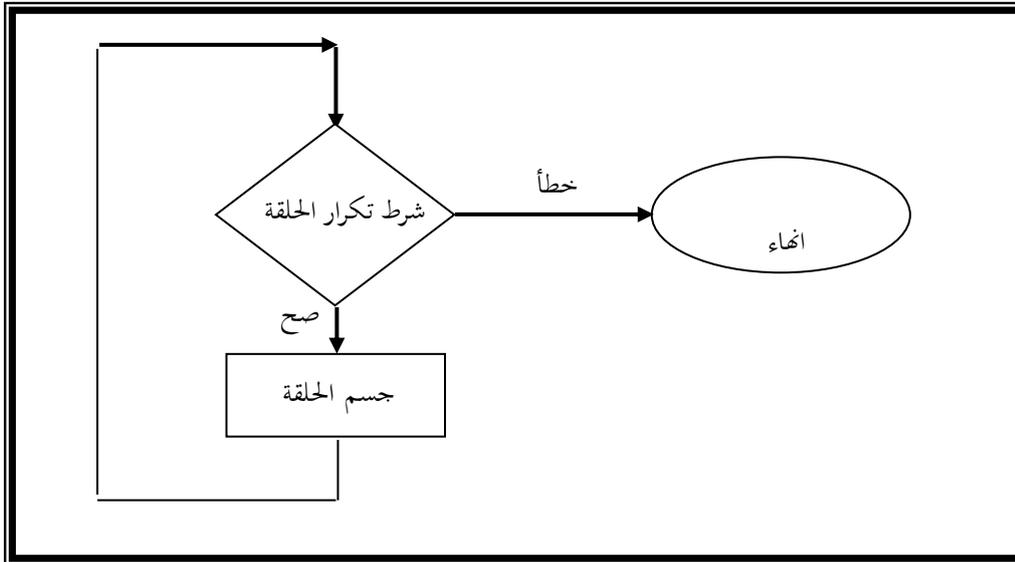
```
while( condition)
  {statement}
```

مثال 5 :

```
#include <iostream.h>
main( )
{ int n=3;
  while (n<30)
    { cout<<n<<" ";
      n=n*2 ;
    }
  return 0; }
```

والنتيجة هي: 3 6 12 24  
أي يقوم بطباعة القيم التي تأخذها n طالما هذه القيم أصغر تماماً من 30 أي ستستمر هذه الحلقة في مضاعفة المتغير n إلى أن تصبح قيمة n أكبر من 30 عندها تتوقف .

تتكون الحلقة من الكلمة الأساسية `while` يليها تعبير اختبار بين أقواس ويكون جسم الحلقة محصوراً بين أقواس حاصرة `{ }` إلا إذا كان يتألف من عبارة واحدة. الشكل التالي يبين طريقة عمل الحلقة `while`:



طريقة عمل الحلقة `while`

مما يجدر التنويه إليه هنا أنه يتم فحص شرط الحلقة قبل تنفيذ جسم الحلقة، وعليه لن يتم تنفيذ جسم الحلقة أبداً إذا كان الشرط خطأ عند دخول الحلقة وعليه المتغير  $n$  في المثال السابق يجب تمهيده عند قيمة أقل من 100 .

**مثال 6 :**

برنامج حساب المتوسط الحسابي لعلامات 10 طلاب في امتحان .

```
#include<iostream.h>
main () {
int i, grade, total ;
float average;
total = 0;
i = 1;
while (i <= 10)
{ cin >>grade;
total = total + grade;
i = i + 1;
}
average = total /10;
cout << " Class average is: " << average <<"\n";
return 0; }
```

75 65 51 89 71 54 80 79 81 90

إذا كانت العلامات المدخلة كالتالي

فإن نتيجة التنفيذ:

Class average is : 73.5

### 3. الحلقة **do ...while**:

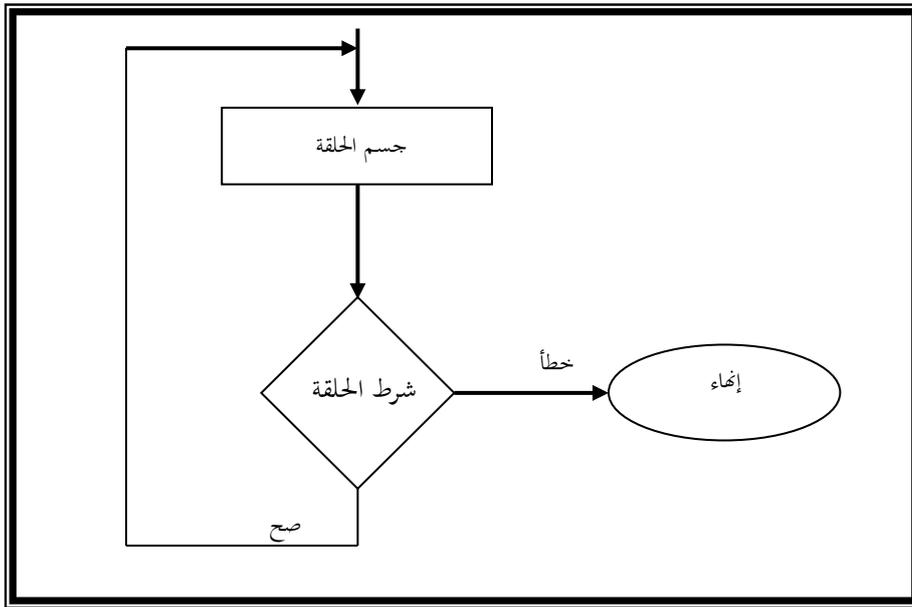
تعمل الحلقة do (غالباً تسمى do...while...) كالحلقة while، إلا أنها تفحص شرط الحلقة بعد تنفيذ جسم الحلقة. وتستخدم أيضاً عندما نريد القيام بجزء من البرنامج مرة واحدة على الأقل. الشكل التالي يبين كيفية عمل الحلقة do. تأخذ الحلقة do الشكل التالي:

**do**

{ statement; }

**while**( condition);

الحلقة do تفحص شرط الحلقة بعد تنفيذ جسم الحلقة، وعليه يتم تكرار جسم الحلقة do مرة واحدة على الأقل حتى ولو كان الشرط غير متحقق وتفسير ذلك أن التحقق من الشرط يتم بعد التنفيذ وليس قبله كما في الحلقتين السابقتين..



طريقة عمل الحلقة do.. while

تبدأ الحلقة do بالكلمة الأساسية do يليها جسم الحلقة بين أقواس حاصرة { } ثم الكلمة الأساسية while ثم تعبير اختبار بين أقواس ثم فاصلة منقوطة.

**مثال 6:** البرنامج التالي يقوم بطباعة الأعداد من 1 إلى 10 .

```
#include <iostream.h>
main ( )
{ int i = 1;
do {cout<< i <<" ";
i++;
} while (i <= 10);
return 0;}
```

تقوم " cout<< " بطباعة مسافة خالية بين كل رقم والآخر وعليه الخرج من البرنامج يكون كالتالي:

1 2 3 4 5 6 7 8 9 10

**تمارين:**

1. برنامج يطبع الأعداد الزوجية الموجودة في المجال [1..100] بشكل تصاعدي وبشكل تنازلي .

● بشكل تصاعدي : طريقة اولى

```
#include<iostream.h>
main()
{ for(int i=2;i<=100;i=i+2)
  { cout<<i<<"\t";}
  return 0;}
```

طريقة ثانية

```
#include<iostream.h>
main()
{
  for(int i=1;i<=100;i++)
  { if(i%2==0)
    { cout<<i<<"\t";}
  }
  return 0;}
```

● بشكل تنازلي

```
#include<iostream.h>
main()
{ for(int i=100;i>=1;i=i-2)
  { cout<<i<<"\t";}
  return 0;}
```

2. برنامج يحسب مجموع الأعداد من 1 الى 10 .

```
#include<iostream.h>
main( )
{ int i,s=0;
  for (i=1;i<=10;i++)
    {s+=i; }
  cout<<s;
  return 0; }
```

باستخدام **for**

\*\*\*\*\*

```
#include<iostream.h>
main( )
{ int i=1,s=0;
  while(i<=10)
    {s=s+i;
    i++;}
  cout<<s;
  return 0; }
```

باستخدام **while**

\*\*\*\*\*

```
#include<iostream.h>
main( )
{ int i=1,s=0;
do {s+=i;
    i++;
    }while(i<=10);
cout<<s;
return 0;}
```

باستخدام do .....while

3. برنامج لإدخال عشرة أحرف وطباعة حرف "A" إن وجد و كم مرة أدخل الحرف:

```
# include <iostream.h>
main( )
{
char letter ;
int count=0;
for (int i=1;i<=5;i++)
{
cin >> letter ;
if(letter=='A')
{count++;}
}
cout<<"Letter A appears:"<<count<<"\n";
return 0;
}
```

4. برنامج يطبع جدول الضرب للرقم المدخل فقط.

```
#include<iostream.h>
main( )
{
int i,x;
cout<<"Enter the number : ";
cin>> x;
for (i=1;i<=10;i++)
{cout<<x<<"*"<<i<<"="<<x*i<<" ";
cout<<"\n";
}
return 0; }
```

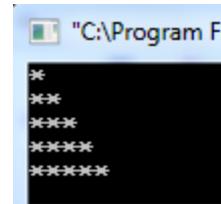
5. برنامج يطبع جدول الضرب الى العدد 10 .

```
#include<iostream.h>
main()
{
    for (int i=1;i<=10;i++)
        {
            cout<<"Multiply Table for "<<i<<"\n\n";
            for(int j=1;j<=10;j++)
                {
                    cout<<i<<"*"<<j<<"="<<i*j<<"\n";
                }
            cout<<"\n";
        }
return 0;}
```

6. برنامج يطبع الشكل التالي :

```
#include<iostream.h>
main()
{ int i,j;
  for (i=1;i<=5;i++)
    {
      for (j=1;j<=i;j++)
        {
          cout<< " *";
        }

      cout<<"\n";
    }
cout<<"\n";
return 0; }
```



7. برنامج يعمل إدخال علامات عشرة طلاب وطباعة المعدل العام للطلاب العشرة:

```
#include <iostream.h>
main()
{ float mark,sum ;
  for(int i=1;i<=10;i++)
    {
      cin >> mark ;
      sum = sum + mark ;
    }
  cout << " average is : " << sum / 10 ;
  return 0; }
```

8. طباعة مجموع الأعداد 3.5 , 4 , 4.5,.....,9.5

```
#include<iostream.h>
main()
{float s=0,i;
  for(i=3.5;i<=9.5;i=i+0.5)
    { s=s+i;}
  cout<<s<<"\n";
  return 0;}
```

9. برنامج ايجاد قواسم عدد ما:

```
# include <iostream.h>
main( )
{ int x ;
  cout << " enter number: " ;
  cin >> x;
  for ( int i=1 ;i<=x ; i++ )
    { if(x%i==0)
      {cout << i <<"\n";}
    }
  return 0 ; }
```

## المصفوفات

### 1- تعريف المصفوفة:

المصفوفة هي نوع من أنواع بنية البيانات، لها عدد محدود ومرتب من العناصر التي تكون جميعها من نفس النوع type، فمثلاً يمكن أن تكون جميعها أعداد صحيحة int أو حقيقية float أو محارف char ولكن لا يمكن الجمع بين نوعين مختلفين في نفس المصفوفة .

وعند تعريف المصفوفة وإنشائها يتم حجز عدد محدد من المواقع المتجاورة في الذاكرة لتخزين البيانات فيها، حيث يتم الوصول الى البيانات المخزنة في هذه المواقع عن طريق اسم المصفوفة ورقم الموقع (index) والغاية من استخدام المصفوفات هي تخزين عدد من القيم تحت اسم واحد فقط (يكون لها النمط نفسه) دون الحاجة الى تخزين كل قيمة في متحول منفصل .

### 2- ميزات المصفوفات :

- أ- تقليل حجم البرنامج .
- ب- سهولة اسناد القيم واسترجاعها.
- ت- استخدام تقنيات البحث والترتيب .
- ث- الوصول المباشر إلى البيانات المخزنة فيها.

### 3- عيوب المصفوفات :

- أ- لا يمكن تحديد حجمها أثناء التنفيذ(فقط يمكن تحديد حجمها عند التصريح عنها).
- ب- جميع عناصرها يجب أن تكون من نوع واحد للبيانات (أي لا تحتوي على عناصر مختلفة من انواع البيانات).

### 4- أنواع المصفوفات :

#### أولاً .. المصفوفات أحادية البعد :

وتكون عبارة عن صف واحد من العناصر . وهي تتكون من مواقع متجاورة لتخزين بيانات عددية فيها على شكل صف واحد وكل موقع (دليل index) رقم مخصص له ويتم ترقيم هذه المواقع (الأدلة) بالتتالي (0,1,2,.....,n-1) ( n طول المصفوفة أي عدد عناصرها).  
لتعريفها وحجز موقع لها :

Data Type     name of array [n];

حيث :

Data Type :نوع بيانات المصفوفة .

name of array : اسم للمصفوفة .

n : عدد عناصر المصفوفة .

مثال :مصفوفة أعداد صحيحة عدد عناصرها 5 .

```
int arr [ 5];
```

أما لإعطاء قيم ابتدائية لهذه المصفوفة :

```
arr [0]=5;
```

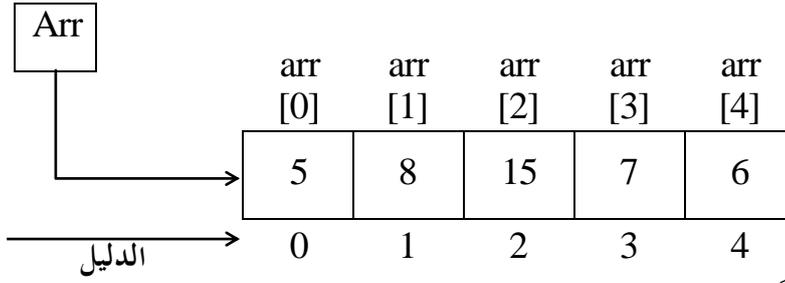
```
arr [1]=8;
```

```
arr [2]=15;
```

```
arr [3]=7;
```

```
arr [4]=6;
```

حيث أن دليل أول عنصر في المصفوفة هو 0 أما آخر عنصر فدليله n-1 .



أما لطباعة عنصر معين مثلاً العنصر الذي دليله 2 من المصفوفة السابقة يكون كالآتي: `cout<<arr[2];`  
ولتعديل القيمة المخزنة في الموقع 1 لتصبح 18 بدل من 8 نكتب:  
`arr [1]=18;`  
ولإدخال عناصر المصفوفة arr من لوحة المفاتيح:

```
cin>>arr[0];
cin>>arr[1];
cin>>arr[2];
cin>>arr[3];
cin>>arr[4];
```

وبالتالي يمكن استخدام حلقة for يبدأ العداد فيها من الـ 0 دليل أول عنصر وينتهي بأصغر تماماً من n عدد عناصر المصفوفة.

```
for(int i=0;i<5;i++)
{ cin>>arr[i];}
```

وكذلك الأمر من أجل طباعة عناصر مصفوفة نستخدم حلقة for :

```
for(int i=0;i<5;i++)
{ cout<<arr[i];}
```

• أمثلة عن طريق إعطاء قيم ابتدائية لعناصر المصفوفة :

1. يمكن إعطاء قيمة الصفر لكل عناصر المصفوفة على الشكل التالي :

```
int A[10]={0};
```

مثلاً

2. يمكن تحديد القيم الابتدائية للمصفوفة أثناء التصريح عنها بالطريقة التالية :

```
int arr [5]={5,8,15,7,6};
```

3. يمكن إعطاء قيم ابتدائية لعناصر المصفوفة كقيمة معينة واحدة لكل العناصر بالشكل التالي :

```
# include <iostream.h>
```

```
main( )
```

```
{ int A[5] ;
```

```
for (int I =0 ; I <5 ; I ++ )
```

```
{ A[I ] = 5 ;}
```

```
return 0 ;}
```

**ملاحظات :**

1. بسبب التصريح التالي : `int n[5] = { 1 , 2 , 34 , 56 , 24 , 14 } ;`

خطأ قواعدياً لأننا أعطينا ستة قيم لمصفوفة مؤلفة من خمسة عناصر فقط .

2. بسبب التصريح التالي : `int n[5] = { 1 , 2 , 4 , 6 , 2 } ;`

إعطاء قيمة الصفر للعنصر الخامس من قبل المترجم .

3. إذا تم حذف حجم المصفوفة أثناء التصريح عنها فإن عدد عناصر هذه المصفوفة يصبح مساوياً

لعدد القيم الابتدائية المعطاة ضمن القائمة الملحقة بالتصريح . لذلك يقوم التصريح التالي :

```
int n[ ] = { 1,2,3,4,5,6 } ;
```

بإنشاء مصفوفة مؤلفة من ستة عناصر .

**مثال 1 :**

اكتب برنامج لإدخال مصفوفة أعداد صحيحة عدد عناصرها 8 من لوحة المفاتيح ومن ثم طباعة هذه المصفوفة بالشكل النظامي وأيضاً طباعتها بالشكل العكسي (معكوس المصفوفة أي ابتداءً من آخر عنصر الى أول عنصر).

```
#include<iostream.h>
main( )
{int a[8];
  for(int i=0;i<8;i++)
    {cin>>a[i];}
  for(int i=0;i<8;i++)
    {cout<<a[i]<<"\t";}
  cout<<"\n";
  for(int i=7;i>=0;i--)
    {cout<<a[i]<<"\t";}
return 0;
}
```

**ملاحظة :**

تحدثنا في المحاضرة الأولى عن المتحول الثابت وهو المتحول الذي لا يمكن تغيير قيمته بعد التصريح عنه . ويمكن استخدام المتحول الثابت في تحديد حجم المصفوفة .

```
const int size =10 ;
int s [size] ;
```

تفيد التعليمات السابقة في تحديد حجم مصفوفة s باستخدام الثابت size . ويفيد استخدام المتحولات الثابتة لتحديد حجم المصفوفات في جعل البرامج أكثر قابلية لتغيير الحجم. فمثلاً حلقة for تقوم بتعبئة 10 عناصر يمكن تعديلها لتقوم بتعبئة 1000 عنصر وذلك بتغيير قيمة الثابت المرتبطة به أما في حالة عدم استخدام الثوابت فيتطلب التعديل السابق عدة تعديلات في أماكن مختلفة من البرنامج .

**تمارين :**

1. اكتب برنامج لإدخال مصفوفة أعداد صحيحة طولها 10 و طباعة الأعداد الزوجية فقط الموجودة في المصفوفة

```
# include <iostream.h>
main( )
{int num[10];
  for(int i=0;i<10;i++)
    {cin>>num[i];}
  for(int i=0;i<10;i++)
    {
      if(num[i]%2==0)
        {cout<<num[i]<<"\n";}
    }
  return 0 ;
}
```

2. اكتب برنامج لإدخال مصفوفة أعداد صحيحة عدد عناصرها 10 ومن ثم طباعة مجموع أعداد هذه المصفوفة .

```
# include <iostream.h>
main( )
{
    int num[10];
    int s=0;
    for(int i=0;i<10;i++)
    {
        cin>>num[i];
        s=s+num[i];
    }
    cout<<"The Sum:"<<s;
    return 0;}

```

3. اكتب برنامج جدول الضرب للعدد 4 وتخزينه في مصفوفة وطباعة قيم الجدول هذا(أي طباعة المصفوفة) .

```
# include <iostream.h>
main()
{
    int mult[11];
    for(int i=0;i<11;i++)
    {
        mult[i]=4*i;
        cout<<"4 *"<<i<<"="<<mult[i]<< "\n";
    }
    return 0;}

```

b[ ]={3,1,7,8}

a[ ]={2,4,1,7}

4. لدينا المصفوفتان الأحاديتان  
اكتب برنامج لتصريح وطباعة ناتج جمع هاتين المصفوفتين

```
# include <iostream.h>
main( )
{
    int a[ ]={2,4,1,7};
    int b[ ]={3,1,7,8};
    int c[4];
    for(int i=0;i<4;i++)
    {
        c[i]=a[i]+b[i];
        cout<<"c["<<i<<"]="<<c[i]<<endl;
    }
    return 0 ;
}

```

5. اكتب برنامج لتخزين مربعات الأعداد من 0 إلى 10 في مصفوفة وطباعتها .

```
#include <iostream.h>
main()
{int sqr[11];
  for(int i=0;i<=10;i++)
  {
    sqr[i]=i*i;
    cout<<"Square "<<i<<"="<<sqr[i]<<"\n";
  }
  return 0;}
```

6. اكتب برنامج لإدخال مصفوفة أعداد صحيحة طولها 15 ومن ثم طباعة أكبر عنصر وأصغر عنصر فيها .

```
#include <iostream.h>
main ( )
{ int x[15] , i, max, min ;
  for (i=0 ; i<15 ; i++)
    {cin >> x[i] ; }
  max=x[0];
  min=x[0];
  for (i=0 ; i<15 ; i++)
  {
    if(x[i]>max) {max=x[i];}
    if(x[i]<min) {min=x[i];}
  }
  cout<<"The max :"<<max <<"The min:"<<min;
  return 0; }
```

7. اكتب برنامج لإنشاء مصفوفة أعداد صحيحة عدد عناصرها 7 قيم هذه المصفوفة عبارة عن العدد 1 اذا كان الدليل عدد فردي والعدد 0 اذا كان الدليل عدد زوجي .

```
#include <iostream.h>
main ( )
{ int num[15] ;
  for (int i=0 ; i<7 ; i++)
  {
    if(i%2!=0)
      {num[i]=1;}
    else
      {num[i]=0;}
  }
  for (int i=0 ; i<7 ; i++)
    {cout<<num[i]<<"\t";}
  return 0;
}
```



# مهارات حاسوب ٢

# Computer Skills 2

صياغة البرامج ولغات البرمجة

**PROGRAMMING & PROGRAMMING LANGUAGES**

المحاضرة الأولى

# صياغة البرامج ولغات البرمجة

بعد أن تكلمنا سابقا عن برامج نظم التشغيل و برامج التطبيقات يبقى سؤال

من كتب هذه البرامج؟

وكيف كتبت؟

وبأي لغة؟

وما هي الخطوات التي اتبعت لبناء هذه البرامج؟

# صياغة البرامج ولغات البرمجة

- البرنامج **Program** هو مجموعة التعليمات المرتبة منطقياً التي توجه الحاسوب لأداء عمل معين على البيانات بهدف الحصول على معلومات مفيدة.
- لغات البرمجة **Programming Languages** هي مجموعة القواعد التي توفر طريقة صياغة تعليمات البرنامج.

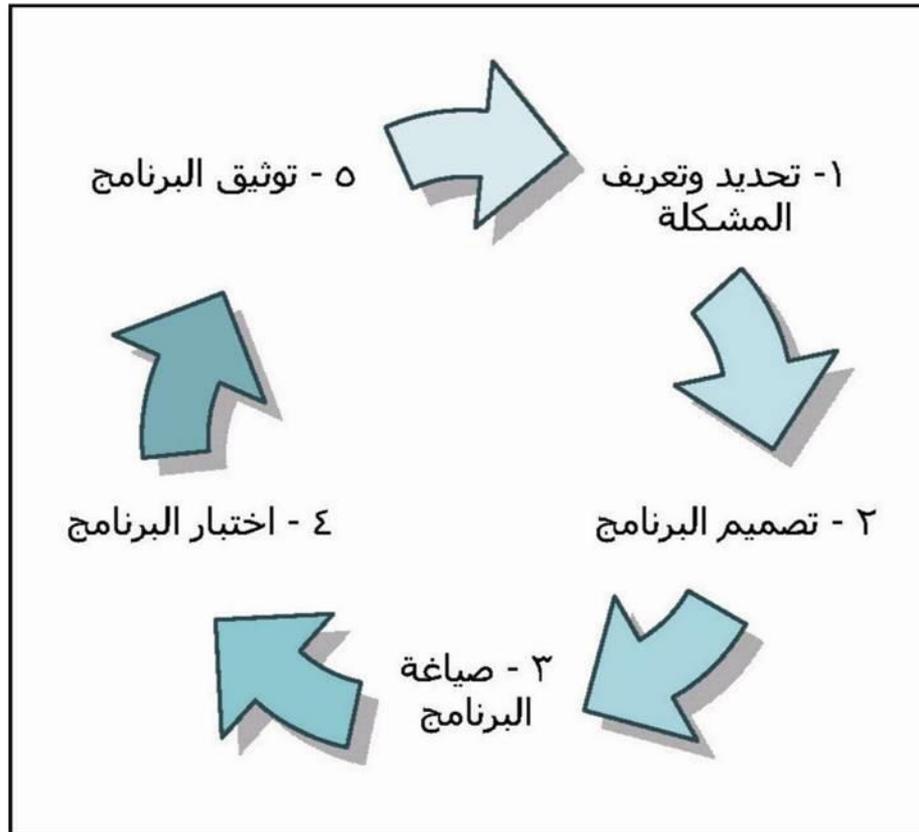
# عمل صائغ البرامج Programmer

يقوم صائغ البرامج (المبرمج) بعدة خطوات لحل مشكلة ما، فيقوم

- أولاً بصياغة المشكلة في صورة تعليمات للحاسوب لحلها،
- ثم ينفذ التعليمات على الحاسوب ويختبر البرنامج لمعرفة مدى صحة النتائج،
- وفي النهاية يكتب تقريراً عن البرنامج.

# خطوات صياغة وتطوير البرامج

## Program Development Steps



مخطط يبين خطوات صياغة وتطوير البرامج

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ١. تحديد وتعريف المشكلة (Defining the Problem)

- في هذه الخطوة يقوم المبرمج بتحديد وتعريف المشكلة وتتضمن هذه الخطوة تحديد التالي بالترتيب:
  - الهدف من البرنامج (حساب ارباح، فواتير استهلاك الماء والكهرباء، أو حساب معدل الطالب التراكمي)
  - نوع وحجم المخرجات ووسائل الإخراج (تقارير – فواتير – شيكات – نقود ...)
  - نوع وحجم البيانات المدخلة ووسائل الإدخال.
  - مستخدمي البرامج والمستفيدين منه.

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ٢. تصميم البرنامج Design the Program

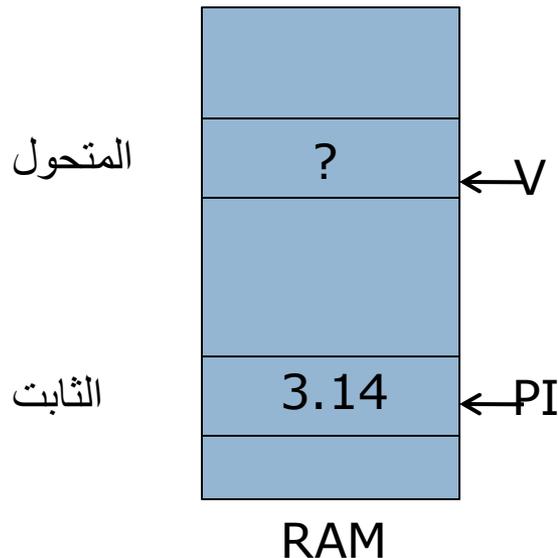
- يتم هنا تحديد المواصفات والخطوات الدقيقة والمرتبة منطقياً والتي تم فهمها ودراستها في الخطوة الأولى.
- ويتم ذلك باستخدام عدة طرق منها:
  - **الطريقة النصية: لغة الخوارزمية Pseudo-code** شبه الترميز
  - **الطريقة البيانية: خرائط التدفق Flowchart** ويطلق عليها أيضاً خرائط سير العمليات وهي مجموعة من الرموز المتعارف عليها تستخدم لتوضيح الخطوات المنطقية اللازمة لحل مشكلة ما.

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ❖ الطريقة النصية: لغة الخوارزمية Pseudo-code شبه الترميز.

- المتحول Variable غرض تتغير قيمته ضمن الخوارزمية
- الثابت Constant غرض لا تتغير قيمته ضمن الخوارزمية
- الصيغة تتألف من متحولات وثوابت وعمليات حسابية أو منطقية
- الشرط: صيغة نتیجتها صح أو خطأ



$$X + 4 * Y \quad \text{الصيغة}$$

$$X > 0 \quad \text{الشرط}$$

# خطوات صياغة وتطوير البرامج

## Program Development Steps

لغة الخوارزمية Pseudo-code شبه الترميز.

### التعليمات :

١. تعليمة القراءة
٢. تعليمة الكتابة
٣. تعليمة الإسناد
٤. التعليمة الشرطية
٥. التعليمة التكرارية

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### التعليمات :

#### ١. تعليمة القراءة

اقرأ < اسم المتحول > : خذ قيمة واحدة من الدخل (لوحة المفاتيح) وضعها في الذاكرة المسماة بـ < اسم المتحول >

#### أمثلة:

اقرأ V ضع القيمة المدخلة في الذاكرة V.

اقرأ A,B ضع القيمتين المدخلتين في الذاكرتين A,B على الترتيب.

# خطوات صياغة وتطوير البرامج

## Program Development Steps

التعليمات :

٢. تعليمة الكتابة

كتابة قيمة معينة على وحدة الخرج (الشاشة) اكتب <صيغة>

أمثلة:

اكتب "The result is"

اكتب  $3+7$

اكتب  $V$ , "The result is"

# خطوات صياغة وتطوير البرامج

## Program Development Steps

التعليمات :

٣. تعليمة الإسناد

تعليمة الإسناد: <صيغة> = <اسم المتحول>

أمثلة:

A=70; B=A

حساب قيمة التابع  $y=x^2+5x+2$  من أجل  $x=4$

حساب قيمة التابع  $y=x^2+5x+2$  من أجل  $x$  قيمة مدخلة من قبل المستثمر.

اقرأ  $x$

$y=x^2+5x+2$

اكتب  $“x=“$ ,  $x,$   $“y=“$ ,  $y$

# خطوات صياغة وتطوير البرامج

## Program Development Steps

التعليمات :

٤. التعليمة الشرطية

أ- إذا < شرط > نفذ

< مجموعة التعليمات >

مثال

إذا  $(N > 0)$  نفذ

$$a = S/N$$

ب- إذا < شرط > نفذ

< مجموعة التعليمات ١ >

وإلا

< مجموعة التعليمات ٢ >

**مثال:** نريد أن يقوم المستثمر بإدخال قيمة ما ،  
يحدد البرنامج هل القيمة تقع ضمن المجال  
[0,10] أم لا .

اقرأ  $a$

إذا  $(a \leq 10)$  and  $(a \geq 0)$  نفذ

اكتب "a is inside the interval [0,10]"

وإلا

اكتب "a is outside the interval [0,10]"

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### التعليمات :

#### ٥. التعليمات التكرارية

مادام <شرط> كرر

< مجموعة التعليمات >

اختبر الشرط إذا كان محققاً نفذ < مجموعة التعليمات > ثم نختبر من جديد إذا كان محققاً نفذ < مجموعة التعليمات > ... عندما يصبح الشرط غير محققاً ننتقل إلى التعليمات التالية لـ مادام

**ملاحظة:** يجب أن تغير < مجموعة التعليمات > متحولات الشرط ليصبح الشرط فيما بعد غير محققاً وإلا سندخل في حلقة لا منتهية .

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### مثال ١

□ حساب مجموع الأعداد من 1 إلى L حيث L قيمة مدخلة من قبل المستثمر.

□ المعطيات L

□ الخرج  $S = 1 + 2 + \dots + L$

□ الخوارزمية:

S=0 المجموع

i=1 عدد عمليات الجمع

اقرأ L

مادام  $(i \leq L)$  كرر

$s = s + i$

$i = i + 1$

اكتب s, "the sum is"

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### مثال ٢

□ نريد حساب مربعات الأعداد من 10 إلى 25 ، الخرج جدول يحوي الأعداد ومربعاتها مسبقاً بتروية

اكتب "قيمة x قيمة y"

x=10

مادام (x<=25) كرر

y=x\*x

اكتب x , " y , "

x=x+1

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### مثال ٣

□ حساب القاسم المشترك الأعظم GCD لعددتين مدخليين.

□ الحل:

١٥ 20 القاسم المشترك الأعظم = ٥

15 5

10 5

٥ 5 القاسم المشترك الأعظم

□ الخوارزمية:

□ قراءة العددين A,B

□ إيجاد القاسم المشترك الأعظم

□ كتابة النتيجة

# خطوات صياغة وتطوير البرامج

## Program Development Steps

اقرأ A,B

مادام  $(A \neq B)$  كرر

إذا  $(A > B)$  نفذ

$A = A - B$

وإلا

$B = B - A$

$GCD = A$

اكتب GCD = "The Greater Common Divider"

# تمارين

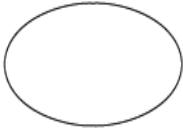
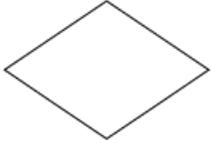
- تمرين ١ : محيط دائرة نصف قطرها 5cm
- تمرين 2: حساب محيط ومساحة مستطيل، طوله و عرضه يتم إدخالهما من قبل المستثمر.
- تمرين ٣: حساب العامل لعدد صحيح n حيث n قيمة مدخلة  
$$n! = n * (n-1) * (n-2) * \dots * 2 * 1$$
- تمرين ٤: حساب مجموع الأعداد الزوجية من 1 إلى L حيث L حيث L قيمة مدخلة من قبل المستثمر.
- تمرين ٥: نريد حساب مكعبات الأعداد من ٥ إلى ٣٥، الخرج جدول يحوي الأعداد ومكعباتها مسبقاً بترويسة

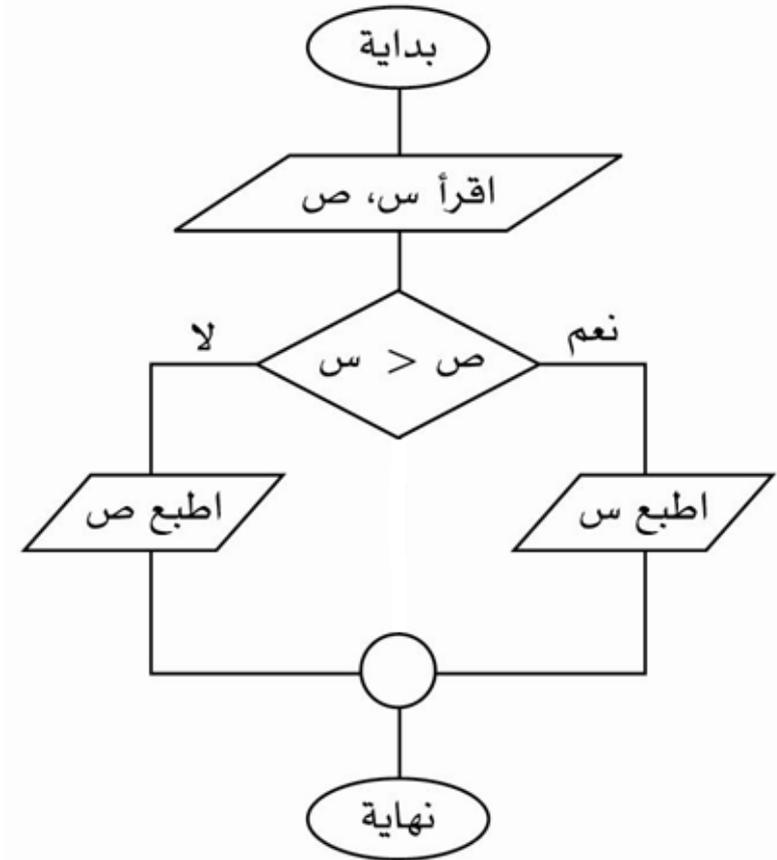
# خطوات صياغة وتطوير البرامج

## Program Development Steps

أهم الرموز المستخدمة في خرائط التدفق

❖ الطريقة البيانية: مخطط التدفق *flowchart*

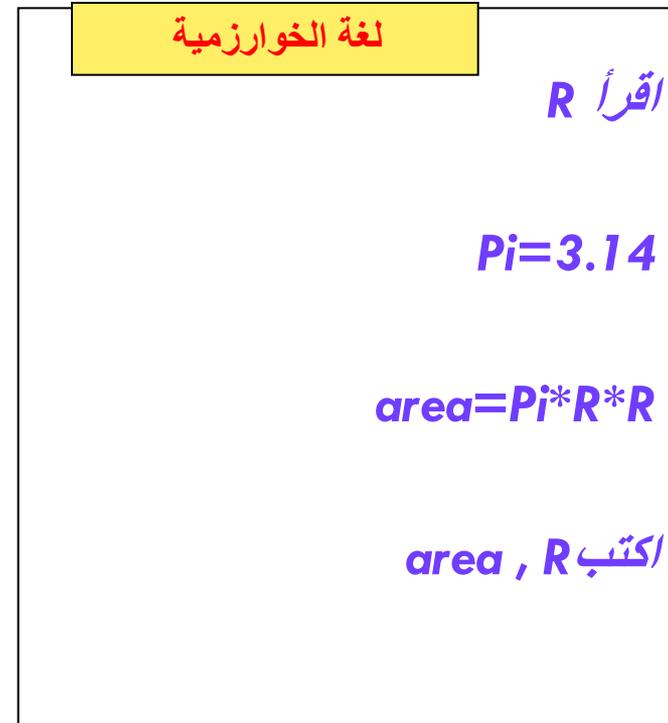
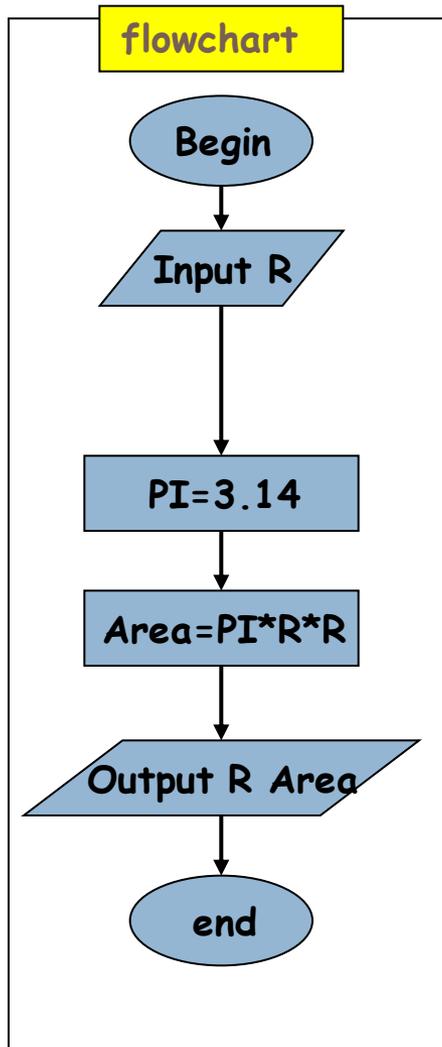
الاسم	الرمز
بداية - نهاية Start - End	
مدخلات - مخرجات Input - Output	
معالجة Processing	
قرار Decision	



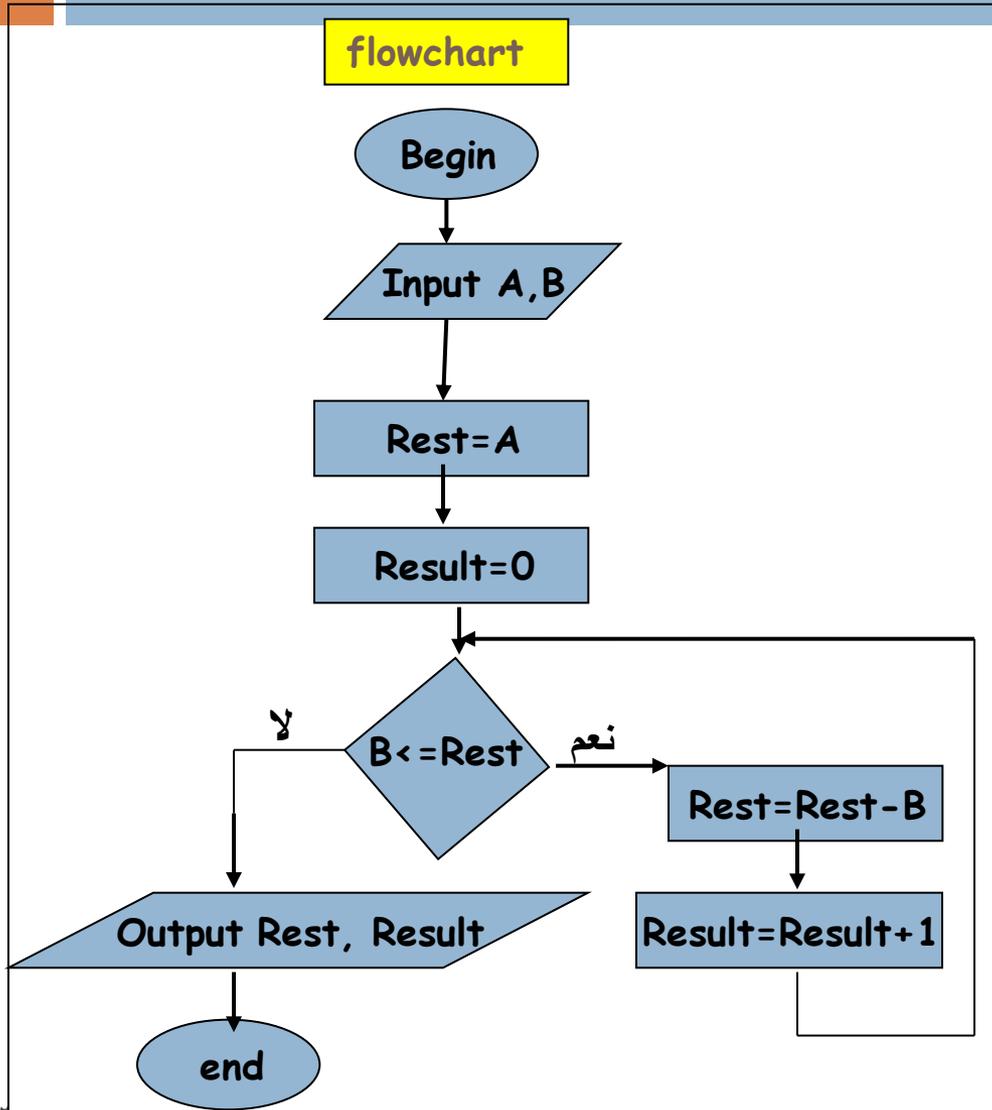
الأسهم تربط بين الأشكال السابقة

اتجاه الأسهم يدل على طريقة تنفيذ التعليمات

# حساب مساحة دائرة نصف قطرها R ، حيث R قيمة مدخلة من قبل المستثمر



# ناتج وباقي قسمة عددين صحيحين مدخلين من قبل المستثمر



لغة الخوارزمية

اقرأ  $A, B$

$Rest = A$

$Result = 0$

مادام  $(B \leq Rest)$  كرر

$Rest = Rest - B$

$Result = Result + 1$

اكتب  $Result, Rest$

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ٣. صياغة البرنامج Coding the Program

- بعد الانتهاء من تصميم البرنامج يتم اختيار إحدى لغات البرمجة المناسبة لصياغة أوامر البرنامج Coding وذلك بالاستعانة بخريطة التدفق Flow Chart أو غيرها.
- يجب عند صياغة البرنامج اتباع قواعد صيانة لغة البرمجة المستخدمة حيث ان لكل لغة برمجة قواعد خاصة بها ولا يعمل البرنامج اذا كان هنالك اخطاء املائية او اخطاء في قواعد اللغة Syntax Errors.

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ٤. اختبار البرنامج وتصحيح الأخطاء Program Debugging and Testing

□ خلال عملية الترجمة Compilation قد تظهر اخطاء في صياغة البرنامج المصدر ينبغي على المبرمج تصحيحها.

□ هناك ثلاث انواع من الأخطاء:

١. **اخطاء في قواعد اللغة Syntax Errors:** اخطاء املائية في كتابة الأوامر.
٢. **اخطاء منطقية Logical Errors:** لا يكتشفها الحاسوب وتظهر عند تنفيذ البرنامج على عينه من البيانات فنحصل على نتائج خاطئه او غير متوقعة، ويقوم المبرمج بتتبع خطوات البرنامج لمعرفة مصدر الخطأ وتصحيحه وتسمى هذه العملية Tracing.
٣. **اخطاء اثناء التشغيل Run-Time Errors:** تظهر عند تنفيذ البرنامج مثل عدم حجز مساحة كافية للمدخلات او الدخول في دوران بلا نهاية، وتظهر رسالة بنوع الخطاء.

# خطوات صياغة وتطوير البرامج

## Program Development Steps

### ٥. توثيق البرنامج Documenting the Program

- في هذه المرحلة تتم كتابة وصف تفصيلي لصياغة البرنامج، ويشمل هذا التوثيق أصل المشكلة وخطوات الحل وخرائط الحل وتعليمات التشغيل ومتطلبات التشغيل والمدخلات والمخرجات وكيفية التحكم في البرنامج في المواقع المختلفة

# تصنيف لغات البرمجة

تصنف لغات البرمجة إلى ثلاثة أنواع هي:

١. لغات برمجة ذات مستوى منخفض Low Level Languages
٢. لغات برمجة ذات مستوى عال High Level Languages
٣. لغات الجيل الرابع Fourth Generation Languages

# تصنيف لغات البرمجة

## ١. لغات البرمجة ذات المستوى المنخفض Low Level Languages

□ تعتبر لغات البرمجة ذات المستوى المنخفض من أوائل لغات البرمجة ومنها:

□ لغة الآلة Machine Language

□ لغة التجميع Assembly language

□ ميث باللغات المنخفضة المستوى نظراً لأن المبرمجين يكتبون أوامر البرنامج

بمستوى قريب من مستوى فهم الآلة (الحاسوب)، حيث تستخدم هذه اللغة (٠ ، ١)

في كتابة البرامج.

# تصنيف لغات البرمجة

## ٢. لغات البرمجة ذات المستوى العالي High Level Languages

- سميت بهذا الاسم لأنه أصبح بإمكان المبرمج كتابة البرامج دون معرفة تفاصيل كيفية قيام الحاسب بهذه العمليات، كمواقع التخزين و تفاصيل الجهاز الدقيقة.
- تعبيرات اللغات ذات المستوى العالي شبيهة الى درجة كبيرة باللغة التي يستخدمها الانسان في التخاطب و التواصل مع الاخرين
- تتميز بسهولة اكتشاف الاخطاء و تصحيحها و يمكن تشغيلها على اكثر من جهاز.  
كما يمكن استخدام أكثر من لغة برمجية على جهاز واحد.

# تصنيف لغات البرمجة

## ٣. لغات الجيل الرابع Fourth Generation Languages

- تسمى هذه اللغات أيضاً باللغات عالية المستوى بصورة كبيرة جداً **Very High Level Languages** حيث إنها لغات سهلة الاستخدام والفهم وقريبة جداً من لغة الإنسان.
- يستطيع المبرمج القيام بكثير من العمليات بسهولة تغنيه عن صياغة **Coding** صفحات عديدة من أوامر البرنامج. ويهتم المبرمج بماذا يريد من الكمبيوتر دون ان يوجهه بكيفية القيام بذلك.
- من لغات الجيل الرابع: SQL , DBase

# أنواع لغات البرمجة

١. لغة البيسك BASIC Language ولغة فيجوال بيسك Visual Basic
٢. لغة الجافا Java Language
٣. لغة الكوبل COBOL Language
٤. لغة الباسكال PASCAL Language
٥. لغة اللوجو LOGO Language
٦. لغات الذكاء الاصطناعي Artificial Intelligence Languages
٧. لغة سي ولغة سي بلس بلس C & C++ Language

# أنواع لغات البرمجة

## ١. لغة البيسك BASIC Language ولغة فيجوال بيسك Visual Basic

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

# أنواع لغات البرمجة

## ٢. لغة الجافا Java Language

- تعتبر لغة الجافا من اللغات عالية المستوى وتعرف بأنها من اللغات المرئية Visual والشيئية Objects، وهي من اللغات العامة الأغراض والتي تستخدم لإنتاج برمجيات متنوعة.
- تشبه لغة الجافا لغة ++C إلا أنها تتسم بالسهولة.

# أنواع لغات البرمجة

## ٣. لغة الكوبل COBOL Language

□ تستخدم هذه اللغة بصفة رئيسية في الأعمال التجارية مثل البنوك والشركات.

□ هي لغة واسعة الانتشار، وكلمة كوبل مشتقة من الكلمة

### Common Business Oriented Language

□ وبدأ ظهور هذه اللغة سنة ١٩٥٩ وقد أجري عليها عدة تعديلات لزيادة كفاءتها وكان آخرها سنة ١٩٧٤.

# أنواع لغات البرمجة

## ٤ . لغة الباسكال PASCAL Language

- سميت نسبة إلى العالم الفرنسي في علم الحاسوب Blaise Pascal و يرجع تاريخها إلى ١٩٧٣ وتستخدم للأغراض العامة وكلغة تعليمية.
- على الرغم من وضوح بنائها إلا أنها أصعب في التعلم من لغة البيسك.
- تعتبر لغة باسكال من لغات البرمجة الرئيسية التي تدرس لطلبة المدارس والكليات نظرا لوضوح السمات الأساسية لتخطيط البرامج البنائية بها  
.Structured Programming

# أنواع لغات البرمجة

## ٥. لغة اللوجو LOGO Language

- هي لغة تطبيقات علمية تتميز ببساطة وسهولة تعلمها وقد صممت خصيصا ليستخدمها الأطفال فهي تشجع على الإستخدام المنطقي والتركيبى.
- تعتمد هذه اللغة على استخدام روبوت صغير يسمى بالسلحفاة Turtle من أجل ابراز استعمالاتها كتعلم الأفكار الحسابية مثل الزوايا والقياسات.

# أنواع لغات البرمجة

## ٦. لغات الذكاء الاصطناعي Artificial Intelligence Languages

□ هي لغات خاصة بإنتاج حاسبات ذكية تحاكي الإنسان في قدراته الحركية والبصرية والتحليل والاستنتاج واتخاذ القرارات بناء على نظم الخبرة التي ستغذي بها الحاسبات.

□ من أهم هذه اللغات:

□ لغة برولوج **Prolog**: يطلق عليها اسم لغة البرمجة المنطقية **Programming in Logic**.

□ لغة ليسب **Lisp**: يطلق عليها اسم لغة برمجة القوائم **List Programming Language**.

# أنواع لغات البرمجة

## ٧. لغة سي ولغة سي بلس بلس C & C++ Language

- تتميز 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.



جامعة حماة  
كلية الاقتصاد  
السنة الثانية

# مهارات حاسوب 2 القسم العملي

م. رولا عبود

م. لمى أزهري

الجلسة الأولى

# تمهيد

□ حالة 1:

لنفرض عزيزي الطالب أنه وفي فترة الانقطاع هذه ولدى دراستك لمقرر محاسبة مثلا وجدت مسألة مطروحة مع الحل بعد عناء ستتوجه لمدرس المقرر أو لأحد أصدقائك بسؤال مهم :

ماهي طريقة الحل

وقد يكون الجواب كالتالي:

✓ ترسم جدول من أربع أعمدة

✓ في العمود الأول نضع المبلغ

✓ في العمود الثاني لمن انتقلت النقود

✓ في العمود الثالث المبلغ المنتقل

✓ في العمود الرابع من من تحركت النقود

✓ عليك ان تقرأ المسألة

✓ تضع خط تحت المعلومات المهمة لحركة النقود

✓ وهكذا .....

الحاسبة (1)

رابعاً: الحسابات الختامية في المنشآت التجارية

محللات الوفاء

منه المصروفات د/ المتاجرة عن السنة المنتهية في ٢٠/١٢/١٤٢٢ هـ الإيرادات له

بيان	مبلغ	بيان	مبلغ
من د/ المبيعات	١,٢٦٠,٠٠٠	إلى د/ المخزون (أول المدة)	١١١,٠٠٠
من د/ مردوبات المشتريات	٤٢,٠٠٠	إلى د/ المشتريات	٨٩٤,٠٠٠
من د/ مسموحات المشتريات	٦,٠٠٠	إلى د/ مصروفات نقل المشتريات (4) مشتريات	٣١,٥٠٠
من د/ خصم مكتسب	١٠,٥٠٠	إلى د/ مردوبات المبيعات	٧٥,٠٠٠
		إلى د/ مسموحات المبيعات	





shutterstock.com • 586937405

## تمهيد

□ حالة 2:

لصناعة قالب من الكاتو نحتاج المكونات والطريقة

مثلا نحن بحاجة لـ:

✓ 4 بيضات كاس سكر كأسين دقيق ملعقة صغيرة فانيليا ملعقة صغيرة بيكنغ باودر كأس حليب

✓ تخلط المقادير في وعاء عميق

✓ توضع في القالب

✓ تدخل الفرن على حرارة 180

✓ بعد نصف ساعة تخرج من الفرن وتوضع في صحن التقديم

## تمهيد

قد يتبادر إلى ذهنك عزيزي الطالب ما علاقة  
مادة مهارات الحاسوب بالحالة الأولى أو  
الثانية 😊

الجواب:

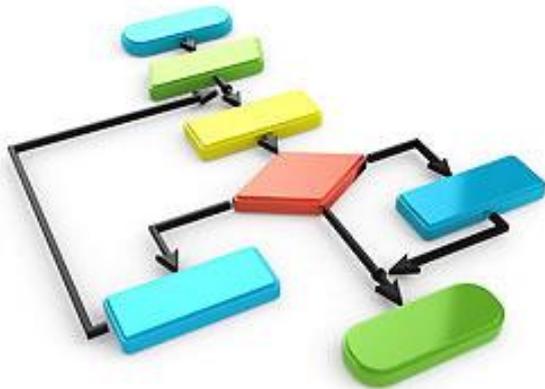
إن ماورد في الحالتين عبارة عن طريقة لحل  
مشكلة أو تحقيق هدف ما وبأسلوب مفهوم  
من المتلقي بعيد عن التوثيق والقوانين  
والرياضيات وهذا مانسميه في مقررنا  
بخوارزمية الحل 😊

أي أننا نسرد خطوات حل المسألة أو صناعة  
قالب الكيك بطريقة سهلة واضحة ومفهومة  
ودون الالتزام بقوانين أو قواعد



# تعريف الخوارزمية

- وتسمى أيضا شبه الترميز أو **psuedo code**
- تعرف بأنها مجموعات خطوات لحل مسألة معينة دون الالتزام بلغة برمجة
- يمكن التعبير عنها كتابيا (في مقررنا سنختار كلمات محددة لغرض توحيد الحل فقط )
- يمكن التعبير عنها بمخططات رمزية



# أقسام الخوارزمية

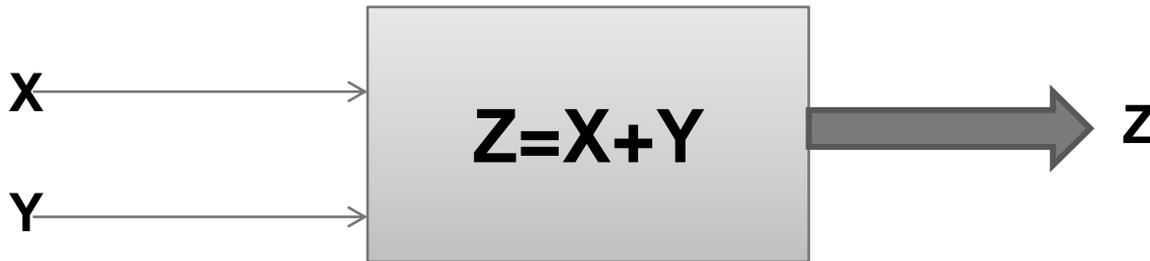
- المدخلات : مايلزمنا كي تبدأ الخوارزمية في العمل
- ✓ في مثال الكيك تعتبر المكونات من بيض وطحين وسكر ...مدخلات الخوارزمية
- ✓ أما في حال خوارزمية لجمع عددين فإن المدخلات هي العدد الاول والثاني
- ✓ وكما يلزمنا لمكونات الكيك كأس أو ملعقة للمكونات ففي خوارزمية جمع عددين يلزمنا متحولين لتخزين العددين مثلا  $X$  و  $Y$
- ✓ إن تحديد المدخلات هو أول خطوة في كتابة الخوارزمية.
- المعالجة : وهي العمليات التي سنجريها على المدخلات
- ✓ في مثال الكيك عملية الخلط والسكب والوضع بالفرن واخراج الكيك
- ✓ في جمع العددين عملية المعالجة في عملية الجمع بحد ذاتها أي  $X+Y$

# أقسام الخوارزمية

□ المخرجات : ناتج الخوارزمية

✓ في مثال الكيك هو الكيك الناضج بعد ذاته موضوع في صحن التقديم

✓ في خوارزمية الجمع هو ناتج جمع العددين ونخزنه في متحول خرج مثلا Z



# مفردات الجلسة

□ وأكرر أن لدينا في الخوارزمية حرية التعبير عن الخطوة فمثلا كلمة اقرأ أو أدخل متساوية وتعطي نفس النتيجة

✓ عملية القراءة للمدخلات سنعتبر عنها بـ اقرأ

✓ عملية الاخراج سنعتبر عنها بـ اكتب

✓ الشرط نعتبر عنه خلال عملية المعالجة بـ:

إذا كان ( شرط ) نفذ

تعليمات

وإلا إذا كان ( شرط ) نفذ

تعليمات

وإلا نفذ

تعليمات

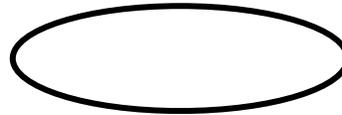
# أشكال المخطط التدفقي



الادخال و الاخراج



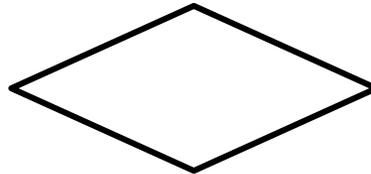
عملية معالجة



البداية والنهاية



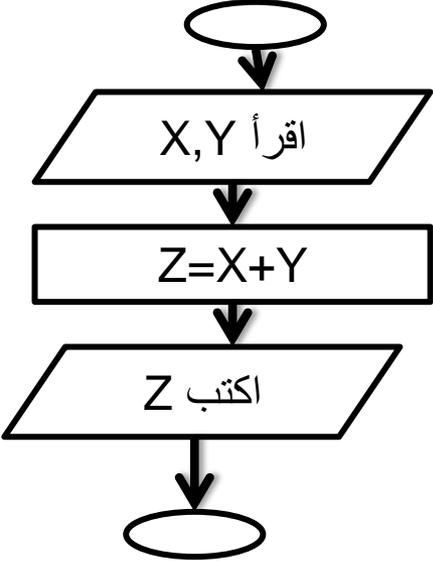
الربط



الشرط

# تطبيقات

## خوارزمية جمع عددين

المخطط التدفقي	الطريقة الكتابية
 <pre>graph TD; Start([ ]) --&gt; Read[/اقرأ X, Y/]; Read --&gt; Process[Z=X+Y]; Process --&gt; Write[/اكتب Z/]; Write --&gt; End([ ])</pre>	<p>اقرأ X , Y <math>Z=X+Y</math> اكتب Z</p>

□ خوارزمية تطلب اسمك وتطبعه على الشاشة

الحل:

✓ نحن بحاجة لمتحول نخزن فيه الاسم حتى نطبعه

✓ طريقة أولى:

المدخلات : الاسم N

المخرجات : الاسم N

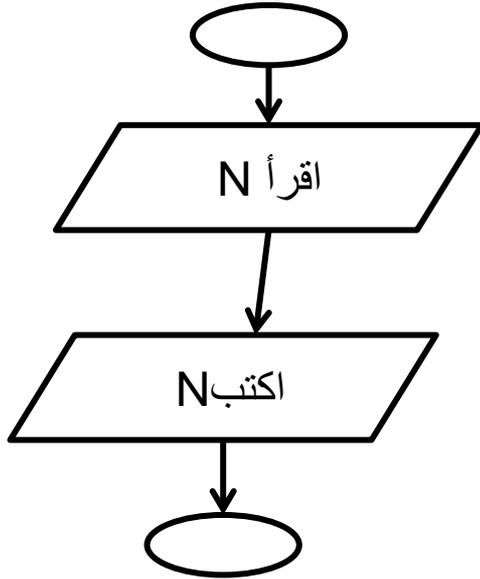
✓ طريقة ثانية:

اقرأ N

اكتب N

□ المخطط التدفقي

الحل:



## خوارزمية لطباعة مساحة ومحيط مستطيل.

الحل:

✓ نعرف أن لحساب مساحة ومحيط المستطيل نحتاج الطول والعرض لذلك المدخلات هي الطول والعرض ونخزنها في متحولين  $H$  و  $W$

✓ قوانين المساحة والمحيط هي المعالجة وتخزن النتيجة في متحولين المساحة  $A$  والمحيط  $S$

✓ المخرجات هي  $S, A$

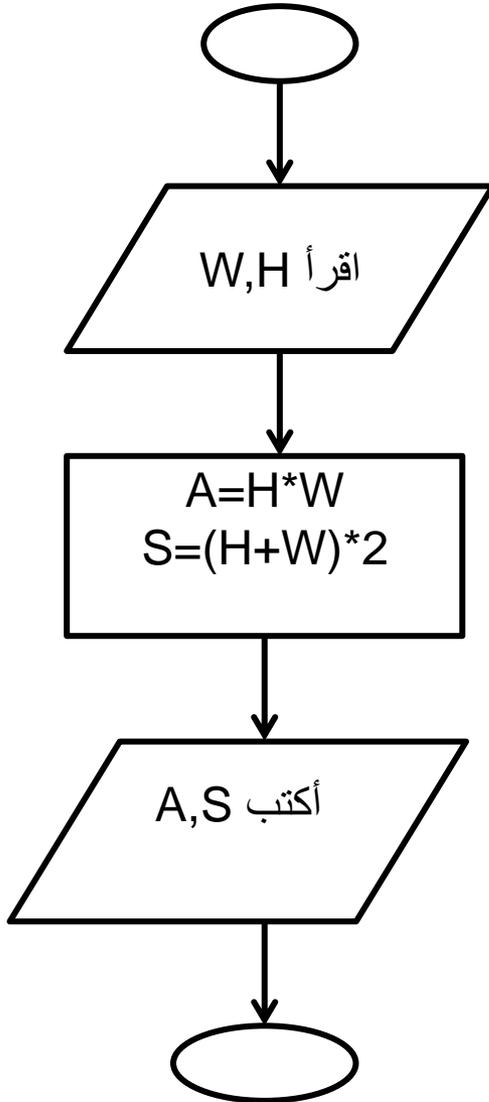
✓ الخوارزمية المطلوبة

اقرأ  $W, H$

$A = H * W$

$S = (H + W) * 2$

اكتب  $A, S$



□ خوارزمية لطباعة فيما اذا كان العدد المدخل زوجي أم فردي

□ الحل:

✓ المدخلات : العدد X

✓ المعالجة : هل العدد يقبل القسمة على 2 أم لا

✓ المخرجات: كلمة زوجي أو كلمة فردي

✓ الخوارزمية :

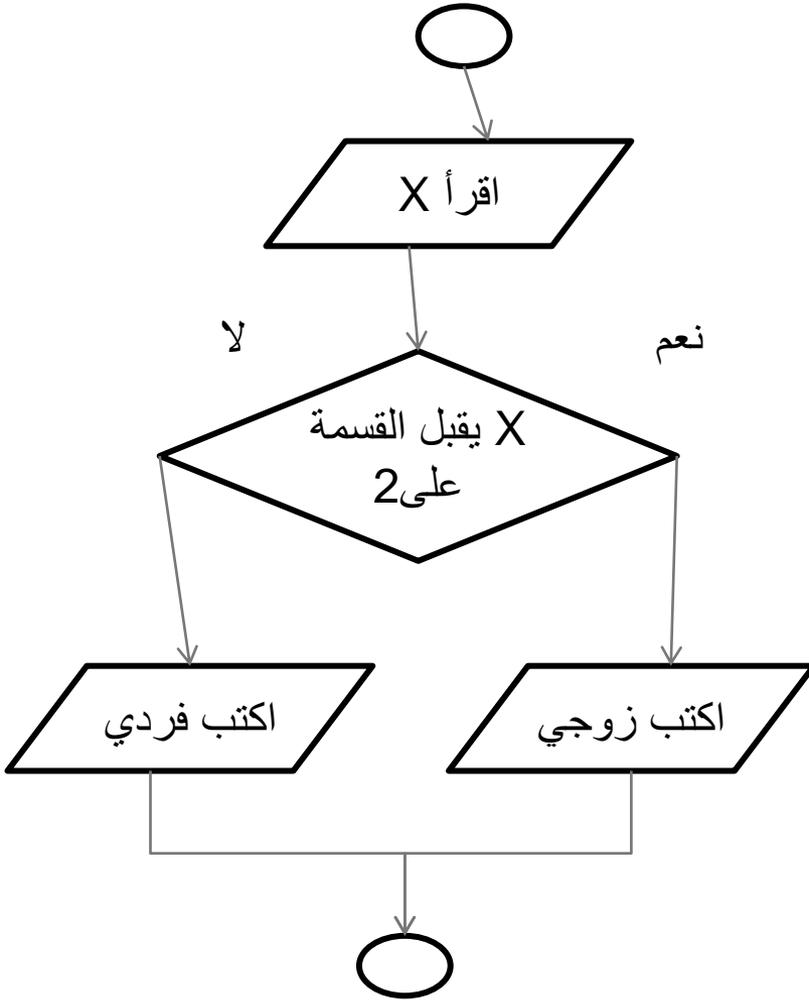
اقرأ X

اذا كان ( X يقبل القسمة على 2 ) نفذ

اكتب زوجي

و إلا نفذ

اكتب فردي



ملاحظة على التمرين السابق وتفيد في الجلسات اللاحقة

يقبل عدد  $x$  القسمة على عدد  $y$  اذا كان باقي قسمة  $x$  على  $y$  يساوي الصفر و نرسم عملية باقي القسمة عادة بالرمز %

أي

10 تقبل قسمة على 2 لأن  $10\%2=0$

أي أن باقي قسمة 10 على 2 يساوي الصفر

أما 11 لايقبل القسمة على 3 لأن

$$11\%3=2$$

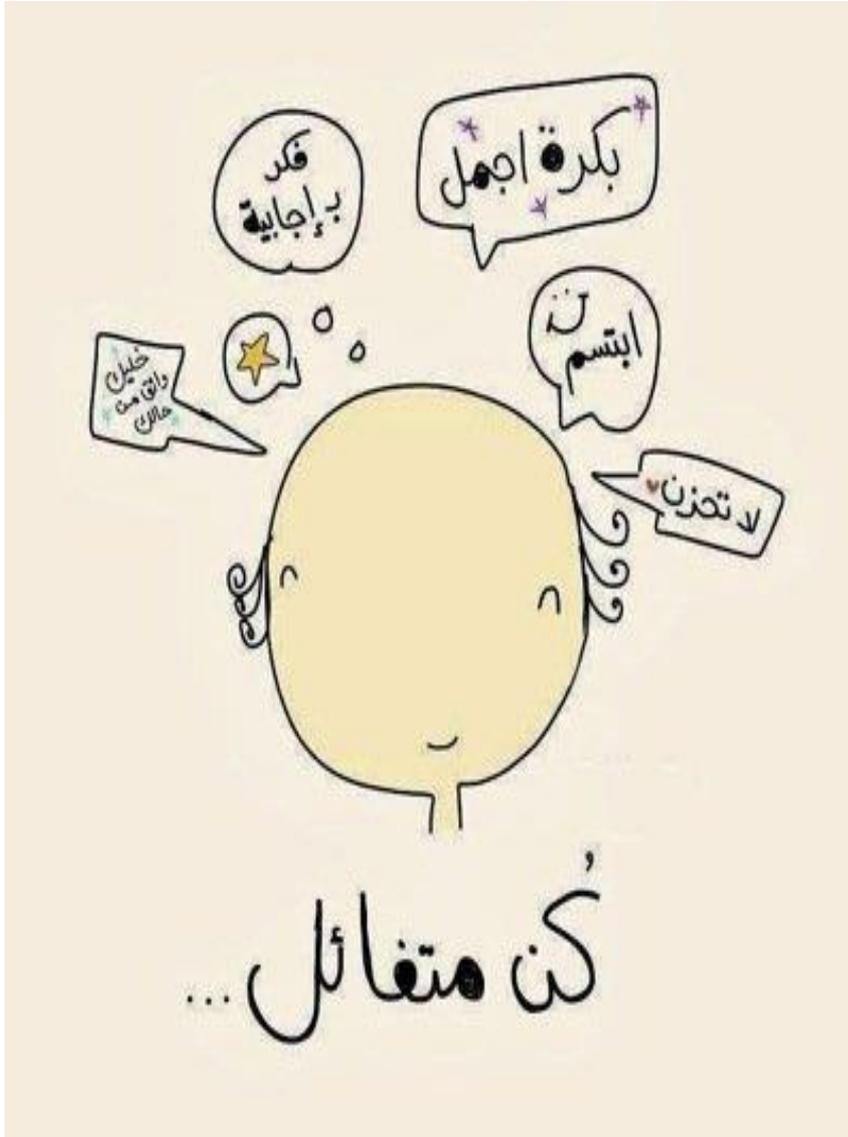
2 هو الباقي وليس ناتج القسمة

# تمارين للحل خلال الجلسة

- ❖ خوارزمية تطبع العدد سالب أم موجب أم صفر
- ❖ خوارزمية تطلب اسمك وتطبعه عشر مرات
- ❖ خوارزمية تطبع فيما إذا كان عدد أول من قواسم عدد ثاني أم لا
- ❖ خوارزمية تأخذ علامة الطالب وتطبع فيما إذا كان ناجح أم راسب علما أن علامة النجاح

# عزيزي الطالب

- ما ورد سابقا هو شرح يعرض عادة في المخبر وليس توثيق أو مرجع للجلسة كالعادة .
- مرجعك للجلسة محاضرة النظري المنشورة سابقا.
- يرجى حل التمارين السابقة ومناقشتها على منشور الجلسة في التعليقات على الصفحة الرسمية للكلية
- ودمتم سالمين ☺





جامعة حماة  
كلية الاقتصاد  
السنة الثانية

# مهارات حاسوب 2 القسم العملي

م. رولا عبود

م. لمى أزهر

الجلسة الثانية

# التكرار

□ حالة 1:

نفرض عزيزي الطالب أنه طلب منك خوارزمية لطباعة اسمك خمس مرات كما ورد في الجلسة السابقة سيكون الحل ببساطة:

المدخلات : الاسم ويخزن في متحول  $N$

المعالجة : لا يوجد

المخرجات: المتحول  $N$  خمس مرات

تكون الخوارزمية

اقرأ  $N$

اكتب  $N$

اكتب  $N$

اكتب  $N$

اكتب  $N$

اكتب  $N$

□ بتفكير آخر

قراءة الاسم ثم تكرار عملية الطباعة ل 5 مرات أي كل مرة سنطبع سنقوم بالعد من 1 و حتى 5

يتم التعبير عن هذا التفكير بالخوارزمية كالتالي:

اقراً N

$i=1$  (هذا عداد من أجل مرات الطباعة)

طالما  $i < 5$

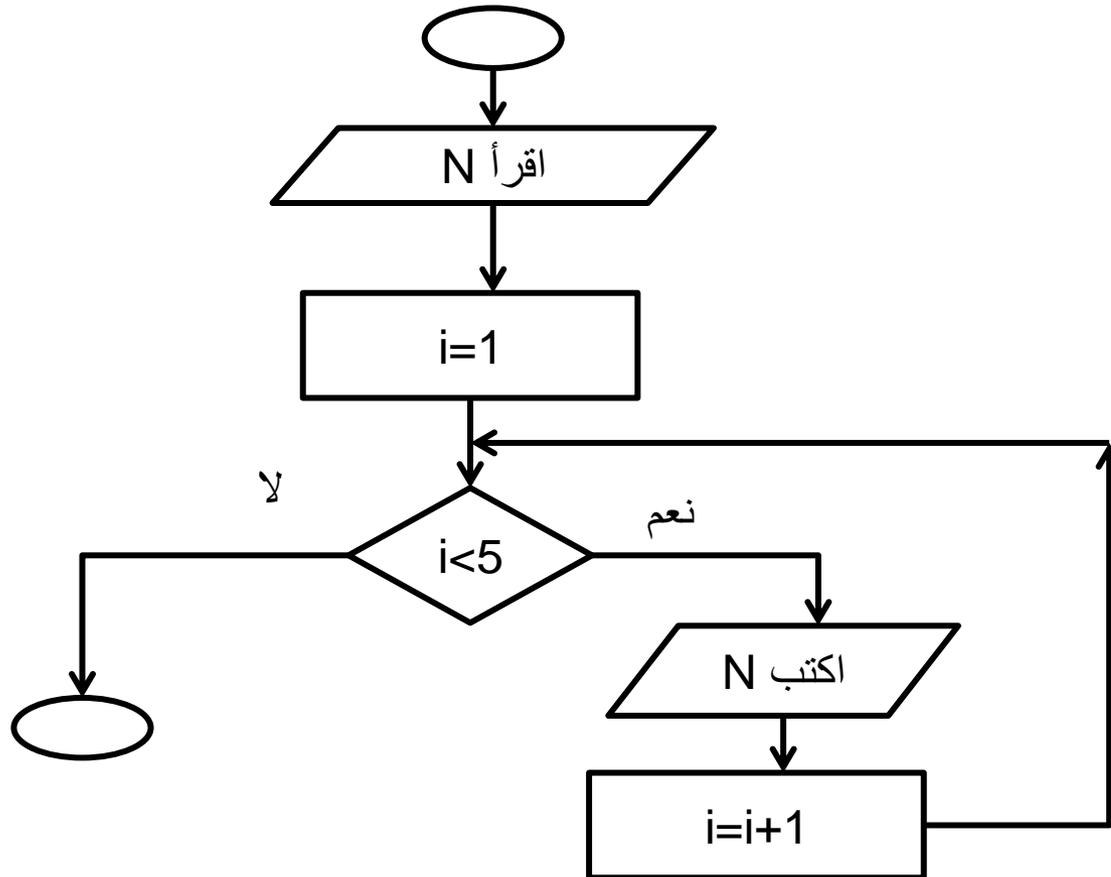
اكتب N

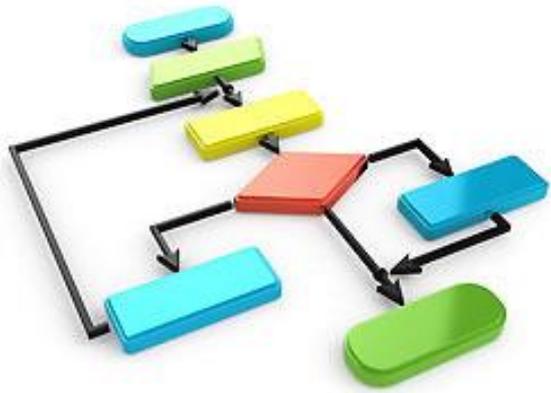
$i=i+1$  (زيادة العداد عند كل تكرار)

نهاية التكرار

# المخطط التدفقي

نستخدم للتعبير عن التكرار معين قرار مع العودة اليه بعد اتمام العمليات التي ستتكرر





ComputerHope.com

# تمارين



# مطلوب خوارزمية لطباعة الأعداد من 1 الى 10

الحل الكتابي:

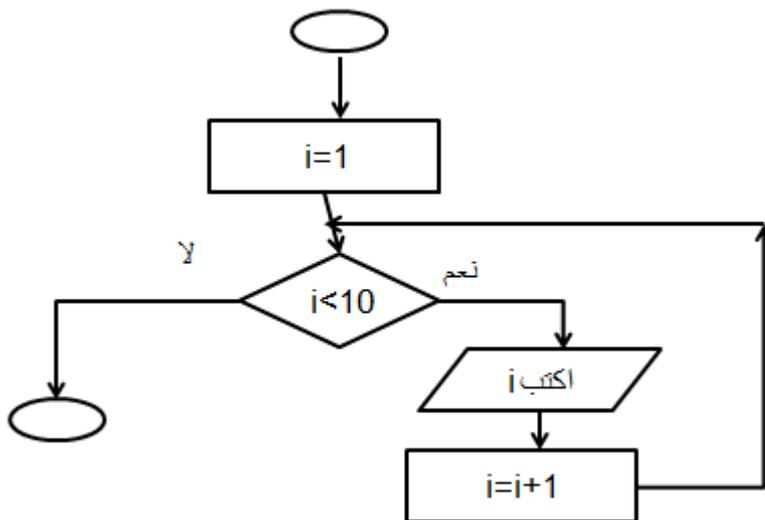
$i=1$

طالما  $i < 10$  كرر

اكتب  $i$

$i=i+1$

نهاية التكرار



# مطلوب خوارزمية لطباعة الأعداد الزوجية بين 1 الى 10

الحل الكتابي:

$i=1$

طالما  $i < 10$  كرر

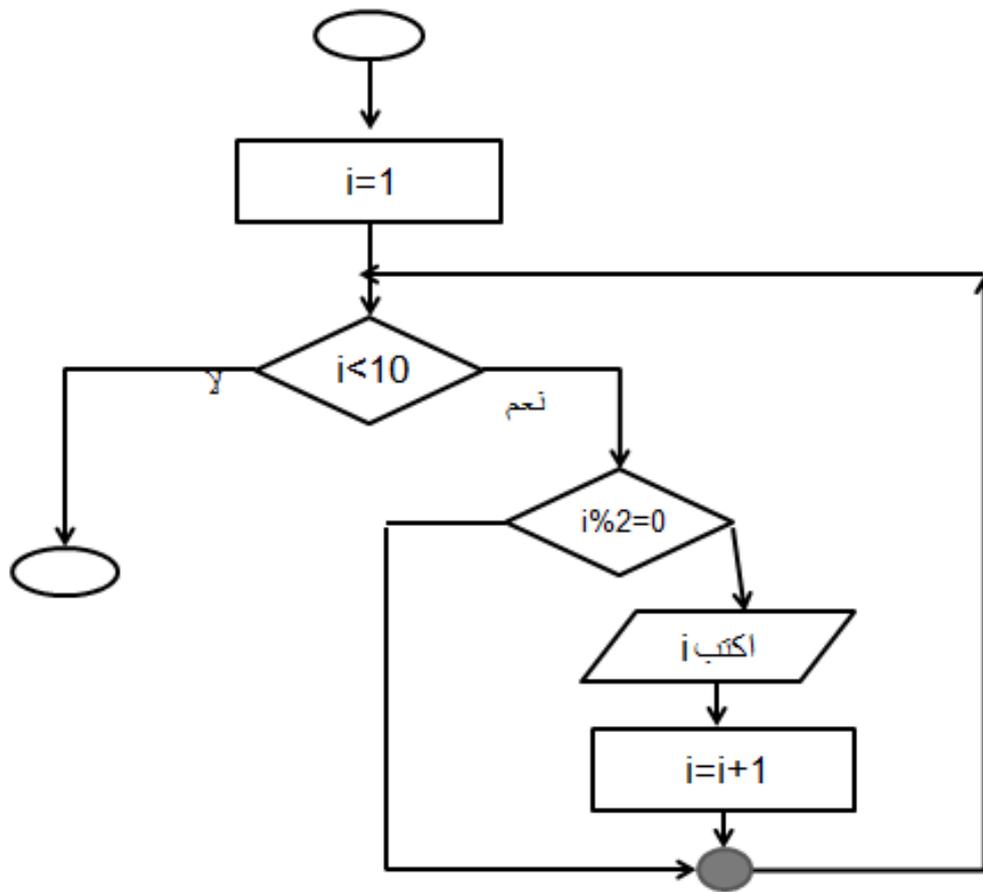
إذا كان  $i \% 2 = 0$  نفذ

اكتب  $i$

نهاية الشرط

$i=i+1$

نهاية التكرار



## مطلوب خوارزمية لايجاد القاسم المشترك لعددين بطريقة اقليدس

تتلخص فكرة الطريقة بأنه طالما العددين غير متساويين نستبدل الكبير بالفارق بين العددين وذلك حتى الوصول إلى التساوي عندها يكون أحد العددين هو القاسم المشترك الأكبر

مثال: العددين 18 و 30

<u>30</u>	18
$30-18=12$	<u>18</u>
<u>12</u>	$18-12=6$
$12-6=6$	6

القاسم المشترك الأكبر هو 6

# الحل

الحل الكتابي:

اقرأ  $X, Y$

طالما  $X \neq Y$  كرر

إذا كان  $X > Y$  نفذ

$X = X - Y$

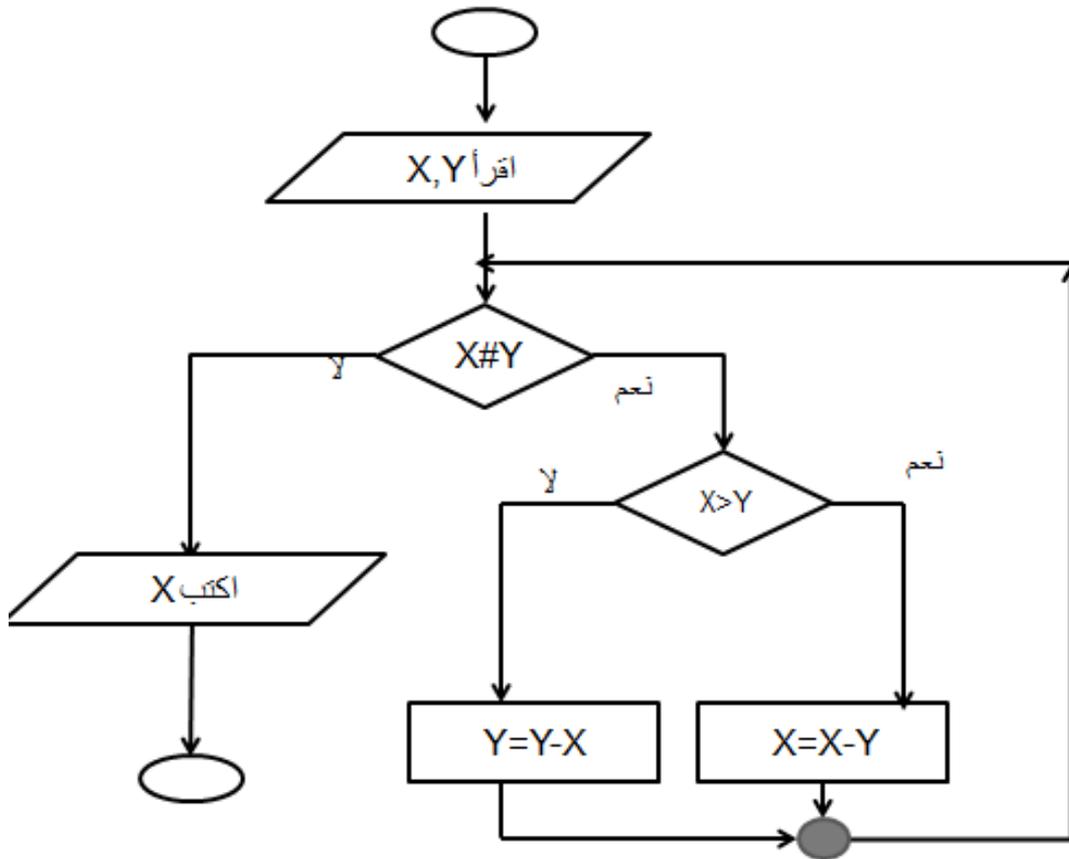
وإلا

$Y = Y - X$

نهاية الشرط

نهاية التكرار

اكتب  $X$



# تمارين للحل خلال الجلسة

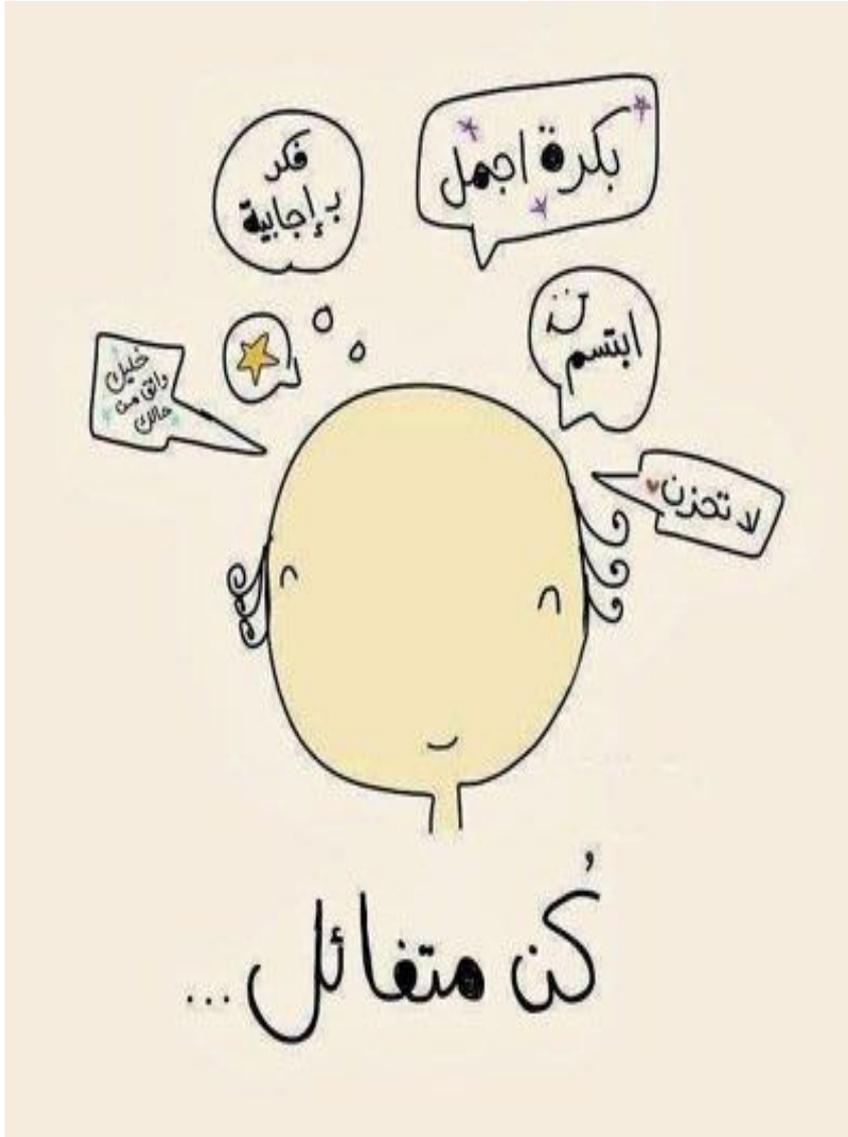
❖ خوارزمية تطبع قواسم العدد 124

❖ خوارزمية تطبع قواسم عدد يدخله المستخدم

❖ خوارزمية تطلب علامات 5 طلاب واسماؤهم وتطبع اسم الطالب وهل هو ناجح أم راسب

# عزيزي الطالب

- ما ورد سابقا هو شرح يعرض عادة في المخبر وليس توثيق أو مرجع للجلسة كالعادة .
- مرجعك للجلسة محاضرة النظري المنشورة سابقا.
- يرجى حل التمارين السابقة ومناقشتها على منشور الجلسة في التعليقات الصف 4ejqu44
- ودمتم سالمين ☺



## برنامجك الأول بلغة C++

في الجلسات السابقة قمنا بتوضيح مفهوم الخوارزمية والتي هي مجموعة من التعليمات التي تقوم بأداء مهمة معينة ولكنها مكتوبة دون قواعد او كلمات محددة .

يمكن استخدام اللغة المحكية او الرسومات او المخططات التدفقية لكتابة الخوارزميات .  
تعتبر الخوارزمية الخطوة السابقة لكتابة البرنامج.

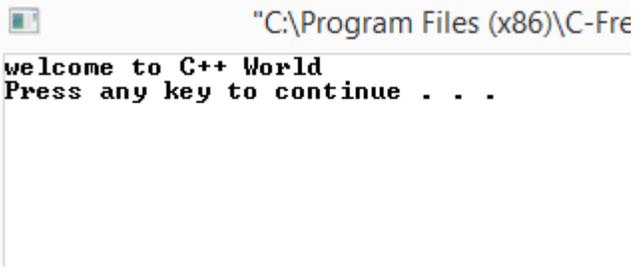
### البرنامج:

هو مجموعة من التعليمات لاداء مهمة معينة مكتوبة بلغة برمجة محددة أي تتبع قواعد وتستعمل كلمات ثابتة حسب هذه اللغة

ماسنقوم به هو اسقاط للخوارزميات على لغة C++

### البرنامج الأول:

طباعة عبارة welcome to C++ World

الخوارزمية	البرنامج
بداية اطبع " welcome to C++ World " نهاية	<pre>#include&lt;iostream.h&gt; void main() { cout&lt;&lt;"welcome to C++ World\n"; }</pre>
	

### شرح البرنامج

السطر الأول: #include<iostream.h>

هذه العملية تضمين لمكتبة عمليات الادخال والايخراج التي تسمى iostream ولولا وجود هذا السطر لسألك المترجم مامعنى كلمة cout التي تمثل عملية الطباعة .

يمائل هذا اسطر في عمله طلبك لوجبة سريعة من مطعم توجد لديه الوجبات جاهزة ولاداعي لاعدادها في المنزل كذلك هذه المكتبة تحوي مجموعة من العمليات الجاهزة فيكفي طلبها بدلا من اعادة برمجتها من جديد.

السطر الثاني: void main()

وهو اسم البرنامج الرئيسي الذي يبحث عنه المترجم لينفذ تعليماته

السطر الثالث : قوس بداية البرنامج.

السطر الرابع : تعليمة الطباعة <<cout.

في حالة طباعة قيمة متحول يكفي ذكر اسمه دون علامات تنصيب أما عند طباعة نص فيجب وضعه ضمن علامات تنصيب .

السطر الخامس : قوس نهاية البرنامج.

### ملاحظة :

تضمنت الجملة النصية علامة \n بينما لم تظهر في نتيجة تنفيذ البرنامج فما السبب

السبب هو أن هذه العلامة \ تسمى حرف الهروب أو سلاسل الهروب حسب الحرف المستخدم فوضع الحرف n كما في المثال تعني نقل المؤشر الى السطر التالي بعد طباعة ما قبله .

<pre> Untitled4.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 cout&lt;&lt;"welcome to \n C++ World\n"; 5 } </pre>	<pre> "C:\Program Files (x86)\C-F welcome to C++ World Press any key to continue . . . </pre>
<pre> Untitled4.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 cout&lt;&lt;"welcome to \t  C++ World\n"; 5 } </pre>	<pre> "C:\Program Files (x welcome to      C++ World Press any key to continue . . . </pre>

<pre> Untitled4.cpp* 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 cout&lt;&lt;"welcome to \" C++ \" World\n"; 5 } </pre>	<pre> "C:\Program Files (x8 welcome to " C++ " World Press any key to continue . . . </pre>
--	---

### التصريح عن المتحولات :

char  
short int , int , long int  
float , double , long double  
bool  
const

➤ المتحول المحرفي  
➤ المتحولات الصحيحة  
➤ المتحولات الحقيقية  
➤ المتحول المنطقي  
➤ الثابت (ويضاف لنوع المتحول)

### تعلية الادخال :

في C++ تعليمة الادخال هي <>cin وبعدها اسم المتحول الذي سنخزن به القيمة المدخلة من المستخدم .

مثال: برنامج يطلب من المستخدم ادخال رقم ويطبعه على الشاشة

<pre> Jntitled4.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 cout&lt;&lt;"Enter number \n"; 5 int N; 6 cin&gt;&gt;N; 7 cout&lt;&lt;N; 8 cout&lt;&lt;"\n"; 9 } </pre>	<p> بداية  اطبع " Enter number"  اقرأ N  اطبع N  نهاية </p>
--	---

```

"C:\Program Files (x86)\
Enter number
55
55
Press any key to continue . . .

```

السطر 5 عملية حجز لمتحول من نوع عدد صحيح اسمه N

السطر 6 ادخال العدد الصحيح من لوحة المفاتيح وتخزينه في المتحول N

السطر 7 طباعة العدد الصحيح المخزن ضمن المتحول N

### برنامج جمع عددين صحيحين:

الآن لتتذكر خوارزمية جمع عددين وطباعة الناتج

بداية

اقرأ  $x, y$

$Z = x + y$

اطبع  $z$

النهاية

نلاحظ أننا استخدمنا ثلاث متحولات واحد للخروج واثنان للدخول في C++ نحن بحاجة لتحديد نوع القيم التي ستوضع ضمن المتحولات في عملية تسمى حجز المتحولات

في المثال السابق نحن بحاجة بثلاثة متحولات من نوع أعداد صحيحة كما في الخوارزمية فيكون البرنامج

```

Untitled4.cpp
1 #include<iostream.h>
2 void main()
3 {int x,y,z;
4 cout<<"Enter first number \n";
5 cin>>x;
6 cout<<"Enter second number \n";
7 cin>>y;
8 z=x+y;
9 cout<<z;
10 cout<<"\n";
11 }

"C:\Program Files (x86)\
Enter first number
1
Enter second number
4
5
Press any key to continue . . .

```

**ملاحظات:**

يمكن الاستغناء عن السطر رقم 4 و 6 لأنهما لاخبار المستخدم بالمطلوب منه فقط اي مجرد ديكور.

في السطر 3 تم تعريف المتحولات معا بسطر واحد وهذا صحيح وافضل من استخدام `int` ثلاث مرات.

يجب أن ينتهي كل سطر في البرنامج بقاصلة منقوطة.

## العمليات الحسابية والمنطقية

### أولا العمليات الحسابية:

الوصف	الاستخدام	العملية
جمع المتحول x مع y	x+y	+
طرح المتحول y من x	x-y	-
ضرب المتحول x بالمتحول y	x*y	*
قسمة المتحول x على المتحول y	x/y	/
باقي قسمة المتحول x على المتحول y	x%y	%

### التزايد والتناقص:

الوصف	الاستخدام	المعامل
زيادة قيمة المتحول s بمقدار 1, استخدام قيمة s قبل زيادته	s++	++
زيادة قيمة المتحول s بمقدار 1, استخدام قيمة s بعد زيادته	++s	++
انقاص قيمة المتحول s بمقدار 1, استخدام قيمة s قبل انقاصه	s--	--
انقاص قيمة المتحول s بمقدار 1, استخدام قيمة s بعد انقاصه	--s	--

مثال أوجد خرج البرنامج التالي:

البرنامج	التنفيذ
<pre>#include&lt;iostream.h&gt; void main() {     int a=5;      cout&lt;&lt;a&lt;&lt;"\n";     cout&lt;&lt;a++&lt;&lt;"\n";     cout&lt;&lt;a&lt;&lt;"\n";     cout&lt;&lt;--a&lt;&lt;"\n";     cout&lt;&lt;a++&lt;&lt;"\n";     cout&lt;&lt;a&lt;&lt;"\n"; }</pre>	

```

1 #include<iostream.h>
2 void main()
3 {
4     int a=10;
5     int b=5;
6     cout<<(a++)-b<<"\n";
7     cout<<a<<"\n";
8     cout<<(--b)-b<<"\n";
9
10 }

```



```

5
11
0

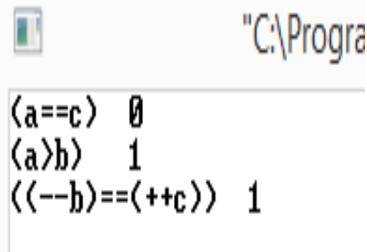
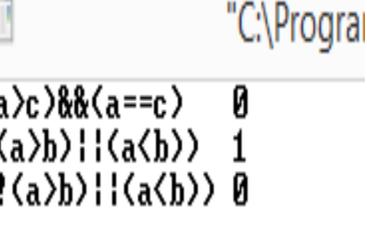
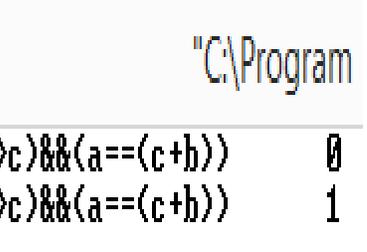
```

### عمليات المقارنة والعمليات المنطقية

الوصف	الاستخدام	العملية
إعادة True في حال كانت x أكبر تماماً من y و False إذا لم يكن x أكبر تماماً من y	x>y	>
إعادة True في حال كانت x أكبر أو تساوي y و False في الحالة غير ذلك.	x>=y	>=
إعادة True في حال كانت x أصغر تماماً من y و False إذا لم يكن x أصغر تماماً من y	x<y	<
إعادة True في حال كانت x أصغر أو تساوي y و False في الحالة غير ذلك.	x<=y	<=
إعادة True في حال كانت x تساوي y و False في حال عدم المساواة .	x==y	==
إعادة True في حال كانت x لا تساوي y و False في حال المساواة .	x!=y	!=

الرمز	العملية
&&	AND
	OR
!	NOT

أمثلة

<pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int a=10; 5     int b=5; 6     int c=3; 7     cout&lt;&lt;"(a==c) \t"&lt;&lt;(a==c)&lt;&lt;"\n"; 8     cout&lt;&lt;"(a&gt;b)\t"&lt;&lt;(a&gt;b)&lt;&lt;"\n"; 9     cout&lt;&lt;"((--b)==(++c))\t"&lt;&lt;((--b)==(++c)) &lt;&lt;"\n"; 10    cout&lt;&lt;"\n"; 11 }</pre>	 <pre> "C:\Progra (a==c) 0 (a&gt;b) 1 ((--b)==(++c)) 1</pre>
<pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int a=10; 5     int b=5; 6     int c=3; 7     cout&lt;&lt;"(a&gt;c) &amp;&amp;(a==c) \t"&lt;&lt;((a&gt;c) &amp;&amp;(a==c))&lt;&lt;"\n"; 8     cout&lt;&lt;"((a&gt;b)    (a&lt;b))\t"&lt;&lt;((a&gt;b)    (a&lt;b))&lt;&lt;"\n"; 9     cout&lt;&lt;"(! (a&gt;b)    (a&lt;b))\t"&lt;&lt;(! (a&gt;b)    (a&lt;b)) &lt;&lt;"\n"; 10    cout&lt;&lt;"\n"; 11 }</pre>	 <pre> "C:\Progra (a&gt;c)&amp;&amp;(a==c) 0 ((a&gt;b)    (a&lt;b)) 1 (!(a&gt;b)    (a&lt;b)) 0</pre>
<pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int a=10; 5     int b=5; 6     int c=4; 7     cout&lt;&lt;"((a&gt;c) &amp;&amp;(a==(c+b)) \t"&lt;&lt;((a&gt;c) &amp;&amp;(a==(c+b)))&lt;&lt;"\n"; 8     c++; 9     cout&lt;&lt;"((a&gt;c) &amp;&amp;(a==(c+b)) \t"&lt;&lt;((a&gt;c) &amp;&amp;(a==(c+b)))&lt;&lt;"\n"; 10    cout&lt;&lt;"\n"; 11 }</pre>	 <pre> "C:\Program ((a&gt;c)&amp;&amp;(a==(c+b))) 0 ((a&gt;c)&amp;&amp;(a==(c+b))) 1</pre>

## If الشرطية

تقابل في الخوارزميات عبارة (إذا كان ..وإلا إذا كان ...وإلا)

### الشكل العام للتعليمة:

If (شرط)

{تعليمات}

Else if (شرط)

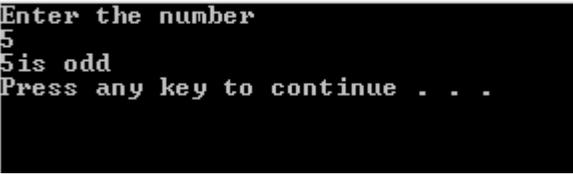
{تعليمات}

Else

{تعليمات}

### البرنامج الأول:

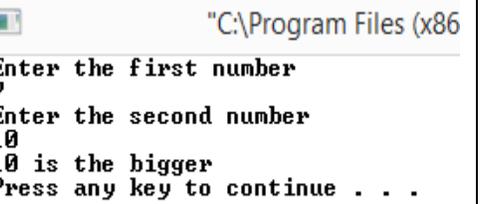
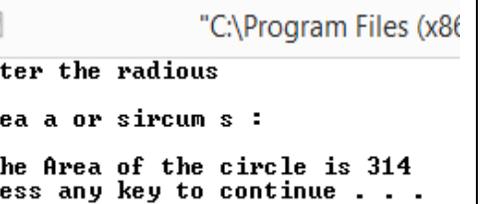
طباعة فيما اذا كان العدد زوجي أم فردي

البرنامج	الخوارزمية
<p>Untitled5.cpp</p> <pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int x; 5     cout&lt;&lt;"Enter the number \n"; 6     cin&gt;&gt;x; 7     if(x%2==0) 8     {cout&lt;&lt; x &lt;&lt;"is even \n";} 9     else 10    {cout&lt;&lt; x &lt;&lt;"is odd \n";} 11 }</pre>	<p>بداية أقرأ x إذا كان <math>x\%2=0</math> اطبع "is even" x وإلا اطبع "is odd" x نهاية</p>
<p>"C:\Program Files (x86)\C</p> 	

ملاحظة

سطر الشرط لا يحتاج فاصلة منقوطة لأن التعليمة لم تنتهي بعد

مجموعة تمارين للتدريب

<pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int x,y; 5     cout&lt;&lt;"Enter the first number \n"; 6     cin&gt;&gt;x; 7     cout&lt;&lt;"Enter the second number \n"; 8     cin&gt;&gt;y; 9     if(x&gt;y) 10    {cout&lt;&lt; x &lt;&lt;" is the bigger \n";} 11    else if(y&gt;x) 12    {cout&lt;&lt; y &lt;&lt;" is the bigger\n";} 13    else 14    {cout&lt;&lt; " are equal \n";} 15 }</pre>	<p>إيجاد العدد الأكبر بين عددين</p> 
<pre> 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4     int r;char choise; 5     cout&lt;&lt;"Enter the raiious \n"; 6     cin&gt;&gt;r; 7     cout&lt;&lt;"Area a or sircum s :\n"; 8     cin&gt;&gt;choise; 9     double Area,sircum; 10    if(choise=='a') 11    {Area=3.14*r*r; 12    cout&lt;&lt;" The Area of the circle is "&lt;&lt; Area&lt;&lt;"\n" ;} 13    else if(choise=='s') 14    {sircum=2*3.14*r; 15    cout&lt;&lt;" The sircum of circle is "&lt;&lt; sircum&lt;&lt;"\n" ;} 16    else 17    {cout&lt;&lt; " wrong choise \n" ;} 18 }</pre>	<p>حساب مساحة أو محيط دائرة حسب طلب المستخدم</p> 

```

#include<iostream.h>
d main()

int M;
M=12;
if (M==12)
{ cout<<" the number is 12"<<"\n" ;}
if (M<5)
{cout<<" The number is smaller then 5 "<<"\n" ;}
if (M>5)
{cout<<" The number is greater then 5 "<<"\n" ;}

```

أوجد خرج البرنامج

```

"C:\Program Files (x86)\C
the number is 12
The number is greater then 5
Press any key to continue

```

### بنية switch :

هي بنية بديلة عن الشرط if تستخدم للتبديل بين عدة حالات حسب قيمة متحول يلزم اضافة تعليمة break في نهاية كل حالة حتى لا يستمر بباقي الحالات.

مثال: برنامج يحسب مساحة أو محيط دائرة حسب خيار المستخدم

```

1 #include<iostream.h>
2 void main()
3 {
4     int R;
5     char ch;
6     cout<<"Enter the raduis \n";
7     cin>>R;
8     cout<<" Enter a for area or s for sircum \n" ;
9     cin>>ch;
10    switch (ch) {
11        case 'a':cout<<" The Area is \t "<<3.14*R*R<<"\n";break;
12        case 's':cout<<" The Sircum is \t "<<2*3.14*R<<"\n";break;
13        default:
14            cout<<" wrong choise "<<"\n" ;}
15 }

```

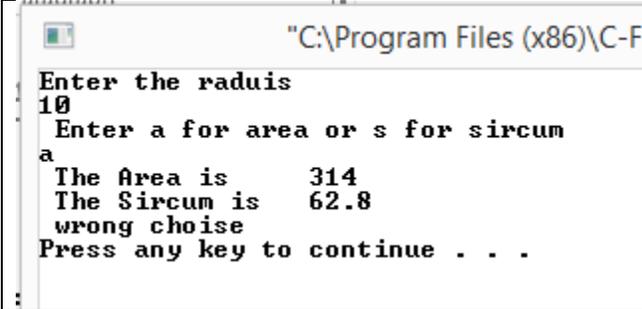
```

"C:\Program Files (x86)\C-Fi
Enter the raduis
10
Enter a for area or s for sircum
s
The Sircum is 62.8
Press any key to continue . . .

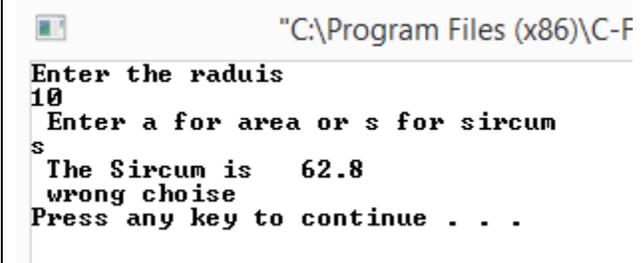
```

في حال حذفنا كلمة `break` سيتم تنفيذ كل الحالات التالية لحالة المحققة كما في المثال التالي

```
main05.cpp
1 #include<iostream.h>
2 void main()
3 {
4     int R;
5     char ch;
6     cout<<"Enter the raduis \n";
7     cin>>R;
8     cout<<" Enter a for area or s for sircum \n" ;
9     cin>>ch;
10    switch(ch){
11        case 'a':cout<<" The Area is \t "<<3.14*R*R<<"\n";
12        case 's':cout<<" The Sircum is \t "<<2*3.14*R<<"\n";|
13        default:
14            cout<<" wrong choise "<<"\n" ;}
15 }
```



```
"C:\Program Files (x86)\C-F
Enter the raduis
10
Enter a for area or s for sircum
a
The Area is      314
The Sircum is    62.8
wrong choise
Press any key to continue . . .
```



```
"C:\Program Files (x86)\C-F
Enter the raduis
10
Enter a for area or s for sircum
s
The Sircum is    62.8
wrong choise
Press any key to continue . . .
```

## التكرار والحلقات

تقابل في الخوارزميات عبارة (كرر طالما (شرط توقف))

### حلقة While:

( شرط تتوقف عنده الحلقة) while

{تعليمات من ضمنها تغير الشرط }

### مثال:

طباعة الأعداد من 1 الى 10

البرنامج	الخوارزمية
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i=1; 5 while (i&lt;=10) 6 {cout&lt;&lt;i; 7 i++;} 8 cout&lt;&lt;"\n"; 9 } </pre>	<p>بداية i=1 كرر طالما i&lt;=10 اطبع i i++ نهاية التكرار نهاية</p>
<pre> "C:\Program Files (x86) 12345678910 Press any key to continue . . . </pre>	

ايجاد القاسم المشترك الأكبر

البرنامج	الخوارزمية
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int x,y; 5 cout&lt;&lt;"Enter first number \n"; 6 cin&gt;&gt;x; 7 cout&lt;&lt;"Enter second number \n"; 8 cin&gt;&gt;y; 9 cout&lt;&lt;"x\t y \n"; 10 cout&lt;&lt;x&lt;&lt;"\t"&lt;&lt;y&lt;&lt;"\n"; 11 while (x!=y) 12 { 13     if (x&gt;y) 14         {x=x-y;} 15     else 16         {y=y-x;} 17     cout&lt;&lt;x&lt;&lt;"\t"&lt;&lt;y&lt;&lt;"\n"; 18 } 19 cout&lt;&lt;"the GCD is"&lt;&lt;x&lt;&lt;"\n"; 20 } </pre>	<p>بداية  أقرأ x,y  كرر طالما x لا يساوي y  إذا كان x&gt;y  x=x-y  والا  y=y-x  نهاية التكرار  اطبع x  نهاية</p>
<pre> "C:\Program Files Enter first number 30 Enter second number 24 x      y 30     24 6      24 6      18 6      12 6      6 the GCD is6 Press any key to continue . . . </pre>	

ملاحظة

كل من الأسطر 9 10 17 لتوضيح خطوات التنفيذ لحساب القاسم المشترك ويمكن الاستغناء عنها ليعطي البرنامج النتيجة مباشرة.

**حلقة do While**

تعليمات do{

} while ( شرط تتوقف عنده الحلقة )

**مثال:**

طباعة الأعداد من 1 الى 10

البرنامج	الخوارزمية
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i=1; 5 6 7 do{ 8     cout&lt;&lt;i; 9     i++; 10 }while(i&lt;=10); 11 cout&lt;&lt;"\n"; 12 } </pre>	<p>بداية i=1 كرر اطبع i i++ طالما i&lt;=10 نهاية</p>
<pre> "C:\Program Files (x86 12345678910 Press any key to continue . . . </pre>	

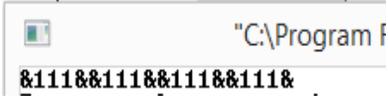
ملاحظة:

تختلف حلقة while عن حلقة do while بأن حلقة do while تنفذ تكرار على الأقل عند عدم تحقق الشرط في البداية

**مثال**

<pre>Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i=1; 5 6 7 do{ 8     cout&lt;&lt;i; 9     i++; 10 }while(i&gt;5); 11 cout&lt;&lt;"\n"; 12 }</pre>	<pre>Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i=1; 5 6 while(i&gt;5) 7 { 8     cout&lt;&lt;i; 9     i++; 10 } 11 cout&lt;&lt;"\n"; 12 }</pre>
<pre>"C:\Program Files (x86) 1 Press any key to continue . . .</pre>	<pre>"C:\Program Files (x86) Press any key to continue . . .</pre>

مجموعة تمارين للتدريب اوجد خرج البرامج التالية :

<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i=1; 5 cout&lt;&lt;"*****\n"; 6 while (i&lt;=4) 7 { 8     cout&lt;&lt;"*\t*\n"; 9     i++; 10 } 11 cout&lt;&lt;"*****"; 12 cout&lt;&lt;"\n"; 13 } </pre>	
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int i,j=1; 5 while (j&lt;=4) 6 {i=1; 7     cout&lt;&lt;"&amp;"; 8     while (i&lt;=3) 9     { 10         cout&lt;&lt;"1"; 11         i++; 12     } 13     cout&lt;&lt;"&amp;"; 14     j++; 15 } 16 cout&lt;&lt;"\n"; 17 } </pre>	
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4 int j=1; 5 while (j&lt;=4) 6 { 7     cout&lt;&lt;"&amp;"; 8       9 } 10 cout&lt;&lt;"\n"; 11 } </pre>	<p>حلقة لا نهائية لعدم وجود ++j</p>

اكتب برنامج يطبع الاعداد الزوجية بين 1 و 20 بينهم فاصلة

```

Untitled5.cpp
1 #include<iostream.h>
2 void main()
3 {
4 int j=1;
5 while (j<=20)
6 {
7     if (j%2==0)
8         {cout<<j<<" , ";}
9     j++;
10
11
12 }
13 cout<<"\n";
14 }

```

"C:\Program Files (x86)\C  
2,4,6,8,10,12,14,16,18,20.  
Press any key to continue . . .

اكتب برنامج يدخل من المستخدم 5 اعداد صحيحة ويجمع الزوجية فقط ويطبع المجموع

```

Untitled5.cpp
1 #include<iostream.h>
2 void main()
3 {
4 int j=0;
5 int x;
6 int sum=0;
7 while (j<5)
8 { cin>>x;
9     if (x%2==0)
10        {sum+=x;}
11    j++;
12 }
13 cout<<"SUM ="<<sum<<"\n";
14 }

```

"C:\Program I  
2  
3  
8  
9  
6  
SUM =16  
Press any key to continue .

## التكرار والحلقات

تقابل في الخوارزميات عبارة (كرر طالما (شرط توقف))

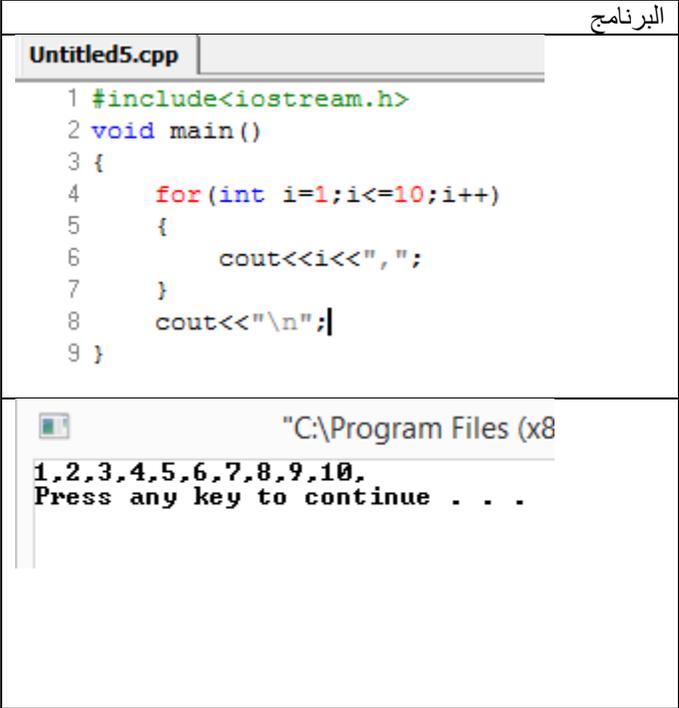
### حلقة For:

( الزيادة أو النقصان; شرط التوقف; تهيئة العداد) For

{ تعليمات }

### مثال:

طباعة الأعداد من 1 الى 10



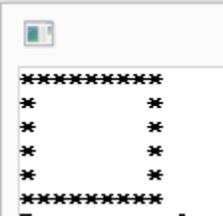
The screenshot shows a C++ program in a text editor and its execution output. The program is named 'Untitled5.cpp' and contains the following code:

```
1 #include<iostream.h>
2 void main()
3 {
4     for(int i=1;i<=10;i++)
5     {
6         cout<<i<<" ";
7     }
8     cout<<"\n";
9 }
```

The output window shows the result of the program execution:

```
"C:\Program Files (x86)
1,2,3,4,5,6,7,8,9,10.
Press any key to continue . . .
```

**مجموعة تمارين للتدريب** اوجد خرج البرامج التالية :

<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { cout&lt;&lt;"*****\n"; 4   for(int i=1;i&lt;=3;i++) 5     { 6       cout&lt;&lt;"*\t* \n"; 7     } 8   cout&lt;&lt;"*****\n"; 9 } </pre>	 <pre> ***** *       * *       * *       * *       * ***** </pre>
<pre> Untitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4   for(int j=1;j&lt;=4;j++) 5     { 6       cout&lt;&lt;"&amp;"; 7       for(int i=1;i&lt;=3;i++) 8         {cout&lt;&lt;"1";} 9       cout&lt;&lt;"&amp;"; 10    } 11   cout&lt;&lt;"\n"; 12 13 } </pre>	 <pre> "C:\Program f &amp;111&amp;&amp;111&amp;&amp;111&amp;&amp;111&amp; </pre>
<pre> ntitled5.cpp 1 #include&lt;iostream.h&gt; 2 void main() 3 { 4   for(int j=1;j&lt;=18;j++) 5     { 6       if(18%j==0) 7         {cout&lt;&lt;j&lt;&lt;" ";} 8     } 9   cout&lt;&lt;"\n"; 10 11 } </pre>	<p>لاحظ أن الخرج هو قواسم ال18</p>  <pre> "C:\Program Files (x86)\ 1,2,3,6,9,18, Press any key to continue . . . </pre>

أكتب برنامج يطبع مجموع الاعداد الفردية بين 1 و 10

```

Jntitled5.cpp
1 #include<iostream.h>
2 void main()
3 { int sum=0;
4   for(int i=1;i<=10;i++)
5   {
6     if(i%2!=0)
7       {sum+=i;}
8   }
9   cout<<sum<<"\n";
10
11 }

```

"C:\Program Files (x86

25  
Press any key to continue . . .

اكتب برنامج يدخل من المستخدم 5 أعداد صحيحة يدخلها المستخدم ويجمع مضاعفات ال3 فقط ويطبع المجموع

```

Untitled5.cpp
1 #include<iostream.h>
2 void main()
3 { int sum=0;int x;
4   for(int i=1;i<=5;i++)
5   { cin>>x;
6     if(x%3==0)
7       {sum+=x;}
8   }
9   cout<<sum<<"\n";
10
11 }

```

"C:\Program Files (x8

5  
9  
3  
7  
6  
18  
Press any key to continue . . .

لاحظ أن مهمة الحلقة في التمرين الاخير عد الاعداد المدخلة من المستخدم فقط .