

Structural Programming with C++

المفاهيم والعناصر الأساسية للبرمجة المهيكلة

1- Identifiers (المسميات)

وهي الاسماء التي يطلقها المبرمج على عناصر ومكونات البرنامج.
أي: كل شيء يعرف ضمن البرنامج يجب ان يعطى أسم.. وهذه المسميات يكون المبرمج (أنت) حر بتسمية عناصر برنامجك.

المسميات: هي مجموعة من الحروف والارقام بحيث:

- 1- يجب ان يبدأ المسمى بحرف.
- 2- لا يتخلله أي من العلامات الرياضية (+، -، *،) وعلامات التنقيط (!، ؟، :، ؛، ...)
- 3- ممكن استخدام علامة واحدة فقط ضمن المسمى وهي الـ underscore (_).

2- Keywords (الكلمات الدالة)

وهي الاسماء التي تكون محجوزة ضمن اللغة ولا يجوز نهائيا تسمية مسميات عناصر البرنامج بهذه الاسماء. بمعنى: هي الاسماء التي يستخدمها المعالج الخاص بلغة C++ لمعالجة برنامجك.

مثلاً:

وغيرها..... include while for switch false TRUE main cin

أمثلة صحيحة:

a us b12 B12 hello
I_am_a_student fly727 no
yes_16 std1 std20 ship5h1

أمثلة خاطئة:

586_cpu object-programming
US\$ goto true FALSE
8M4c 2020-2021

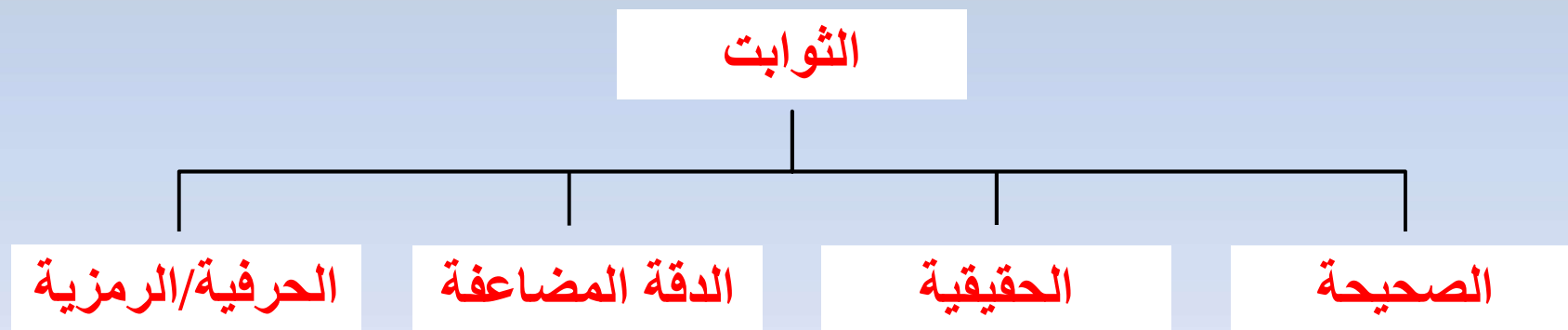
3- Constants (الثوابت)

وهي القيم العددية (أو الحرفية) التي لا تتغير اثناء تنفيذ البرنامج.
مثلا في المعادلة التالية:

$$y = 2 * x + 5 ;$$

فإن القيمتين في المعادلة (2 والـ 5) لا يتغيرون مهما تم تنفيذ هذه المعادلة.

وللتوابت انواع، منها ما يلي:



الثوابت الصحيحة

وهي الثوابت العددية (الموجبة والسالبة) التي لا تتخللها الفاصلة العشرية. أي لا يوجد فيها (.)

مثلاً:

15 -101 8 2359 -223344

الثوابت الحقيقية

وهي الثوابت العددية (الموجبة والسالبة) التي تتخللها الفاصلة العشرية. أي يوجد فيها (.)

مثلاً:

6.3 -0.0052 111.223344 -99.9

ثوابت الدقة المضاعفة

وهي الثوابت العددية **الحقيقية** (الموجبة والسالبة) التي تكون إما **كبيرة جداً** أو **صغيرة جداً** بحيث أن الحاسبة لا تستطيع تخزينها بشكلها الكبير جداً أو الصغير جداً إلا بعد إجراء تحويل عليها لتصبح ذات دقة مضاعفة. وحسب المثال التالي:

الثابت كبير جداً:

مثلاً العدد الصحيح (987654321) كبير جداً،
وتعتبر الفارزة العشرية له في أقصى اليمين
أي أن العدد في الأساس هو (987654321.0).

ولتصغير هذا العدد يجب تزحيف الفارزة العشرية إلى اليسار لعدة مرات. مثلاً 7 مرات،

ولبقاء العدد يحمل نفس القيمة الأساسية بعد التزحيف فيجب ضربه بـ 10^{+n} .

أي يصبح 98.7654321×10^7

ويكتب هذا الثابت العددي بلغة C++ :

98.7654321E+07

حيث أن E تقرأ (عشرة للأس).

أمثلة:

2233114455998877 → 2233.114455998877 E +12

5000000 → 5.0 E+06

1800123 → 0.1800123 E +07

الثابت صغير جداً:

مثلاً العدد الحقيقي (0.0000000015) صغير جداً،
ولصغر هذا العدد مثلاً والقريب جداً من الصفر، فلا يمكن
تمثيله داخل الحاسبة الا بعد تحويله الى عدد ذو دقة مضاعفة.

ولتكبير هذا العدد يجب تزحيف الفارزة العشرية الى اليمين لعدة مرات. مثلاً 8 مرات،

ولبقاء العدد يحمل نفس القيمة الاساسية بعد التزحيف فيجب ضربه بـ 10^{-n} .

أي يصبح 0.15×10^{-8}

ويكتب هذا الثابت العددي بلغة ++C :

0.15E - 08

حيث ان E تقرأ (عشرة للأس).

أمثلة:

0.0000532 → 0.532 E -04

0.0000532 → 53.2 E -06

0.0000000089 → 89.0 E -10

الثوابت الحرفية

وهي الثوابت الرمزية (الحرفية) غالباً المعالج للغة لا يتدخل بمعالجتها. وهي عادة ما توضع بين علامات اقتباس

أمثلة:

“a” “z” “5” “!” “?” “6+4” “a-b+c” “18!”

4- Variables (المتغيرات)

وهي القيم التي **تتغير** اثناء تنفيذ البرنامج. مثلاً في المعادلة التالية:

$$y = 2 * x + 5 ;$$

فإن قيمة y في المعادلة تتغير بتغير قيمة x كلما تم تنفيذ هذه المعادلة.

والمتغيرات هي التي تعتبر عناصر البرنامج التي يستخدمها المبرمج لحل المشكلة قيد الحل. فيجب تسمية هذه المتغيرات واعطائها اسماء (قدر الامكان) تدل على الغرض من عمل هذه المتغيرات.

مثلاً:

s a b c x z s0 s1 s2

student_no sum factorial fact average av result

5- Data Types (انواع البيانات)

هناك العديد من البيانات المتعددة ومنها المشتقة والمتفرعة من البيانات الاساسية. سنذكر انواع البيانات الرئيسية والتي تشبه انواع الثوابت والمتغيرات الاربعة:

int

وهو الذي يمثل البيانات الصحيحة، وتعرف المتغيرات الصحيحة ضمن البرنامج بالشكل التالي:

```
int a ;  
int std24 , b , z ;
```

float

وهو الذي يمثل البيانات الحقيقية، وهي البيانات التي قيمها تكون كسرية وليست صحيحة، أي تحتوي على فاصلة عشرية، وتسمى ايضاً ببيانات (النقطة العائمة) حيث ان الفاصلة العشرية تعوم الجزء الصحيح عن الجزء الكسري للقيمة العددية. وتعرف المتغيرات الحقيقية ضمن البرنامج بالشكل التالي:

```
float sum ;  
float y , average , total ;
```

double

وهو الذي يمثل البيانات الحقيقية التي تكون قيمها كبيرة جداً أو صغيرة جداً، أي البيانات ذات الدقة المضاعفة وهي البيانات التي قيمها تكون أيضاً كسرية وليست صحيحة. ولكنها تحتوي على الـ $(10^{\pm n})$.

وتعرف المتغيرات ذات الدقة المضاعفة ضمن البرنامج بالشكل التالي:

```
double av1 , av2 ;
```

```
double x , y , z ;
```

char

وهو الذي يمثل البيانات الحرفية والتي تكون قيمها عبارة عن رموز وأحرف. وتعرف المتغيرات الحرفية ضمن البرنامج بالشكل التالي:

```
char symbol1 , symbol2 ;
```

```
char ss ;
```

```
char a , h ;
```


Data Type



6- Initializing Variables (اعطاء قيم ابتدائية للمتغيرات عند تعريفها)

لو اردنا تعريف المتغير num على اعتباره متغير صحيح، فيجب تعريفه بالشكل:

```
int num ;
```

وعلى افتراض ان num سوف يأخذ قيمة ابتدائية في بداية البرنامج تساوي 5، فيجب

```
num = 5 ;
```

وبدل كتابة جملتين منفصلتين، الاولى تعريف num، والثانية تخصيص قيمة ابتدائية له، فمن الممكن تعريف المتغير وتخصيص القيمة الابتدائية له مباشرة اثناء التعريف:

```
int num = 5 ;
```

امثلة:

```
int a = 16 , b = -1 ;
```

```
double x = 15.2E8
```

```
float x = 0.02 ;
```

```
double f , z = 1E-8 ;
```

```
float m , r = 0.0 ;
```

```
char a , b = "m" ;
```

7- Expressions (التعابير)

وهي مجموعة من الثوابت والمتغيرات تفصلها علامات رياضية او منطقية او كلاهما.

$$x + 5 - b \quad (\text{تعبير حسابي})$$

$$a > c \quad (\text{تعبير منطقي})$$

$$f + n - 2 < 4 * 3 + s \quad (\text{تعبير منطقي})$$

8- Statements (العبارة/الجملة)

وهي كل تعبير مسبق بعلامة (=)، أي معادلة:

$$3 * x - m + 9 \quad (\text{تعبير حسابي})$$

$$y = 3 * x - m + 9 ; \quad (\text{عبارة حسابية})$$

وتشمل ايضاً الجمل البرمجية التي سوف تكتب في البرنامج، مثلاً

جملة تعريف المتغير (int a ;) هذه **جملة** تعريف المتغير a .

جملة القراءة

جملة الكتابة

جملة المقارنة

جملة التكرار ... وغيرها والتي سيتم ذكرها في وقتها

9- Mathematical Operators (العمليات الرياضية)

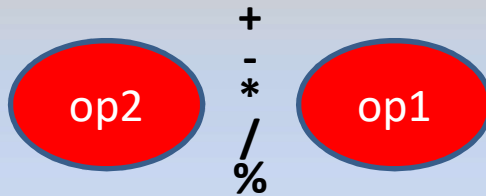
العملية	المعنى	امثلة
+	الجمع	$12 + 6 \rightarrow 18$
-	الطرح	$12 - 6 \rightarrow 6$
*	الضرب	$12 * 6 \rightarrow 72$
/	القسمة	$12 / 6 \rightarrow 2$
%	باقي القسمة	$12 \% 6 \rightarrow 0, 5 \% 3 \rightarrow 2$

هناك (خمسة)
عمليات رياضية

ملاحظـة:
عملية باقي القسمة
(%) **تعمل فقط على**
متغيرات **صحيحة**

10- Unary Operators (العمليات الاحادية)

نلاحظ ان جميع العمليات الرياضية تحتاج **قيمتين عدديـة** (operand) لاجراء العملية.
واحدة على يمين العملية والاخرى على يسارها.



العملية	المعنى	امثلة
++	الزيادة بـ 1	$5 ++ \rightarrow 6$
--	النقصان بـ 1	$5 -- \rightarrow 4$

بينما في العمليات الاحادية نحتاج الى
قيمة واحدة فقط.. ولهذا تسمى
بالعمليات الاحادية.

11- Relational Operators (العمليات العلائقية)

العملية	المعنى	امثلة
>	اكبر من	if(a > 17)
<	اصغر من	if(a < z + 3)
>=	اكبر أو يساوي	if(av >= 50)
<=	اصغر أو يساوي	if(sum <= total)
==	يساوي	if(a + b == c)
!=	لا يساوي	if(3 + 5 != 9)

هناك (ستة) عمليات
علائقية.. وعادة
تستخدم هذه العمليات
للمقارنة بين حدين.

12- Logical Operators (العمليات المنطقية)

العملية	المعنى	امثلة
&&	and (و)	if(a > 3 && b < 12)
	or (أو)	if(av >= 50 av < 60)

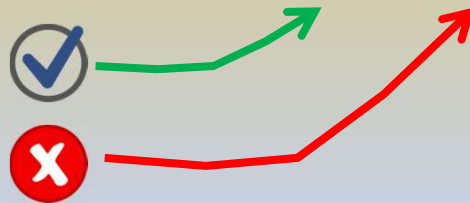
هناك (عمليتين) منطقيه..
وعادة تستخدم هذه العمليات
للمقارنة بين حدين.

13- Conditional Operator (العملية الشرطية)

وهي العملية التي تنفذ جملة حسب الشرط المعطى لها..... ولها الشكل التالي:



var = (condition) ? St1 : St2 ;



m = (15 >= 6 + 9) ? 1 : -1 ;

→ m = 1

مثال:

x = (av >= 50) ? "p" : "f" ;

مثال:

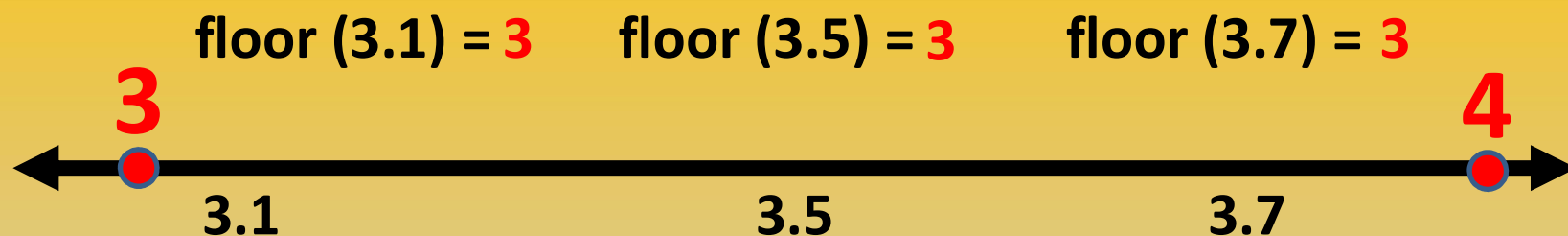
14- Precedence of Operators (الأسبقية في العمليات)

التسلسل	العمليات	الملاحظات
1	()	كل شيء بين اقواس
2	++ , --	العمليات الاحادية (حسب الاسبقية من جهة اليسار)
3	* , / , %	الضرب/القسمة/باقي القسمة (حسب الاسبقية من اليسار)
4	+ , -	الجمع/الطرح (حسب الاسبقية من جهة اليسار)
5	< , <=	أصغر من/أصغر أو يساوي (بالتوالي)
6	> , >=	أكبر من/أكبر أو يساوي (بالتوالي)
7	== , !=	يساوي/لا يساوي (حسب الاسبقية من جهة اليسار)
8	&&	(و)
9		(أو)
10	a ? b : c	العملية الشرطية
11	=	المساواة في العبارات الحسابية أو (الجملة)

15- Library Functions (الدوال المكتبية)

رياضياً	بلغة C++	نوع ادخال الدالة	نوع اخراج الدالة
\sqrt{x}	sqrt(x)	float	float
$ x $	abs(x)	int	int
$ x $	fabs(x)	float	float
e^x	exp(x)	float	float
sin x	sin(x)	float	float
cos x	cos(x)	float	float
tan x	tan(x)	float	float
x^y	pow(x,y)	float	float
	floor(x)	float	int
	ceil(x)	float	int
	random(n)	int	int (0→n-1)

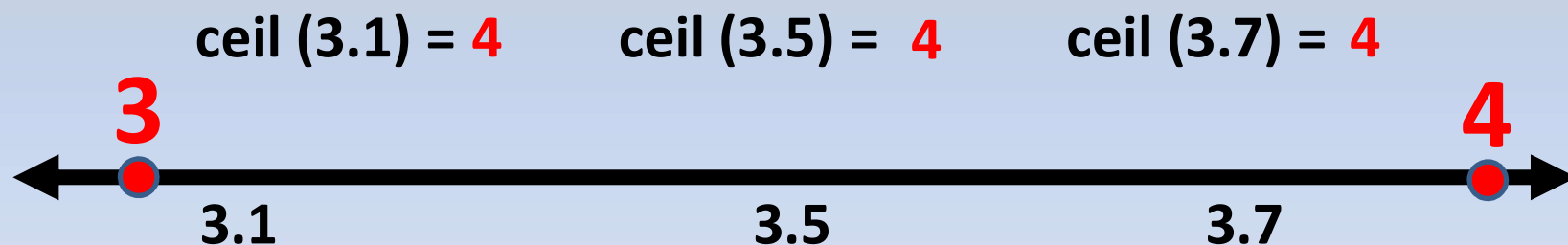
floor: تعني الأرضية أي بمعنى القيمة الأدنى... وهي الدالة التي تستقبل عدد حقيقي وتعيد أول عدد صحيح أصغر من العدد الحقيقي المدخل.



أمثلة: جد ناتج ما يلي

$\text{floor}(8.7)$, $\text{floor}(-5.1)$, $\text{floor}(-2.5)$

ceil: تعني السقف أي بمعنى القيمة الأعلى... وهي الدالة التي تستقبل عدد حقيقي وتعيد أول عدد صحيح أكبر من العدد الحقيقي المدخل.




أمثلة: جد ناتج ما يلي

$\text{ceil}(8.7)$, $\text{ceil}(-5.1)$, $\text{ceil}(-2.5)$

ملاحظة مهمة جداً

$+$ $-$ $*$ $/$	int	float
int	int	float
float	float	float



$$7 / 2 \rightarrow 3$$

$$7.0 / 2 \rightarrow 3.5$$

$$7 / 2.0 \rightarrow 3.5$$

$$7.0 / 2.0 \rightarrow 3.5$$

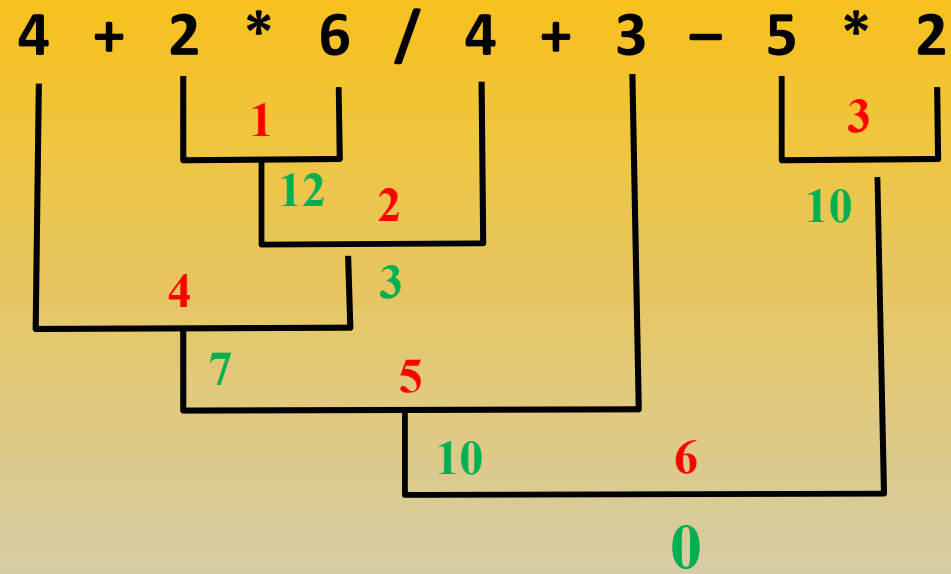
لان صحيح / صحيح = صحيح

لان حقيقي / صحيح = حقيقي

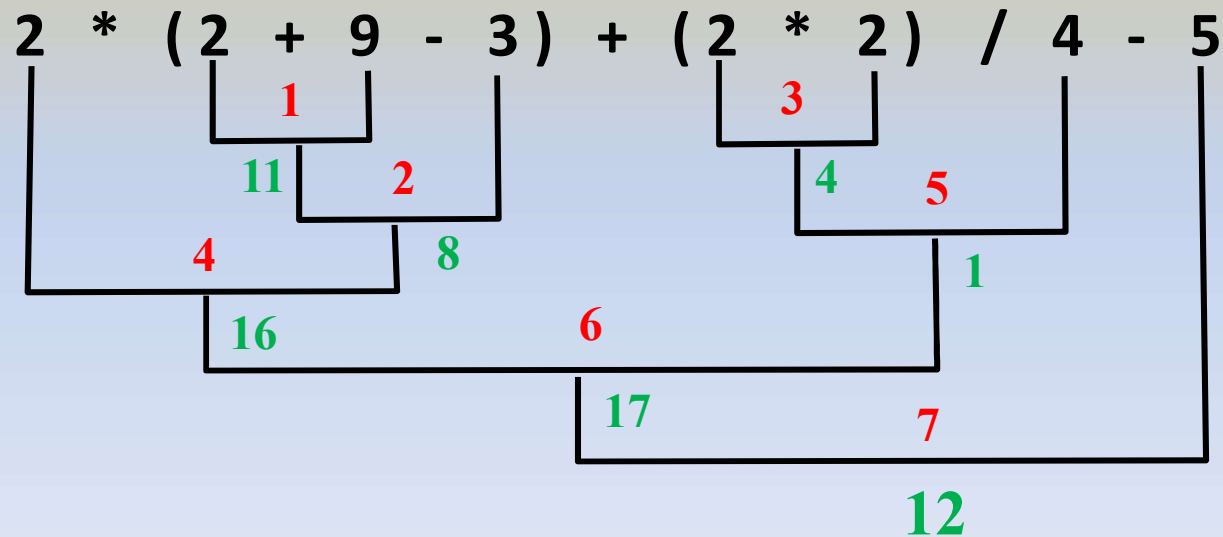
لان صحيح / حقيقي = حقيقي

لان حقيقي / حقيقي = حقيقي

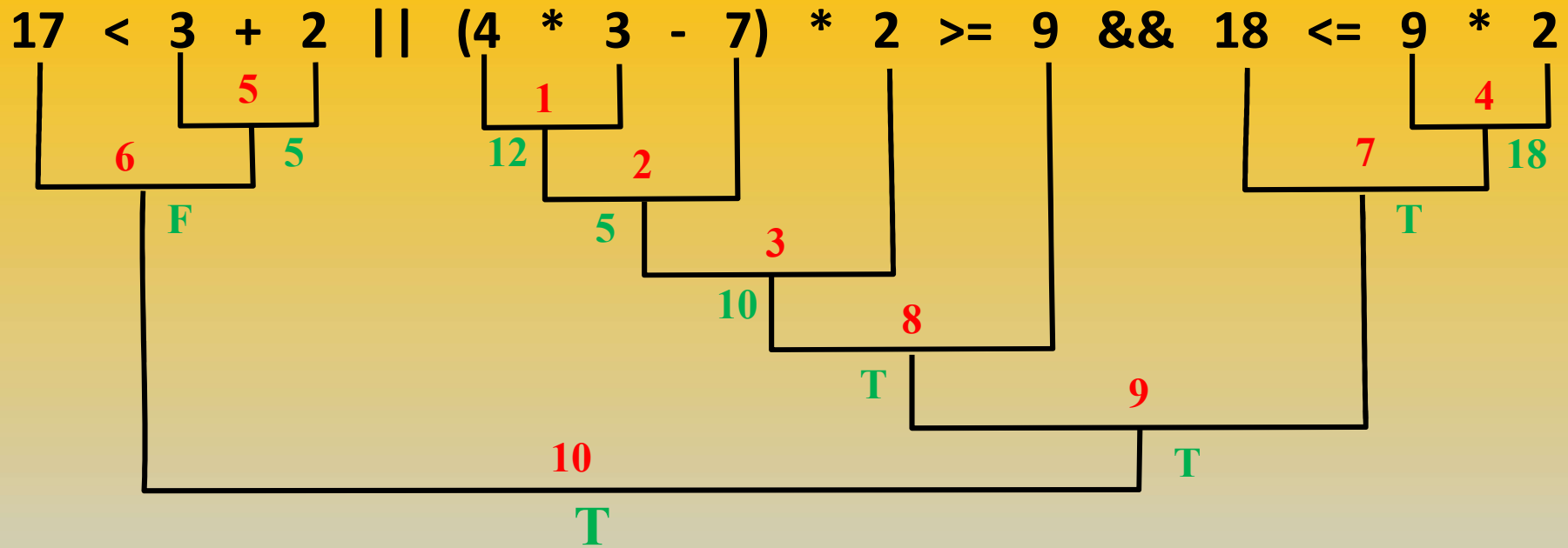
مثال: جد ناتج التعبير الحسابي التالي:



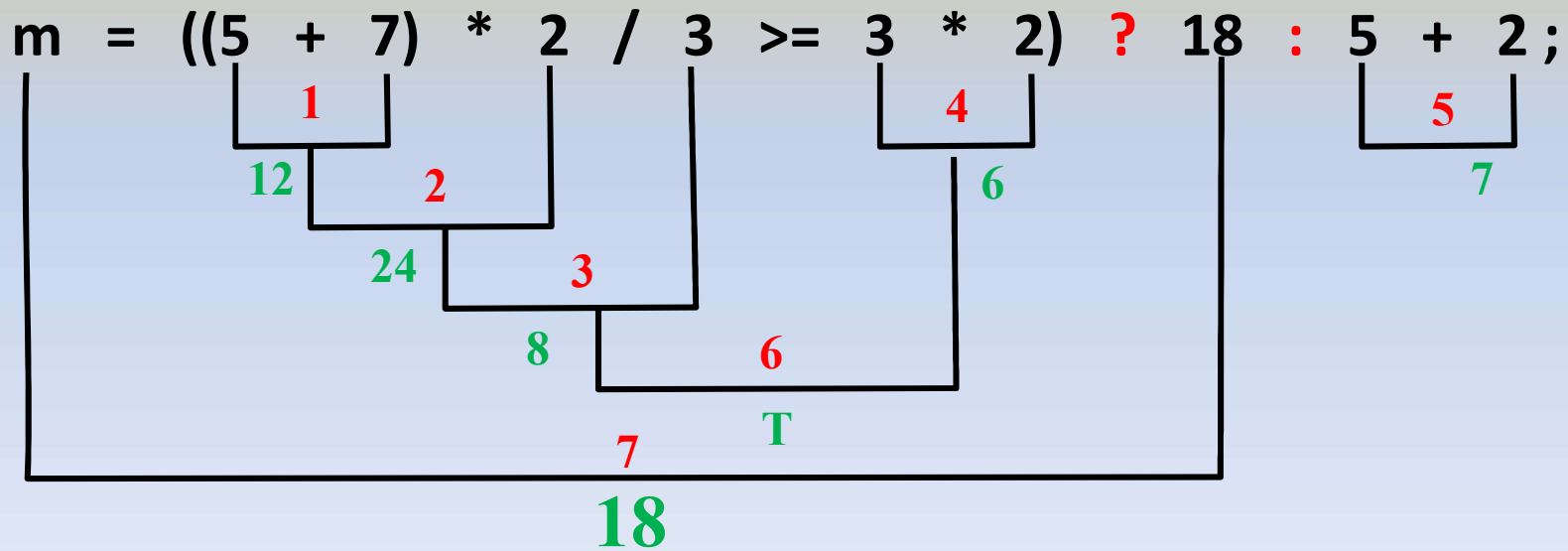
مثال: جد ناتج التعبير الحسابي التالي:



مثال: جد ناتج التعبير التالي:



مثال: جد ناتج العبارة التالية:



10- Converting Math. Expressions into Programmable Expressions

تحويل التعابير الرياضية الى تعابير بلغة البرمجة:

$$\frac{a}{b} \rightarrow a / b$$

$$\frac{a}{b} + c \rightarrow a / b + c$$

$$a + \frac{b}{c} \rightarrow a + b / c$$

$$\frac{a}{b} + \frac{c}{d} \rightarrow a / b + c / d$$

اذن: عند تحويلك للتعابير الرياضية، ابدأ بالتحويل كما تقرأها تماماً.

$$\frac{a + b}{c} \rightarrow (a + b) / c$$

ملاحظة:

كل بسط أو مقام

$$\frac{a + b}{2 * c} \rightarrow (a + b) / (2 * c)$$

(أكثر من حد جبري واحد)

ضعه بين أقواس

$$\frac{a + b}{\sin x} \cdot \frac{\cos y}{f+1} \rightarrow (a + b) / \sin(x) * \cos(y) / (f + 1)$$

$$\sqrt{\frac{5}{\left| \frac{x}{m+1} \right|}} \cdot \frac{e^{r+1}}{a^8} \rightarrow \text{sqrt} (5 / \text{abs} (x / (m+1))) * \text{exp} (r + 1) / \text{pow} (a, 8)$$

تمارين: حول التعبيرات الرياضية التالية الى تعابير بلغة C++

$$\frac{\sin \frac{f}{\left| \frac{a+1}{r+1} \right|} - \sqrt{\frac{a^5}{m}}}{\left| \frac{r-1}{r-1} \right|} \cdot \sqrt{\frac{1-p}{1-p}} \cdot \frac{1}{e^{x-1} + e^{x+1}}$$

$$\frac{\cos \frac{\sin \frac{a+1}{b+1}}{\sin \frac{x}{2}}}{r} \cdot \tan \frac{b-1}{z}$$

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\sqrt{a^2 + b^2} + \frac{\sqrt{a^2 + b^2}}{\cos \frac{a}{b+c}}$$

11- The structure of a C++ program

الهيكل العام لبرنامج بلغة C++:

الهيكل العام (للمرحلة الحالية) يتكون من ثلاثة اجزاء:

الـ header:

بشكل عام، يشتمل البرنامج على عناصر برمجة متنوعة مثل الدوال الجاهزة والمسميات المحجوزة وغيرها، والتي تكون في مكتبات (مجلدات) للغة C++ القياسية. ومن أجل استخدام هذه العناصر في البرنامج، يجب تضمين عنوان مناسب لها في البرنامج لاستدعائها. ويكون هذا الاستدعاء أو التضمين في بداية البرنامج. (أي في رأس البرنامج) ولهذا سميت بالـ header.

header

declarations

```
main ( ) {  
    -----  
    -----  
    -----  
}
```

ولكن الـ header المهم والضروري والذي يجب ان يكون متضمناً في كل برنامج سوف تكتبه هو:

<iostream.h>

مثل هذه المجلدات (الهيدرات/header):

<iostream.h>

<math.h>

<stdlib.h>

<stdio.h>

Declarations: الجزء الثاني هو جزء التعريفات. وهو الجزء الذي يقوم المبرمج (أنت) بتعريف جميع المتغيرات التي سوف تستخدمها في البرنامج. مثلاً جملة (int a ;) أو (float m,z ;) وغيرها.. فهذه هي جمل لتعريف هذه المتغيرات.

header

declarations

```
void main ( ) {  
    -----  
    -----  
    -----  
}
```

Main: الجزء الثالث هو تعريف الدالة الرئيسية للبرنامج مع جسم هذه الدالة من الايعازات والاورامر. فأن معالج البرنامج ال compiler سوف يقوم بالبحث عن هذه الدالة اولاً عند التنفيذ ثم يقوم بمراجعة كل الجمل البرمجية وجميع المتغيرات المستخدمة في البرنامج، هل تم تعريفها أم لا في جزء التعريفات....

ملاحظة:

بما أن **main** هي الدالة الرئيسية في البرنامج.. فهي دالة.. وكل دالة في علم الرياضيات يجب ان يكون لها قيمة أرجاع أو اخراج..

عليه.. فأن كلمة **void** تعني (null) أي (لا شيء).. أي هو ايعاز للمعالج بعدم ارجاع أي قيمة من الدالة الرئيسية main وإنما فقط تنفيذها.. لأن المبرمج (أنت) سوف يقوم بتوضيح وطباعة نتائج البرنامج داخل ومن ضمن هذه الدالة فلا حاجة لارجاع قيمة او قيم الاخراج من خلال هذه الدالة... فعليه يكتب قبل أسم الدالة الرئيسية main كلمة void.

على المبرمج (أنت) ان يقوم بتزويد البرنامج بالبيانات التي يستند عليها حل المشكلة التي كُتِبَ البرنامج لحلها. وهو عملية تزويد الحاسوب بالبيانات الاولية (المعطيات). وتتم هذه العملية عن طريق كتابة جمل الإدخال.. مثل جملة القراءة (**cin**).
 أما استخراج النتائج من الحاسبة وعرضها على شاشة الحاسوب تتم عن طريق جمل الأخراج والتي منها جملة الطباعة على الشاشة (**cout**).

جملة القراءة **cin**: جملة تزويد البرنامج بالبيانات

صيغتها العامة:

cin >> var₁ >> var₂ >> >> var_n ;

من الممكن كتابة عدة جمل قراءة من عدة جمل **cin**:

cin >> var₁ ;

cin >> var₂ ;

⋮

cin >> var_n ;

جملة الطباعة cout: جملة استخراج وطباعة النتائج النهائية على الشاشة

صيغتها العامة: `cout << var1 << var2 << << varn ;`

كمثال: على افتراض ان نتائج برنامجك بعد التنفيذ هي:

`a = 18 ; b = - 8 ; c = 25 ;`

شاشة التنفيذ ستكون

فان طباعة قيم هذه المتغيرات ستكون:

`cout << a << b << c ;`

18-825

لم يتم فصل القيم بفراغات لعدم اعطاء الامر بطباعة فراغ بعد كل قيمة:

`cout << a << " " << b << " " << c ;`

18 -8 25

ولو اردنا طباعة كل قيمة على سطر، فأغلب الظن هو طباعة كل متغير ب cout منفصلة:

`cout << a ;`

`cout << b ;` لم يتم طباعة كل قيمة على سطر لعدم

اعطاء جملة (نهاية سطر) يعني اترك

السطر: وهو ايعاز أو جملة

endl

18-825

`cout << a << endl << b << endl << c ;`

18

-8

25

مدخل البرمجة بلغة C++

كل مسألة رياضية تحتاج الى تحديد المعطيات (المدخلات)، والتي تعتبر البيانات الاساسية التي يستند عليها حل المسألة... ثم البدء بحل المسألة اعتماداً على هذه المدخلات.. واخيراً استخراج نتائج حل المسألة.

بنفس الفكرة.. فكل مسألة برمجية يجب ان تمر ضمن هذه المراحل الثلاثة وهي:

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

Write a program that reads the length of a square. Then, find its circumference and area and print them?
كتابة برنامج لقراءة طول ضلع مربع ثم جد محيطه ومساحته واطبع الناتج

```
# include <iostream.h>
int a ;
int x , y ;
void main () {
    cin >> a ;
    x = a * 4 ;
    y = a * a ;
    cout << x << " " << y << endl ;
}
```

```
# include <iostream.h>
int a ;
void main () {
    cin >> a ;
    cout << a * 4 << endl ;
    cout << a * a << endl ;
}
```

Write a program to read THREE numbers. Then, find:

- The summation of them?
- The square of their summation?
- The summation of their squares?

كتابة برنامج لقراءة 3 اعداد. ثم جد:

- مجموع قيمهم؟
- مربع المجموع؟
- مجموع مربعاتهم؟

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

```
int a , b , c ;
```

```
int sum1 , sq , sum2 ;
```

```
void main ( ) {
```

```
cin >> a >> b >> c ;
```

```
sum1 = a + b + c ;
```

```
sq = sum1 * sum1 ;
```

```
sum2 = a * a + b * b + c * c ;
```

```
cout << sum1 << endl ;
```

```
cout << sq << endl ;
```

```
cout << sum2 << endl ;
```

```
}
```

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

أو $sq = pow (sum1 , 2) ;$

أو $sum2 = pow (a , 2) + pow (b , 2) + pow (c , 2) ;$

Write a program to read an integer number consists of 2 digits. Then, separate and print each digit on a single line?

كتابة برنامج لقراءة عدد صحيح يتكون من مرتبتين (آحاد وعشرات). ثم أعزل واطبع كل مرتبة على سطر منفرد؟

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

```
int num ;
```

```
int d1 , d2 ;
```

```
void main () {
```

```
cin >> num ;
```

```
d2 = num / 10 ;
```

```
d1 = num % 10 ;
```

```
cout << d1 << endl << d2 << endl ;
```

```
}
```

مثلاً: العدد 27

لو قسمنا العدد على 10

$$27 / 10 = 2$$

لان: $int/int=int$

لو اخذنا باقي قسمة العدد على 10

$$27 \% 10 = 7$$

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

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

```
int c ;
```

```
float f ;
```

```
void main () {
```

```
cin >> c ;
```

```
f = 9 / 5.0 * c + 32 ;
```

```
cout << f ;
```

```
}
```

Write a program to read a centigrade temperature. Then convert it to a Fahrenheit ?

كتابة برنامج لقراءة درجة حرارة مئوية. ثم حولها الى درجة حرارة فهرنهايتية ؟ استخدم القانون:

$$F = \frac{9}{5} * C + 32$$

تمارين

Q1) Write a program to read a circle radius. What is the area of the smallest square can be drawn outside this circle?

اكتب برنامج لقراءة نصف قطر دائرة. ما هي المساحة لأصغر مربع ممكن رسمه خارج حدود الدائرة؟
ملاحظة: أرسم دائرة منتظمة ثم ارسم مربع يحيطها ويلامس الدائرة باضلاعه الاربعة.

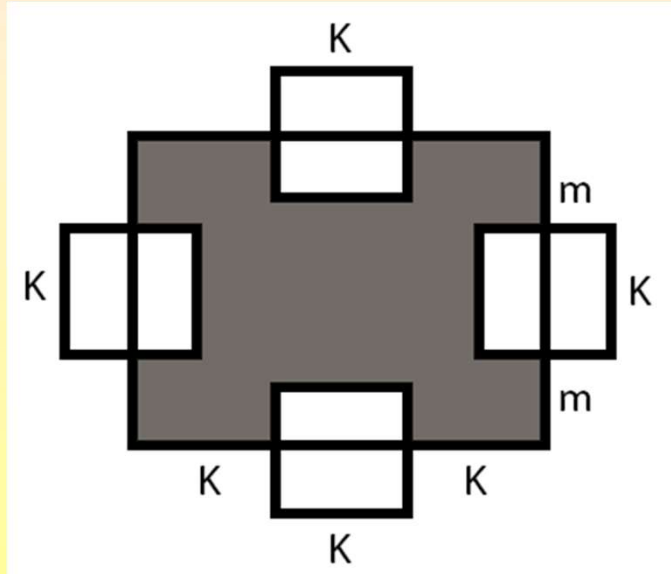
Q2) Write a program to read 3 lengths of a triangle. Then find the circumference and the area?

اكتب برنامج لقراءة 3 اطوال اضلاع مثلث. ثم جد المحيط والمساحة له؟ **أستخدم:**

$$s = a + b + c$$

$$area = \sqrt{(s - a)(s - b)(s - c)}$$

Q3) Calculate the gray area of the following figure:



اكتب برنامج لحساب المنطقة الرصاصية للشكل التالي؟
تنويه: أقرأ المتغيرات في الشكل، ثم جد مساحة المربعات الصغيرة واطرحها من مساحة المربع الكبير.

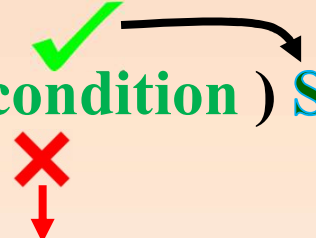
if statement

جملة if الشرطية

جملة (إذا كان.... فأن) في اللغة العربية.. هي اداة شرطية يتم فيها التحقق من **الشرط**، فاذا كان صحيحاً يتم تنفيذ جملة جواب الشرط. أو **عدم تنفيذ شيء**.

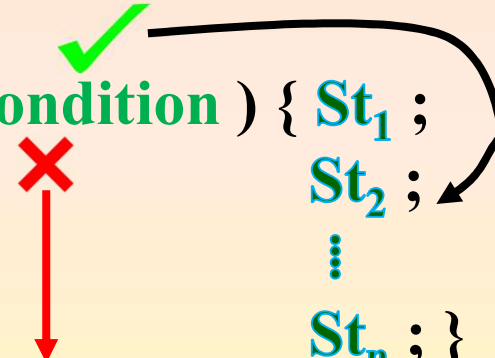
الصيغة العامة:

if (**condition**) **St** ;



إذا كان (**الشرط**) جملة جواب الشرط

if (**condition**) { **St₁** ;
St₂ ;
⋮
St_n ; }



إذا كان (**الشرط**) جمل جواب الشرط

Write a program to read 2 numbers. Then print the biggest one?

مثال:

```
#include <iostream.h>
int a , b ;
void main () {
    cin >> a >> b ;
    if ( a>b ) cout << a ;
    if ( b>a ) cout << b ;
}
```

أكتب برنامج لقراءة عددين. ثم أطلع أكبر واحد فيهم؟

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

Write a program to find and print the smallest number among 3 numbers ? مثال:

برنامج لايجاد وطباعة أصغر عدد من بين ثلاثة اعداد ؟

```
# include <iostream.h>
int a , b , c ;
void main ( ) {
    cin >> a >> b >> c ;
    if ( a < b && a < c ) cout << a ;
    if ( b < a && b < c ) cout << b ;
    if ( c < a && c < b ) cout << c ;
}
```

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

Write a program to read 3 marks. Then find the average, and print "PASS" مثال:

if it pass or print "FAIL" if it fail ?

برنامج لقراءة 3 درجات. ثم أحسب المعدل لها، وأطبِع "PASS" اذا المعدل ناجح أو أطيِع "FAIL" اذا كان راسباً؟

```
# include <iostream.h>
int m1 , m2 , m3 ;
float av ;
void main ( ) {
    cin >> m1 >> m2 >> m3 ;
    av = ( m1 + m2 + m3 ) / 3.0 ;
    if ( av >= 50 ) cout << "PASS" ;
    if ( av < 50 ) cout << "FAIL" ;
}
```

1- قراءة المعطيات (cin)

2- المعالجة

3- طباعة النتائج (cout)

مثال: You have a rectangle of (16 , 9) and a square of (12). Find the area of each, and print “R>S” if the area of the rectangle is bigger than the square’s, or print “R<S” if the area of the rectangle is smaller than the square’s, or print “R=S” if their areas are equal ?

لديك مستطيل بالابعاد (16 و 9) ولديك مربع ذات طول ضلع (12). جد مساحة كل منهما، واطبع “R>S” اذا كانت مساحة المستطيل اكبر من مساحة المربع، واطبع “R<S” اذا كانت مساحة المستطيل اصغر من مساحة المربع، واطبع “R=S” اذا تساوت مساحتهما ؟

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

```
int a , b , area1 ;
```

```
int x , area2 ;
```

```
void main ( ) {
```

```
    cin >> a >> b ;
```

```
    cin >> x ;
```

```
    area1 = a * b ;
```

```
    area2 = x * x ;
```

```
    if ( area1 > area2 ) cout << “R>S” ;
```

```
    if ( area1 < area2 ) cout << “R<S” ;
```

```
    if ( area1 == area2 ) cout << “R=S” ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

شاشة التنفيذ

16 9

12

R=S

مثال:

Write a program to read 2 integers. Then find and print:

- Their division **if** both is even ?
- Their multiplication **if** both is odd ?
- Their summation **if** the first is odd and the second is even ?
- Their subtraction **if** the first is even and the second is odd ?

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

```
int x , y ;
```

```
float result ;
```

```
void main ( ) {
```

```
    cin >> x >> y ;
```

```
    if ( x % 2 == 0 && y % 2 == 0 )    result = x / (y * 1.0) ;
```

```
    if ( x % 2 != 0 && y % 2 != 0 )    result = x * y ;
```

```
    if ( x % 2 != 0 && y % 2 == 0 )    result = x + y ;
```

```
    if ( x % 2 == 0 && y % 2 != 0 )    result = x - y ;
```

```
    cout << result ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Write a program to find and print the value of y from the following?

مثال:

برنامج لايجاد وطباعة قيمة y من المعادلة التالية؟

$$y = \begin{cases} \sqrt{a^3 + b} & \text{if } a > b \\ |a + b - 50| & \text{if } a < b \\ e^a - e^b & \text{if } a = b \end{cases}$$

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

```
# include <math.h>
```

```
int a , b ;
```

```
float y ;
```

```
void main () {
```

```
    cin >> a >> b ;
```

```
    if ( a > b )    y = sqrt ( pow ( a , 3 ) + b ) ;
```

```
    if ( a < b )    y = abs ( a + b - 50 ) ;
```

```
    if ( a == b )  y = exp ( a ) - exp ( b ) ;
```

```
    cout << y ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Write a program to read integer (m). Then find and print:

- If $m = 1$, then find the area of a SQUARE ?
- If $m = 2$, then find the area of a RECTANGLE ?
- If $m = 3$, then find the area of a TRIANGLE ?
- If $m = 4$, then find the area of a CIRCLE ?

مثال:

إذا $m=1$ جد مساحة مربع
إذا $m=2$ جد مساحة مستطيل
إذا $m=3$ جد مساحة مثلث
إذا $m=4$ جد مساحة دائرة

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

```
int m , a , x , y , b , h , r ;
```

```
float area ;
```

```
void main ( ) {
```

```
    cin >> m ;
```

```
    if ( m == 1 ) { cin >> a ;  
                    area = a * a ; }
```

```
    if ( m == 2 ) { cin >> x >> y ;  
                    area = x * y ; }
```

```
    if ( m == 3 ) { cin >> b >> h ;  
                    area = 0.5 * b * h ; }
```

```
    if ( m == 4 ) { cin >> r ;  
                    area = r * r * 3.14 ; }
```

```
    cout << area ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

ملاحظة مهمة:

جملة جواب الشرط (جملة التنفيذ للـ if)

يجب ان تكون جملة واحدة فقط.

فاذا كان جواب الشرط أكثر من جملة

واحدة، نضع كل جمل جواب الشرط بين

اقواس المجموعة... أي { }

if else statement

جملة if else الشرطية

جملة (إذا كان (الشرط) فإن..... وإلا.....) في اللغة العربية.. هي اداة شرطية يتم فيها التحقق من الشرط، فاذا كان صحيحاً يتم تنفيذ جملة جواب الشرط (فإن). أو تنفيذ جملة (وإلا).

الصيغة العامة:

```
if ( condition ) St ;  
else St ;
```

إذا كان (الشرط) جملة جواب الشرط
والآ جملة ثانية

```
if ( condition ) { St1 ;  
St2 ;  
⋮  
Stn ; }
```

```
else { St1 ;  
St2 ;  
⋮  
Stn ; }
```

إذا كان (الشرط) جمل جواب الشرط

ملاحظة مهمة:

الجملة التي بعد if أو الجملة التي بعد else يجب ان تكون جملة واحدة فقط.

فاذا كانت أكثر من جملة واحدة، نضع كل الجمل بعد if أو التي بعد else بين اقواس

المجموعة... أي { }

ملاحظة مهمة:

تستخدم جملة (if ... else ...) عندما يكون هناك في السؤال
مطلبين لاحتمالين متضادين.

زوجي فردي
يقبل القسمة لا يقبل القسمة
ذكر مؤنث
ناجح راسب
⋮

Write a program to read an integer. Then print “EVEN“ or “ODD” ?

مثال:

برنامج ليجاد لقراءة عدد صحيح. ثم أطلع اما “EVEN” أو “ODD” ؟

```
# include <iostream.h>
int a ;
void main () {
    cin >> a ;
    if ( a % 2 == 0 ) cout << “EVEN” ;
    if ( a % 2 != 0 ) cout << “ODD” ;
}
```

```
# include <iostream.h>
int a ;
void main () {
    cin >> a ;
    if ( a % 2 == 0 ) cout << “EVEN” ;
    else cout << “ODD” ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

مثال:

Write a program that reads 4 marks of a student. Then find the average and print whether he is “Pass” or “Fail” ?

برنامج لقراءة 4 درجات لطالب. ثم جد معدله وطباعة فيما اذا كان ناجحاً أم راسباً؟

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

```
int a , b , c , d ;
```

```
float av ;
```

```
void main ( ) {
```

```
    cin >> a >> b >> c >> d ;
```

```
    av = (a + b + c + d) / 4.0 ;
```

```
    if ( av < 50 ) cout << “Fail” ;
```

```
        else cout << “Pass” ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

```
sum = 1000 ;
```

ولو كان مقدار الراتب هو

```
ratio = 0.02;
```

وأن النسبة المئوية هي

فإن نسبة الزيادة في الراتب مع الزيادة هي:

```
sum = sum + sum * ratio ;
```

المبلغ الكلي بعد الزيادة

المبلغ الاصيل

المبلغ الاصيل * نسبة الزيادة = قيمة الزيادة

فكرة برمجية جديدة:

كيفية اضافة واحتساب نسبة مئوية لقيمة عددية:

فرضاً انه لديك راتب شهري معين مقداره

.sum

وأن الزيادة في هذا الراتب هي بنسبة 2%.

كيف يمكن حساب القيمة الكلية للراتب مع

الزيادة؟

مثال: Write a program to read a value and its increasing percentage. Then find the total value after adding this percentage?

برنامج لقراءة قيمة ونسبة الزيادة المئوية لها. أحسب قيمتها الكلية بعد اضافة هذه النسبة المئوية؟

```
# include <iostream.h>
float val , per ;
void main () {
    cin >> val >> per ;

    val = val + val * per ;
    cout << val ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

مثال: Write a program to read 2 values representing the number of two populations. Suppose the increasing percentage in both populations are 1% and 2%. Find the total amount in both populations and print out which one is bigger?

برنامج لقراءة قيمتين عددية تمثل تعدادين سكانيين. افترض ان النسبة المئوية في الزيادة لهما هي 1% و 2%. جد

القيمة الكلية لكل منهما (بعد الزيادة) واطبع ايهما أكبر؟

```
# include <iostream.h>
float p1 , p2 ;
void main () {
    cin >> p1 >> p2 ;
    p1 = p1 + p1 * 0.01 ;
    p2 = p2 + p2 * 0.02 ;
    if ( p1 > p2 ) cout << "p1 is greater" ;
    else cout << "p2 is greater" ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

مثال: Write a program to read the lengths of a rectangle and the radius of a circle. Then find their areas and print which one is smaller?

برنامج لقراءة اطوال اضلاع مستطيل و نصف قطر دائرة. ثم احسب مساحتهما واطبع أيهما أصغر؟

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

```
int a , b , r ;
```

```
int area1 ;
```

```
float area2 ;
```

```
void main ( ) {
```

```
    cin >> a >> b ;
```

```
    cin >> r ;
```

```
    area1 = a * b ;
```

```
    area2 = r *r * 3.14 ;
```

```
    if ( area1 < area2 ) cout << "Rectangle is the smaller" ;
```

```
        else cout << "Circle is the smaller" ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Write a program to read an integer number. Then check out whether this number is divisible by 3 or not ? **مثال:**

برنامج لقراءة عدد صحيح. ثم دقق فيما اذا كان هذا العدد يقبل القسمة على 3 أم لا ؟

```
# include <iostream.h>
int n ;
void main () {
    cin >> n ;
    if ( n% 3 == 0 ) cout << "Yes" ;
        else cout << "No" ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Find the value of (y) from the following equation ?

مثال:

جد قيمة (y) من المعادلة التالية ؟

$$y = \frac{\sqrt{a+b}}{a+b} \quad \text{if } a \geq b$$
$$y = \sin a + b \quad \text{otherwise}$$

```
# include <iostream.h>
# include <math.h>
float a , b , y ;
void main () {
    cin >> a >> b ;
    if ( a >= b ) y = sqrt ( a + b ) / ( a + b ) ;
        else y = sin ( a ) + b ;
    cout << y ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Nested if and nested if...else

if المتداخلة و if...else المتداخلة

`if (a < b && c > d) ;` \equiv `if (a < b) if (c > d) ;`
`if (av >= 50 && av <= 100) ;` \equiv `if (av >= 50) if (av <= 100) ;`
`if (expr1 && expr2 && expr3)` \equiv `if (expr1) if (expr2) if (expr3)`

أذن من الممكن تحويل كل أداة ربط (**and**) أي كل (**&&**) الى جملة **if** منفصلة.. لتقليل وقت المقارنات في الـ **if** الواحدة. أي **فصل الشروط بـ if جديدة**.

```
if ( condition1 ) expression1 ;  
else if ( condition2 ) expression2 ;  
else if ( condition3 ) expression3 ;  
⋮  
else if ( conditionn-1 ) expressionn-1 ;  
else expressionn ;
```

ملاحظة مهمة: تستخدم

طريقة (**if ... else if ...**) عندما يكون لدينا مجموعة من الشروط المتعددة ويوجد فيها شرط واحد مبهم أو كبير أو صعب، عندها نستخدم هذه الطريقة ونضع الشرط الصعب في الايخير (أي في اخر else) لكي نتخلص من كتابة الشرط الصعب هذا.

مثال:

Write a program that reads an integer number. Then check if this integer is divisible by 2 , 3 and 7 ?
كتابة برنامج لقراءة عدد صحيح. دقق اذا كان العدد يقبل القسمة على 2 و 3 و 7 ؟

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

```
int n ;
```

```
void main () {
```

```
    cin >> n ;
```

```
    if ( n%2 == 0 && n%3 == 0 && n%7 == 0 ) cout << "YES";
```

```
        else cout << "NO" ;
```

```
}
```

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

```
int n ;
```

```
void main () {
```

```
    cin >> n ;
```

```
    if ( n%2 == 0 ) if ( n%3 == 0 ) if ( n%7 == 0 ) cout << "YES";
```

```
        else cout << "NO" ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

أو

Write a program that reads 4 marks of a student. Then find the average and print his grade ?

مثال:

برنامج لقراءة 4 درجات لطالب. ثم جد معدله وطباعة تقديره (راسب، مقبول، متوسط، جيد، جيد جداً، امتياز) ؟

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

```
int a , b , c , d ;
```

```
float av ;
```

```
void main ( ) {
```

```
cin >> a >> b >> c >> d ;
```

```
av = (a + b + c + d) / 4.0 ;
```

```
if ( av < 50 ) cout << "Fail" ;
```

```
else if ( av >= 50 && av < 60 ) cout << "Fair" ;
```

```
else if ( av >= 60 && av < 70 ) cout << "Midium" ;
```

```
else if ( av >= 70 && av < 80 ) cout << "Good" ;
```

```
else if ( av >= 80 && av < 90 ) cout << "V. Good" ;
```

```
else cout << "Excellent" ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Write a program that reads a character. Then print whether this character is **مثال:**
a CAPITAL letter , SMALL letter , a DIGIT or a PUNCTUATION mark ?

برنامج لقراءة حرف. ثم اطبع فيما اذا كان الحرف يمثل حرف كبير ، حرف صغير ، حرف رقمي أو علامة تنقيط ؟

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

```
char c ;
```

```
void main ( ) {
```

```
cin >> c ;
```

```
if ( c >= 'A' && c <= 'Z' ) cout << "CAPITAL" ;
```

```
else if ( c >= 'a' && c <= 'z' ) cout << "SMALL" ;
```

```
else if ( c >= '0' && c <= '9' ) cout << "DIGIT" ;
```

```
else cout << "PUNCTUATION" ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

في مثل هذا المثال:

جميع الحروف الصغيرة والحروف الكبيرة والحروف الرقمية معروفة لدينا، وانما يبقى حروف علامات التنقيط غير معلومة لدينا كلها.. لذا نستخدم جملة if....else المتداخلة وبدل من كتابة if لعلامات التنقيط الغير معروفة لدينا.. نضعها في اخر else.

Write a program to find (y) from the following equation ?

مثال:

$$y = \begin{cases} a + b & \text{if } i = 1 \\ \sqrt{a} + \sqrt{b} & \text{if } i = 2, 3 \\ e^{a+b} & \text{if } i = 4, 5 \\ 0 & \text{otherwise} \end{cases}$$

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

```
# include <math.h>
```

```
int a , b , i ;
```

```
float y ;
```

```
void main ( ) {
```

```
    cin >> a >> b >> i ;
```

```
    if ( i == 1 ) y = a + b ;
```

```
        else if ( i == 2 || i == 3 ) y = sqrt ( a ) + sqrt ( b ) ;
```

```
            else if ( i == 4 || i == 5 ) y = exp ( a + b ) ;
```

```
                else y = 0 ;
```

```
    cout << y ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

في هذا المثال:

بسبب وجود كلمة

otherwise، كان لابد

وحتماً استخدام طريقة

if...else if لأن باقي

الاحتمالات كبيرة جداً

وغير منتهية.

Write a program that find the area of a rectangle if (m=1) or find the area of a **مثال:** circle if (m=2) or find the area of a square if (m=3) or find the area of a triangle otherwise ?

برنامج ليجاد مساحة مستطيل اذا كانت (m=1) أو مساحة دائرة اذا كانت (m=2) أو مساحة مربع اذا كانت (m=3) أو مساحة مثلث خلافاً لذلك (ماعداد ذلك) ؟

```
# include <iostream.h>
int a , b , r , w , k , h ;
float area ;
void main ( ) {
    cin >> m ;
    if ( m == 1 ) { cin >> a >> b ;
                    area = a * b ; }
    else if ( m == 2 ) { cin >> r ;
                        area = r * r * 3.14 ; }
    else if ( m == 3 ) { cin >> w ;
                        area = w * w ; }
    else { cin >> k >> h ;
          area = 0.5 * k * h ; }

    cout << area ;
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

مثال: Write a program that reads the number of a month. Then print how many days does this month consist of ?

برنامج للقراءة رقم شهر. ثم يقوم بطباعة عدد الايام التي يتكون منها هذا الشهر ؟

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

```
int m ;
```

```
void main () {
```

```
cin >> m ;
```

```
if ( m == 2 ) cout << 28 ;
```

```
if ( m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12 ) cout << 31 ;
```

```
if ( m == 4 || m == 6 || m == 9 || m == 11 ) cout << 30 ;
```

```
}
```

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

```
int m ;
```

```
void main () {
```

```
cin >> m ;
```

```
if ( m == 2 ) cout << 28 ;
```

```
else if ( m == 4 || m == 6 || m == 9 || m == 11 ) cout << 30 ;
```

```
else cout << 31 ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

أو

تمارين

Q1) Write a program to read the lengths of a triangle. Then check and print the following:

- Are these lengths form a triangle? If yes, complete programming the following.
- Does this triangle is equal sided? Print “Equal sided”.
- Does all the sides of this triangle are equal? Print “Equal sides”.
- Does all the sides of this triangle are different? Print “Different sides”.
- Does the triangle is a right angle? Print “Right angle”.

برنامج لقراءة اطوال اضلاع مثلث. ثم بين وأطبع مايلي:

- هل هذه الاطوال تشكل مثلثاً؟ اذا نعم أكمل برمجة ما يلي.
- هل المثلث هو متساوي الساقين؟ اطبع “Equal sided”.
- هل المثلث متساوي الاضلاع؟ اطبع “Equal sides”.
- هل المثلث مختلف الاضلاع؟ اطبع “Different sides”.
- هل المثلث قائم الزاوية؟ اطبع “Right angle”.

Q2) Write a program that reads 2 integers (x , y) and 1 character (c). Then find:

- $x + y$ if $c = '+'$.
- $x - y$ if $c = '-'$.
- $x * y$ if $c = '*'$.
- x / y if $c = '/'$.
- $x \% y$ if $c = '\%'$.
- “error” otherwise.

برنامج لقراءة عددين صحيحين (x , y) و حرف رمزي واحد (c). ثم جد:

Switch.... Case Statement

جملة Switch.... Case

تعتبر جملة **switch..case** حلاً بديلاً عن عبارات **if** الطويلة والمتداخلة والتي تقارن متغيراً واحداً بعدة قيم متسلسلة.

الصيغة العامة لها :

```
switch (n) {  
  case 1: تنفيذ كود برمجي if n = 1; break;  
  case 2: تنفيذ كود برمجي if n = 2; break;  
  case 3: تنفيذ كود برمجي if n = 3; break;  
  ⋮  
  case m: تنفيذ كود برمجي if n = m; break;  
  default: تنفيذ كود برمجي if n لا تطابق اي حالة من الحالات السابقة  
}
```

n يجب ان تكون
صحيحة من نوع
integer

توفر جملة **switch..case** طريقة سهلة لتنفيذ أجزاء مختلفة من الكودات البرمجية بناءً على قيمة التعبير الموجود في **switch(expression)** أو القيمة الصحيحة **switch(n)**. فإذا لم تتحقق أي حالة من الحالات في أي **case**، فممكن كتابة جملة **default** كآخر اختيار. ومن الممكن الاستغناء عنها حسب متطلبات الحل.

جملة **break** : فتستخدم للدلالة على **نهاية case** الحالية **والخروج** من **switch** وانهاؤها.

Write a program to find and print the value of y from the following?

مثال:

برنامج لايجاد وطباعة قيمة y من المعادلة التالية؟

$$y = \begin{cases} \sqrt{a^3 + b} & \text{if } k = 1 \\ |a + b - 50| & \text{if } k = 2 \\ e^a - e^b & \text{if } k = 3 \end{cases}$$

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

```
# include <math.h>
```

```
int a , b , k ;
```

```
float y ;
```

```
void main ( ) {
```

```
    cin >> k ;
```

```
    cin >> a >> b ;
```

```
    switch ( k ) {
```

```
        case 1 :    y = sqrt ( a * a * a + b ) ; break ;
```

```
        case 2 :    y = abs ( a + b - 50 ) ; break ;
```

```
        case 3 :    y = exp ( a ) - exp ( b ) ; break ;
```

```
    }
```

```
    cout << y ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

Write a program to find (y) from the following equation ?

مثال:

$$y = \begin{cases} a + b & \text{if } i = 1 \\ \sqrt{a} + \sqrt{b} & \text{if } i = 2, 3 \\ e^{a+b} & \text{if } i = 4, 5 \\ 0 & \text{otherwise} \end{cases}$$

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

```
# include <math.h>
```

```
int a , b , i ;
```

```
float y ;
```

```
void main ( ) {
```

```
cin >> a >> b >> i ;
```

```
switch ( i ) {
```

```
case 1 : y = a + b ; break ;
```

```
case 2 :
```

```
case 3 : y = sqrt ( a ) + sqrt ( b ) ; break ;
```

```
case 4 :
```

```
case 5 : y = exp ( a + b ) ; break ;
```

```
default : y = 0 ;
```

```
}
```

```
cout << y ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

في هذا المثال:

بسبب وجود كلمة **otherwise**،
كان لابد وحتماً استخدام جملة
default لأن باقي الاحتمالات
كبيرة جداً وغير منتهية.

Write a program that find the area of a rectangle if (m=1) or find the area of a **مثال:** circle if (m=2) or find the area of a square if (m=3) or find the area of a triangle otherwise ?

برنامج ليجاد مساحة مستطيل اذا كانت (m=1) أو مساحة دائرة اذا كانت (m=2) أو مساحة مربع اذا كانت (m=3) أو مساحة مثلث خلافاً لذلك (ماعداد ذلك) ؟

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

```
int a , b , r , w , k , h , m ;
```

```
float area ;
```

```
void main ( ) {
```

```
    cin >> m ;
```

```
    switch ( m ) {
```

```
        case 1 :  cin >> a >> b ; area = a * b ; break ;
```

```
        case 2 :  cin >> r ; area = r * r * 3.14 ; break ;
```

```
        case 3 :  cin >> w ; area = w * w ; break ;
```

```
        default : cin >> k >> h ; area = 0.5 * k * h ;
```

```
    }
```

```
    cout << area ;
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

مثال: Write a program that reads the number of a month. Then print how many days does this month consist of ?

برنامج للقراءة رقم شهر. ثم يقوم بطباعة عدد الايام التي يتكون منها هذا الشهر ؟

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

```
int m ;
```

```
void main () {
```

```
cin >> m ;
```

```
switch ( m ) {
```

```
case 2 : cout << 28 ; break ;
```

```
case 4 :
```

```
case 6 :
```

```
case 9 :
```

```
case 11: cout << 30 ; break ;
```

```
default : cout << 31 ; break ;
```

```
}
```

```
}
```

1- قراءة (cin)

2- المعالجة

3- طباعة (cout)

لغرض فهم مغزى وعمل العدادات... حل المثال التالي:

Write a program to read 5 marks. Then find the **number** of the marks greater or equal to 50? برنامج لقراءة 5 درجات. ثم جد **عدد** الدرجات الاكبر أو التي تساوي 50 (الناجحة)

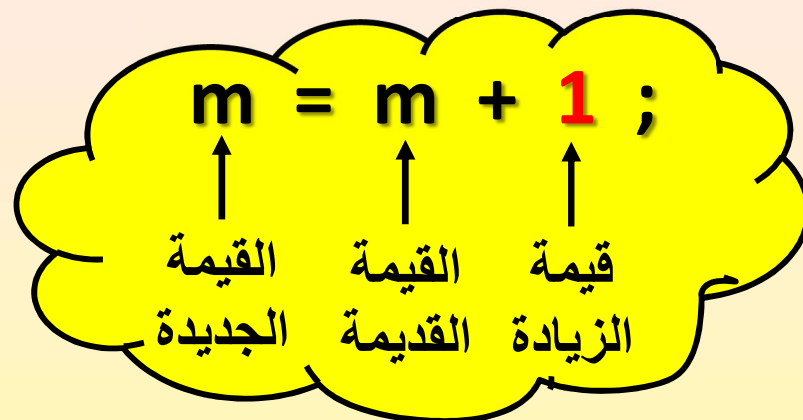
لحل هذا السؤال بطريقة if العادية والمتشعبة.. سيكون حلا طويلا ومربكاً: حيث نحتاج الى ان تكون كل الدرجات ناجحة فيكون عدد الدرجات الناجحة يساوي 5. أو قد تكون واحدة منهم راسبة (من هي؟؟؟) ليكون العدد يساوي 4. أو تكون درجتان راسبة (من هما؟؟؟) ليكون عدد الدرجات الناجحة 3. وهكذا ...

أذن يجب ان نجد طريقة أسهل وهي: **عد** الدرجات الناجحة واحدة بعد الاخرى باستخدام **عداد**.

```
# include <iostream.h>
int a , b , c , d , e ;
int m ;
void main ( ) {
    cin >> a >> b >> c >> d >> e ;
    m = 0 ;
    if ( a >= 50 ) m = m + 1 ;
    if ( b >= 50 ) m = m + 1 ;
    if ( c >= 50 ) m = m + 1 ;
    if ( d >= 50 ) m = m + 1 ;
    if ( e >= 50 ) m = m + 1 ;
    cout << m ;
}
```

تصفير العداد
قبل البدء بالعد

زيادة العداد
بـ 1 عند العد



m = m + 1 ;
جملة الأحلال الحسابي
كل جديد يصبح قديم اذا تم تنفيذ
جملة الاحلال مرة ثانية

عداد العد:

مثلاً.. عندما نطلب من شخص ان يبدأ بالعد من 1 الى 10.

فيبدأ هذا الشخص بتصفير عداد ما... ← $k = 0$;

ثم يقوم بزيادة 1 الى ذهنه... ← $k = k + 1$;
أي أن k تتم اضافة 1 الى قيمته القديمة،

القيمة القديمة مع الزيادة ب 1 تصبح قيمة k الجديدة..

وهكذا... فكلما يتم الرجوع الى جملة $k = k + 1$; يتم فيها
(احلال / جعل) القيمة التي كانت جديدة لتصبح قديمة ...
ولتستخرج قيمة أجدد منها...

دائرة
Loop

∴ العد تكون الزيادة
في العداد دائماً 1.

وعادة ما تستخدم هذه العدادات ضمن
الدورات / Loops : وهي مقاطع برمجية يعاد
تنفيذها في البرنامج الواحد أكثر من مرة.

عداد للمجموع التكراري:

مثلاً.. عندما نطلب من شخص ان يبدأ بجمع درجات مجموعة من الطلاب.

فيبدأ هذا الشخص بتصفير عداد للمجموع.

$s = 0 ;$

ولا ننسى تصفير عداد للدائرة لتدور لكذا من المرات

$i = 0 ;$

ثم يقوم بزيادة 1 الى ذهنه... أي البدء بقراءة الدرجة الاولى للطالب الاول.. فيعتبر عداد القراءة.

$i = i + 1 ;$

$cin \gg X ;$

قراءة الدرجة للطالب الذي تسلسله هو i ،

إذا أصبح i قيمته 1 يعني سيتم قراءة درجة الطالب الاول
إذا أصبح i قيمته 2 يعني سيتم قراءة درجة الطالب الثاني
إذا أصبح i قيمته 3 يعني سيتم قراءة درجة الطالب الثالث
وهكذا...

\vdots

$s = s + X ;$

\vdots

دائرة
Loop

∴ للمجموع تكون الزيادة
في العداد دائماً القيمة
المقروءة X .

• أذن جملة الاحلال الحسابي ($i = i + 1 ;$) لعد الطلاب ليتم قراءة درجاتهم. (عداد القراءة)

• بينما جملة الاحلال الحسابي ($s = s + X ;$) هي لجمع الدرجات.

While Statement

جملة الدارة while

```
while ( condition شرط ) {
```

```
    St1 ;
```

```
    ⋮
```

```
    Stn ;
```

```
}
```

الصيغة العامة هي:

```
{ فنفذ التالي ( الشرط صحيحاً ) ما دام
```

```
    جملة 1 ;
```

```
    ⋮
```

```
    جملة n ;
```

```
}
```

مثال: لقراءة 5 أعداد... يجب عمل التالي:

i = 0 ; ← تصفير عداد القراءة

```
while ( i < 5 ) {
```

```
    i = i + 1 ;
```

```
    cin >> var ;
```

```
    ⋮
```

```
    ⋮
```

```
}
```

متى نتوقف عن القراءة ???

مادام لم نصل الى 5 نستمر بالقراءة

أي

مادام عداد القراءة أقل من 5 نستمر

Write a program to read 15 marks. Then find their summation? مثال

برنامج لقراءة 15 درجة. ثم جد حاصل جمعهم؟

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

```
int a , i ;
```

```
int sum ;
```

```
void main ( ) {
```

```
sum = 0 ;
```

← تصفير عداد لمجموع الدرجات

```
i = 0 ;
```

← تصفير عداد القراءة

```
while ( i < 15 ) {
```

```
    i = i + 1 ;
```

```
    cin >> a ;
```

```
    sum = sum + a ;
```

```
}
```

```
cout << sum ;
```

```
}
```

1- عدادات

2- قراءة (cin)

3- المعالجة

4- طباعة (cout)

Write a program that reads 10 integers. Then find:

1-No. of evens ?

عدد الاعداد الزوجية

2-Sum of odds?

مجموع الاعداد الفردية

```
void main () {
```

```
    k = 0 ;
```

تصفير عدد لعد الاعداد الزوجية

```
    sum = 0 ;
```

تصفير عدد لمجموع الاعداد الفردية

```
    i = 0 ;
```

تصفير عدد القراءة

```
    while ( i < 10 ) {
```

```
        i = i + 1 ;
```

زيادة عدد القراءة بـ 1

```
        cin >> x ;
```

```
        if ( x % 2 == 0 ) k = k + 1 ;
```

```
        if ( x % 2 != 0 ) sum = sum + x ;
```

```
    }
```

```
    cout << k << sum ;
```

```
}
```

1-عدادات

2-قراءة (cin)

3-المعالجة

4-طباعة (cout)

Write a program that reads 50 marks. Then find the average of the passed marks?

برنامج لقراءة 50 درجة. ثم جد معدل الدرجات الناجحة؟

```
void main () {
```

```
s = 0 ;
```

تصفير عداد لمجموع الدرجات الناجحة

```
n = 0 ;
```

تصفير عداد لعدد الدرجات الناجحة

```
i = 0 ;
```

تصفير عداد القراءة

```
while ( i < 50 ) {
```

```
i = i + 1 ;
```

زيادة عداد القراءة بـ 1

```
cin >> mark ;
```

```
if ( mark >= 50 ) { s = s + mark ;
```

```
n = n + 1 ; }
```

```
}
```

```
av = s / ( n * 1.0 ) ;
```

```
cout << av ;
```

```
}
```

1- عدادات
2- قراءة (cin)
3- المعالجة
4- طباعة (cout)

من الطبيعي ان حساب معدل أي شيء
(هنا الدرجات) نحتاج الى معرفة :

1- مجموع الدرجات

2- عدد الدرجات

لكي يتم في اخر مرحلة من الحل ايجاد:

المعدل = المجموع / العدد

Write a program that reads 20 numbers. Then find:

1- Sum of positives ?

2- No. of zeros ?

3- Average of negatives ?

برنامج لقراءة 20 عدد. ثم جد:

1- مجموع الاعداد الموجبة ؟

2- عدد الازفرار ؟

3- معدل الاعداد السالبة ؟

```
void main () {
```

```
    s = 0 ;
```

تصفير عدد لمجموع الاعداد الموجبة

```
    n = 0 ;
```

تصفير عدد لعد الازفرار

```
    w = 0 ; k = 0 ;
```

تصفير عدد لمجموع السالبة

```
    i = 0 ;
```

تصفير عدد لعد السالبة

```
    while ( i < 20 ) {
```

تصفير عدد القراءة

```
        i = i + 1 ;
```

زيادة عدد القراءة ب 1

```
        cin >> num ;
```

```
        if ( num > 0 ) s = s + num ;
```

```
            else if ( num == 0 ) n ++ ;
```

أو → n = n + 1 ;

```
                else { w = w + num ;
```

```
                    k = k + 1 ; }
```

```
        }
```

```
        av = w / ( k * 1.0 ) ;
```

```
        cout << s << n << av ;
```

```
    }
```

1- عدادات
2- قراءة (cin)
3- المعالجة
4- طباعة (cout)

Write a program that reads 15 numbers. Then find:

- 1- Sum of even numbers that divisible by 5 ?
- 2- Sum of positive numbers and greater than 35 ?
- 3- No. of numbers in the interval [20 to 30] ?

برنامج لقراءة 15 عدد. ثم جد:

- 1- مجموع الاعداد الزوجية والتي تقبل القسمة على 5 ؟
- 2- مجموع الاعداد الموجبة والتي تكون أكبر من 35 ؟
- 3- عدد الاعداد التي تقع ضمن الفترة [20 الى 30] ؟

```

void main () {
    se5 = 0 ;
    sp35 = 0 ;
    no = 0 ;
    i = 0 ;
    while ( i < 15 ) {
        i = i + 1 ;
        cin >> a ;
        if ( a % 2 == 0 && a % 5 == 0 ) se5 = se5 + a ;
        if ( a > 0 && a > 35 ) sp35 = sp35 + a ;
        if ( a >= 20 && a <= 30 ) no ++ ;
    }
    cout << se5 << sp35 << no ;
}
    
```

تصفير عداد لمجموع المطلب الاول

تصفير عداد لمجموع المطلب الثاني

تصفير عداد لعد المطلب الثالث

تصفير عداد القراءة

زيادة عداد القراءة ب 1

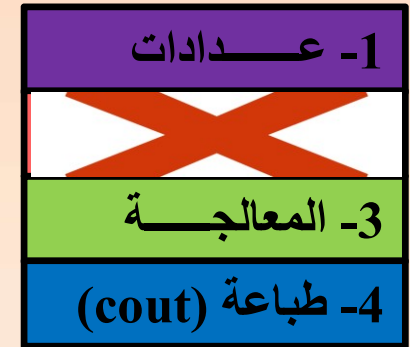
1- عدادات
2- قراءة (cin)
3- المعالجة
4- طباعة (cout)

Write a program to print the numbers from [1 to 12]?

مثال

برنامج لطباعة الاعداد من [1 الى 12] ؟

```
void main () {  
    i = 0 ;  
    while ( i < 12 ) {  
        i = i + 1 ;  
        cout << i << endl ;  
    }  
}
```



Write a program to print the even numbers from [10 to 100]?

مثال

برنامج لطباعة الاعداد الزوجية من [10 الى 100] ؟

```
void main () {  
    i = 8 ;  
    while ( i < 100 ) {  
        i = i + 2 ;  
        cout << i << endl ;  
    }  
}
```

ملاحظة مهمة :

في الأسئلة التي فيها فترة محددة [من - الى]

1- لا يوجد قراءة للمتغيرات.

2- يستخدم عداد الدارة (i) في المعالجة

A program to find the sum of odd numbers in the interval [18 to 150]?

برنامج لايجاد مجموع الاعداد الفردية ضمن الفترة [18 الى 150] ؟

```
void main () {  
    s = 0 ;  
    i = 17 ;  
    while ( i < 149 ) {  
        i = i + 2 ;  
        s = s + i ;  
    }  
    cout << s ;  
}
```

لاحظ ان العداد **i** بدأ قبل جملة الـ **while** بـ **17**، لان الـ **18** ليس عدد فردي، ولهذا يتم الأخذ بنظر الاعتبار أول عدد فردي ضمن الفترة وهو **19**... وعليه بدأ العداد **i = 17** حتى عند أول زيادة بـ **2** تصبح قيمته **19** وهي أول قيمة تؤخذ بالحسبان ضمن الفترة.

Write a program to find the number of integers that divisible by 5 in the interval [13 to 93]?

برنامج لايجاد عدد الاعداد التي تقبل القسمة على 5 ضمن [13 الى 93] ؟

```
void main () {  
    n = 0 ;  
    i = 10 ;  
    while ( i < 90 ) {  
        i = i + 5 ;  
        n ++ ;  
    }  
    cout << n ;  
}
```

فعلياً ← 15 الى 90

Write a program to read 10 integers. Then find the summation of integers that included in the interval (20 – 40]?

برنامج لقراءة 10 أعداد. ثم جد مجموع الأعداد المتضمنة في الفترة (20 – 40] ؟

1- أعدادات
2- قراءة (cin)
3- المعالجة
4- طباعة (cout)

```
void main ( ) {  
    sum = 0 ;  
    i = 0 ;  
    while ( i < 10 ) {  
        i = i + 1 ;  
        cin >> a ;  
        if ( a > 20 && a <= 40 ) sum = sum + a ;  
    }  
    cout << sum ;  
}
```

ملاحظة .. يخطئ فيها معظم الطلاب:

في الاسئلة التي فيها قراءة، وفترة محددة...
فإن الفترة المحددة لا توضع بالدارة while
وأنما تكون ضمن جملة if.

Write a program to read 20 integers. Then find

1- No. of evens and positive ?

2- Sum of odds and negative ?

3- Multiplication of integers in the interval [5 – 12] ?

4- Average of integers consist of one digit ?

برنامج لقراءة 20 عدد صحيح. ثم جد

1- عدد الاعداد الزوجية والموجبة ؟

2- مجموع الاعداد الفردية والسالبة ؟

3- حاصل ضرب الاعداد ضمن الفترة [5 – 12] ؟

4- معدل الاعداد المكونة من رقم واحد (مرتبة آحاد فقط) ؟

```
void main () {
```

```
    n = 0 ;
```

```
    sum = 0 ;
```

```
    mult = 1 ;
```

```
    w = 0 ; m = 0 ;
```

```
    i = 0 ;
```

```
    while ( i < 20 ) {
```

```
        i = i + 1 ;
```

```
        cin >> x ;
```

```
        if ( x % 2 == 0 && x > 0 ) n ++ ;
```

```
        if ( x % 2 != 0 && x < 0 ) sum = sum + x ;
```

```
        if ( x >= 5 && x <= 12 ) mult = mult * x ;
```

```
        if ( x / 10 == 0 ) { w = w + x ; m ++ ; }
```

```
    }  
  
    cout << n << sum << mult << w / ( m * 1.0 ) ;
```

ملاحظة مهمة جداً... أغلب الطلبة

يخطئون بها:

عند وجود عدة مطالب في السؤال الواحد...
لا نحل كل مطلب ببرنامج معزول وعلى حدى..
بل تخضع جميع مطالب السؤال داخل دارة
واحدة وضمن برنامج واحد فقط.

تذكر دائماً

int / int = int

0 / 10 = 0

1 / 10 = 0

5 / 10 = 0

9 / 10 = 0

For Statement (Loop)

جملة For

وهي احدى جمل التكرار والتي تستخدم لغرض تكرار مقطع برمجي لعدة مرات لحين توقف التكرار بشرط معين. وهي تشبه الى حد كبير جملة `while`

الصيغة العامة:

```
for ( statement 1 ; statement 2 ; statement 3 )  
    St ; (الجملة المراد تنفيذها خلال التكرار)
```

حيث أن:

Statement 1: هي عداد بداية الدارة أو اول قيمة للبداية بالتكرار

Statement 2: هو شرط التوقف

Statement 3: هي قيمة زيادة العداد

مثلاً: عداد القراءة ... لعد 15 دارة

```
i = 0 ;  
while ( i < 15 ) {  
    i = i + 1 ;  
    ---  
    ---  
    ---  
}  
for ( i = 1 ; i <= 15 ; i = i + 1 ) {  
    ---  
    ---  
    ---  
}
```

ملاحظة

في جملة `while` فإن العداد `i=0` لا يعتبر بداية للعداد وانما تهيئة لانطلاقه العداد، وبعدها وبداخل `while` فان الزيادة للعداد `i=i+1` هي 1 فستكون انطلاقة `i` ضمن الـ `while` بـ 1. بينما في جملة `for` فإن العداد قد بدأ فعلياً بقيمة البداية وهي `i=1`.

ونلاحظ تطابق جملة شرط التوقف (2) في الدارتين إلا ان بجملة الـ `for` فنضع (أصغر أو يساوي، `<=`) بينما في الـ `while` فنضع فقط (أصغر من، `<`). وذلك بسبب ان جملة الزيادة `i=i+1` تنفذ بعد مقارنة شرط التوقف.. بينما في الـ `for` فان جملة الزيادة تنفذ قبل مقارنة شرط التوقف.

ونلاحظ تطابق جملة زيادة العداد (3) في كلا الدارتين.

```

for ( i = 1 ; i <= 15 ; i = i + 1 ) {
    ---
    ---
    ---
}

```

i=1 هي قيمة بداية الدارة فأن **for** سوف تقوم بمقارنة **i=1** مع شرط التوقف، فإذا كان الشرط صحيحاً فستنفذ الدارة. ومن ثم تعطي الزيادة للعداد ومقارنته قبل تنفيذ الدارة للمرة الثانية والثالثة والرابعة وهكذا... لحين ان يصبح شرط التوقف خاطئاً فتتوقف. أي بمعنى ان **for** تقوم بالزيادة ثم المقارنة ولغاية التوقف.

مثال

Write a program to print the numbers from [1 – 10]?

```

void main( ) {
    i = 0 ;
    while ( i < 10 ) {
        i = i + 1 ;
        cout << i ;
    }
}

```

```

void main( ) {
    for ( i = 1 ; i <= 10 ; i = i + 1 )
        cout << i ;
}

```

نلاحظ انه لا توجد الحاجة لوضع الاقواس ضمن جملة **for** لانه توجد جملة واحدة فقط ضمنها. ولكن ممكن وضع الاقواس على أية حال.

مثال

Write a program to print the **even** numbers from [1 – 10]?

```

void main( ) {
    i = 0 ;
    while ( i < 10 ) {
        i = i + 2 ;
        cout << i ;
    }
}

```

```

void main( ) {
    for ( i = 2 ; i <= 10 ; i = i + 2 )
        cout << i ;
}

```

نلاحظ ان العداد في جملة **for** قد بدأ فعلياً من العدد الزوجي الأول ضمن الفترة وهو **2**. بينما في **while** فقد ابتدأ افتراضياً بقيمة **0** ثم ومن خلال اول دارة ستصبح قيمته الفعلية الاولى **2**.

مثال

Write a program to find the summation of the **odd** numbers from [20 – 100]?

[21 – 99]

```
void main( ) {  
s = 0 ;  
i = 19 ;  
while ( i < 99 ) {  
    i = i + 2 ;  
    s = s + i ;  
}  
cout << s ;  
}
```

```
void main( ) {  
s = 0 ;  
for ( i = 21 ; i <= 99 ; i = i + 2 )  
    s = s + i ;  
cout << s ;  
}
```

مثال

Write a program to find the summation of the **even** numbers that **divisible by 3** from [115 – 2020]?

↓ ↓
[120 – 2016]

تفكير:

الاعداد الزوجية: مضاعفات 2.

الاعداد التي تقبل القسمة على 3: هي من مضاعفات 3.

أذن يجب ايجاد المضاعف المشترك الأصغر للعددين 2 و 3 والذي هو 6.

```
void main( ) {  
s = 0 ;  
i = 114 ;  
while ( i < 2016 ) {  
    i = i + 6 ;  
    s = s + i ;  
}  
cout << s ;  
}
```

المضاعف المشترك الأصغر للعددين 3 و 2

آخر عدد زوجي يقبل القسمة على 3 ضمن الفترة

أول عدد زوجي يقبل القسمة على 3 خارج الفترة

```
void main( ) {  
s = 0 ;  
for ( i = 120 ; i <= 2016 ; i = i + 6 )  
    s = s + i ;  
cout << s ;  
}
```

المضاعف المشترك الأصغر للعددين 3 و 2

أول عدد زوجي يقبل القسمة على 3 ضمن الفترة

آخر عدد زوجي يقبل القسمة على 3 ضمن الفترة

Write a program to read 15 numbers. Then, find the summation of all numbers that divisible by 3?

```
void main( ) {
s = 0 ;
i = 0 ;
while ( i < 15) {
    i = i + 1 ;
    cin >> x ;
    if( x%3 == 0) s = s + x ;
}
cout << s ;
}
```

```
void main( ) {
s = 0 ;
for ( i = 1 ; i <= 15 ; i = i + 1) {
    cin >> x ;
    if( x%3 == 0) s = s + x ;
}
cout << s ;
}
```

مثال: سؤال امتحان نهائي الكورس الأول:

Write a program that reads 15 numbers, then find:

- Sum of even numbers?
- No. of non-zero numbers?
- Sum of numbers that divisible by 3 in the interval (25-100]
- Sum of numbers that consist of three digits?

```
void main( ) {
s = 0 ; k = 0 ; w = 0 ; m = 0 ;
for ( i = 1 ; i <= 15 ; i = i + 1) {
    cin >> x ;
    if( x%2 == 0) s = s + x ;
    if( x != 0 ) k = k + 1 ;
    if( x%3 == 0 && x > 25 && x <= 100 ) w = w + x ;
    if( x > 99 && x <= 999 ) m = m + x ;
}
cout << s << k << w << m ;
}
```

Nested for

for المتداخلة

```
for (زيادة العداد 1 ; شرط التوقف ; بداية العداد 1) {  
    for (زيادة العداد 2 ; شرط التوقف ; بداية العداد 2) {  
        ---- ;  
        ---- ;  
    }  
}
```

عادة تستخدم جمل for المتداخلة عند التعامل مع قيم فردية ولكن لكل قيمة لها أكثر من (عملية/ معالجة/ أو قيم). وعادة تستخدم مع القيم التي تشكل صفوف ولكل صف له عدة أعمدة. مثال ذلك **جدول الضرب** حيث ان لكل قيمة من الجدول لها عدة قيم مضروبة. وتستخدم ايضاً في تمثيل المصفوفات ثنائية الأبعاد.

```
for ( i = 1 ; i <= 3 ; i = i + 1 )  
    cout << "*" ;
```

```
***
```

لاحظ هذا المثال استخدمت دارة for لغرض طباعة * لثلاث مرات وسيكون الناتج كما يلي وواضح للجميع.

```
for ( i = 1 ; i <= 3 ; i = i + 1 )  
    cout << "*" << endl ;
```

```
*  
*  
*
```

في هذا المثال استخدمت دارة for ايضا لغرض طباعة * لثلاث مرات وسيكون الناتج كما يلي وواضح للجميع بسبب استخدام جملة النزول الى سطر جديد (endl).

```
for ( i = 1 ; i <= 3 ; i = i + 1 )  
    for ( j = 1 ; j <= 2 ; j = j + 1 )  
        cout << "*" ;
```

```
*****
```

نلاحظ ان دارة العداد i سوف تكون لـ 3 مرات وفي كل دارة من دارات العداد i فإن العداد j ستكون له دارتين 2. عليه ستكون عدد الدارات الكلية هي 6. وفيها يتم طباعة ستة *.

```
for ( i = 1 ; i <= 3 ; i = i + 1 )  
    for ( j = 1 ; j <= 2 ; j = j + 1 )  
        cout << "*" << endl ;
```

```
*  
*  
*  
*  
*
```

هنا الطباعة ستة * ولكن كل واحدة على سطر بسبب استخدام جملة endl.


```

for ( i = 1 ; i <= 3 ; i = i + 1 ) {
    1 for ( j = 1 ; j <= 2 ; j = j + 1 )
        cout << "*" ;
    2 cout << endl ;
}

```

```

* *
* *
* *

```

هنا دائرة العداد i ستتكرر لـ 3 مرات.
وفي كل مرة سيتنفذ فيها العداد j مرتين
2، مع تنفيذ جملة `endl`. بمعنى ان لكل
دائرة من i سيتم تنفيذ جملتين 1 و 2.
وعليه تم استخدام الأقواس `{ }` ضمن
دائرة العداد i .
وبهذه الطريقة تم تمثيل 3 صفوف
بعمودين 2.

مثال

Write a program to print the following figure?

```

void main( ) {
for ( i = 1 ; i <= 4 ; i = i + 1 ) {
    for ( j = 1 ; j <= 5 ; j = j + 1 )
        cout << "*" ;
    cout << endl ;
}
}

```

```

* * * * *
* * * * *
* * * * *
* * * * *

```

مثال

Write a program to print the following figure?

```

void main( ) {
for ( i = 1 ; i <= 4 ; i = i + 1 ) {
    for ( j = 1 ; j <= i ; j = j + 1 )
        cout << "*" ;
    cout << endl ;
}
}

```

```

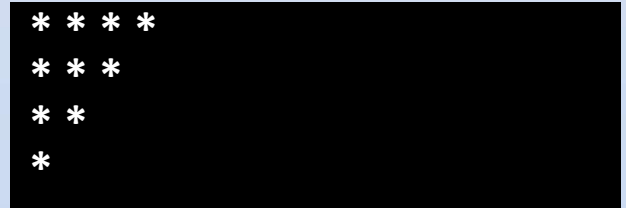
*
* *
* * *
* * * *

```

لاحظ ان العداد i اعتيادياً سيتكرر من 1 الى 4. بينما العداد j سيتكرر بعدد قيم العداد i .
أي سيتكرر من 1 الى i ويقوم بطباعة * في السطر الواحد بعدد قيم العداد i .
لاحظ وانتبه الى قيم العدادات باللون الأحمر والأزرق.

Write a program to print the following figure?

```
void main() {
for ( i = 1 ; i <= 4 ; i = i + 1 ) {
    for ( j = 1 ; j <= 5 - i ; j = j + 1 )
        cout << "*" ;
    cout << endl ;
}
}
```

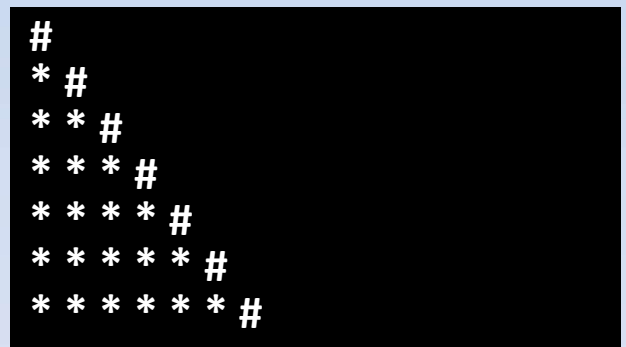


لاحظ ان العداد i سيتكرر من 1 الى 4 اعتيادياً. ولكن في كل مرة فإن العداد j سيبدأ اعتيادياً من 1 ولكنه ينتهي بقيم حسب المعادلة التالية والمشتقة من العلاقة التالية:

i	j يتكرر	لاحظ	العلاقة
1	4	$i + j = 5$	$j = 5 - i$
2	3	$i + j = 5$	$j = 5 - i$
3	2	$i + j = 5$	$j = 5 - i$
4	1	$i + j = 5$	$j = 5 - i$

Write a program to print the following figure?

```
void main() {
for ( i = 1 ; i <= 4 ; i = i + 1 ) {
    for ( j = 1 ; j <= i ; j = j + 1 )
        if ( i == j ) cout << "#" ;
        else cout << "*" ;
    cout << endl ;
}
}
```



لاحظ ان الرمز # يطبع فقط في حالة تساوي قيمة العداد i مع قيمة العداد j واللا سيطلع * أي عندما i و j يساويان 1. وعندما قيمة i و j يساويان 2. وعندما i و j يساويان 3 وهكذا عندما يساويان 4 وهكذا.

مثال

Write a program to print the times table (جدول الضرب) ?

```
void main( ) {  
for ( i = 1 ; i <= 10 ; i = i + 1 ) {  
    for ( j = 1 ; j <= 10 ; j = j + 1 )  
        cout << i * j << " " ;  
    cout << endl ;  
}  
}
```

```
1 2 3 4 5 6 7 8 9 10  
2 4 6 8 10 12 14 16 18 20  
3 6 9 12 15 18 21 24 27 30  
4 8 12 16 20 24 28 32 36 40  
5 10 15 20 25 30 35 40 45 50  
:  
:  
:
```

مثال

Write a program to read the marks of 8 students, each student has 5 marks. Then find and print the average of each student?

```
void main( ) {  
for ( i = 1 ; i <= 8 ; i = i + 1 ) {  
    s = 0 ;  
    for ( j = 1 ; j <= 5 ; j = j + 1 ) {  
        cin >> m ;  
        s = s + m ;  
    }  
    cout << s / 5.0 << endl ;  
}  
}
```

لاحظ أنه لكل طالب (ممثّل بالعداد i) تصفير
عداد الجمع s له، ومن ثم قراءة الـ 5
درجات له وجمعها في دارة j . ثم حساب
المعدل له خارج دارة j ولكن ضمن دارة i
لأنها تعتبر دارة الطالب نفسه.

مثال

Write a program to read 10 integers. Then find and print the factorial for each integer?

```
void main( ) {  
for ( i = 1 ; i <= 10 ; i = i + 1 ) {  
    cin >> a ;  
    f = 1 ;  
    for ( j = 1 ; j <= a ; j = j + 1 )  
        f = f * j ;  
    cout << f << endl ;  
}  
}
```

حيث أن مفكوك العدد a هو حاصل ضرب
الاعداد من 1 الى a . أي
 $a! = 1 * 2 * 3 * \dots * a$
فترة j

مثال

Find the number of iteration (عدد التكرارات) of these loops?

```
for ( i = 1 ; i <= 5 ; i = i + 1 ) {  
    for ( j = 1 ; j <= 7 ; j = j + 1 )  
}
```

لاحظ أن العداد i سيتكرر لـ 5 مرات.
وفي كل مرة سيتكرر العداد j لـ 7 مرات.
فسيكون عدد التكرارات الكلي هو:
 $35 = 7 * 5$ تكرار

تحسب تكرارات أي فترة ضمن أي دائرة for عن طريق:

القيمة العظمى (في شرط التوقف) - القيمة الأولية للعداد + 1

هذا في حالة ان القيمة العظمى فيها علامة يساوي واللا فستؤخذ القيمة العظمى - 1.

مثال

Find the number of iteration (عدد التكرارات) of these loops?

```
for ( i = 4 ; i <= 15 ; i = i + 1 ) {  
    for ( j = 6 ; j < 9 ; j = j + 1 )  
}
```

دورات i هي: $12 = 1 + 4 - 15$
دورات j هي: $3 = 1 + 6 - 8$
∴ عدد الدورات الكلية هي: $36 = 3 * 12$

مثال

Find the number of iteration (عدد التكرارات) of these loops?

```
for ( i = 4 ; i <= 20 ; i = i + 2 ) {  
    for ( j = 2 ; j <= 7 ; j = j + 1 )  
}
```

دورات i هي: $9 = 1 + (4 - 20) / 2$
دورات j هي: $6 = 1 + 2 - 7$
∴ عدد الدورات الكلية هي: $54 = 6 * 9$

مثال

Find the number of iteration (عدد التكرارات) of these loops?

```
for ( i = 12 ; i <= 33 ; i = i + 3 ) {  
    for ( j = 0 ; j <= 100 ; j = j + 5 )  
}
```

دورات i هي: $8 = 1 + (12 - 33) / 3$
دورات j هي: $21 = 1 + (0 - 100) / 5$
∴ عدد الدورات الكلية هي: $168 = 21 * 8$

مثال

Find the number of iteration (عدد التكرارات) of these loops?

```
for ( i = 6 ; i < 18 ; i = i + 1 ) {  
    for ( j = 2 ; j < 18 ; j = j + 2 )  
}
```

دورات i هي: $12 = 1 + 6 - 17$
دورات j هي: $8 = 1 + (2 - 16) / 2$
∴ عدد الدورات الكلية هي: $96 = 8 * 12$

Index Variables Arrays (Vectors)

المتغيرات الموسومة المصفوفات الاحادية (المتجهات)



يجب توفر 3 أشياء عند تخزين قيمة متغير
مثلاً (x) في ذاكرة الحاسبة:

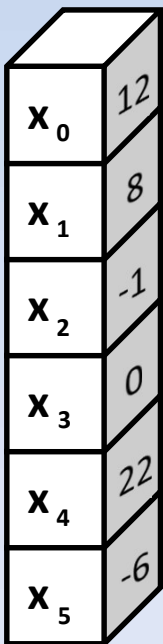
- 1- أسم المتغير x ،
- 2- قيمة المتغير 3 ،
- 3- عنوان الذاكرة لموقع التخزين $A593 BC90$

```
x = 0 ;
for( i = 1 ; i <= 3 ; i ++ )
    x = ( x + 0.5 ) * 2 ;
cout << x ;
```

i	x
	0
1	(0+0.5)*2 = 1
2	(1+0.5)*2 = 3
3	(3+0.5)*2 = 7

نلاحظ ان القيمة النهائية لـ x هي 7، وان قيم x القديمة مثل الـ (1 و 3) قد حذفوا من الذاكرة. فكيف لو ان المبرمج اذا ما احتاج الى هذه القيم المحذوفة في البرنامج فلن يستطيع الحصول عليها نهائياً...

من هنا جاء استخدام المصفوفات الاحادية (المتجهات) لتخزين قيم قديمة وقيم لاحقة في نفس المتغير (مثلاً x).



ملاحظة: في لغة C++ دائماً يبدأ الوسم بـ 0 ، وليس كالرياضيات حيث يبدأ بـ 1.

عليه (حسب المثال) سيتم حجز 6 خلايا في الذاكرة للمتغير x ابتداءً من الـ 0 الى الـ 5.

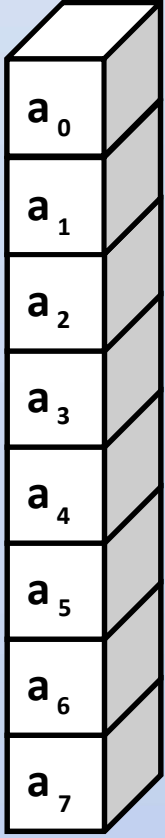
تعريف المتغير الموسوم (المصفوفة) وحجز مواقع له في الذاكرة:

```
Type Name [Size] ;
int x [ 6 ] ;
```



Reading the array

قراءة المصفوفة الاحادية (المتجه)



بما أن متغير المصفوفة (a كما في المثال التالي) يحتوي على أكثر من قيمة، وإن كل قيمة لها **وسم** خاص بها، فيجدر بنا استخدام **جملة for** للمرور على جميع مواقع عناصر المصفوفة a.

للتعريف مثلاً :
 float a [8] ;
 للقراءة :
 for(i=0 ; i < 8 ; i ++)
 cin >> a [i] ;

نلاحظ ان الوسم يبدأ بـ **0** وينتهي **بعدد صحيح**، وبزيادة **1**، وإن هذا الوسم يجب ان يكون من نوع **integer**.

أذن

لقراءة مصفوفة يجب استخدام **for**

```
for( i=0 ; i < size ; i ++ )
    cin >> name [ i ] ;
```

لطباعة مصفوفة يجب استخدام **for**

```
for( i=0 ; i < size ; i ++ )
    cout << name [ i ] ;
```

للمعالجة على المصفوفة يجب استخدام **for** للمرور على عناصر المصفوفة واستخدامها للمعالجة.

ملاحظة: تم استخدام كل من

عنصر المصفوفة

a [i]



a[i] **قراءة**

a[i] **طباعة**

a[i] **المعالجة على**

وسم المصفوفة

i



ضمن الدارة **for**

اسم المصفوفة

a



مثال

Write a program to read 10 marks. Then find and print the average?

بدون استخدام المصفوفات:

```
int x , s , i ;
float av ;
void main( ) {
    s = 0 ;
    for( i=1 ; i <= 10 ; i ++ )
        cin >> x ;
        s = s + x ;
    }
    av = s / 10.0 ;
    cout << av ;
}
```

باستخدام المصفوفات:

```
int x [10] , s , i ;
float av ;
void main( ) {
    for( i=0 ; i < 10 ; i ++ )
        cin >> x [i] ;
    s = 0 ;
    for( i=0 ; i < 10 ; i ++ )
        s = s + x [i] ;
    av = s / 10.0 ;
    cout << av ;
}
```

قراءة	■
عدادات	■
معالجة	■
نتائج	■

مثال

Write a program to read 15 integers. Then find:

- Sum of the positives?
- Sum of integers that divisible by 5?

بدون استخدام المصفوفات:

```
int num , s1 , s2 , i ;
void main( ) {
    s1 = 0 ;
    s2 = 0 ;
    for( i=1 ; i <= 15 ; i ++ ) {
        cin >> num ;
        if( num > 0 ) s1 = s1 + num ;
        if( num % 5 == 0 ) s2 = s2 + num ;
    }
    cout << s1 << s2 ;
}
```

باستخدام المصفوفات:

```
int num [15] , s1 , s2 , i ;
void main( ) {
    for( i=0 ; i < 15 ; i ++ )
        cin >> num [i] ;
    s1 = 0 ;
    s2 = 0 ;
    for( i=0 ; i < 15 ; i ++ ) {
        if( num[i] > 0 ) s1 = s1 + num[i] ;
        if( num[i] % 5 == 0 ) s2 = s2 + num[i] ;
    }
    cout << s1 << s2 ;
}
```



مثال

Write a program to read 8 numbers. Then find the maximum (biggest) number and the minimum (smallest) number?

نحن نعلم ان مثل هذه الأسئلة التي فيها مطلب (أكبر/أصغر/أقصر/أطول/أعلى) التي تكون على وزن (أفعل). فسوف نستخدم طريقة الفرضية، والتي تنص على: **فرض أول عدد هو الأكبر/الأصغر/... ، ثم مقارنة باقي الأعداد مع الفرضية.**

بدون استخدام المصفوفات:

```
int a , big , small , i ;
void main( ) {
    cin >> a ;
    big = a ;
    small = a ;
    for( i=2 ; i <= 8 ; i ++ ) {
        cin >> a ;
        if(a > big) big = a ;
        if(a < small) small = a ;
    }
    cout << big << small ;
}
```

قراءة اول عدد a لغرض فرضه هو الأكبر/الأصغر/...
فرض العدد الأول هو الكبير
فرض العدد الأول هو الصغير
لغرض مقارنة باقي الاعداد
قراءة باقي الاعداد ... عدد بعد عدد ضمن الدارة for
ثم مقارنة العدد المقروء الجديد مع كل من فرضية الكبير مرة والصغير مرة وتبديل الفرضية اذا كان العدد الجديد أكبر من الكبير أو اصغر من الصغير.

باستخدام المصفوفات:

```
int a [8] , big , small , i ;
void main( ) {
    for( i=0 ; i < 8 ; i ++ )
        cin >> a [i] ;
    big = a[0] ;
    small = a[0] ;
    for( i=1 ; i < 8 ; i ++ ) {
        if(a[i] > big) big = a[i] ;
        if(a[i] < small) small = a[i] ;
    }
    cout << big << small ;
}
```

قراءة عناصر المصفوفة أولاً
فرض العنصر الأول هو الكبير
فرض العنصر الأول هو الصغير
ثم مقارنة باقي العناصر مع كل من فرضية الكبير مرة والصغير مرة وتبديل الفرضية اذا كان العنصر الحالي أكبر من الكبير أو اصغر من الصغير.

Write a program to read x[10]. Then find:

- Sum of all even elements?
- Sum of all odd elements?
- Sum of all elements in the even order/index (ترتيب/موقع زوجي)?
- Sum of all elements in the odd order/index?

```
int x[10] ;  
  
int s1 , s2 , s3 , s4 , i ;  
  
void main( ) {  
    for( i=0 ; i < 10 ; i ++ )  
        cin >> x[i] ;  
  
    s1 = s2 = s3 = s4 = 0 ;  
  
    for( i=0 ; i < 10 ; i ++ ) {  
  
        if( x[i] % 2 == 0 ) s1 = s1 + x[i] ;  
  
        else s2 = s2 + x[i] ;  
  
        if( i % 2 == 0 ) s3 = s3 + x[i] ;  
  
        else s4 = s4 + x[i] ;  
  
    }  
  
    cout << s1 << s2 << s3 << s4 ;  
  
}
```



أمثلة على المصفوفات الأحادية (المتجهات)

مثال

Write a program to read x[8]. Then multiply all even elements by 3?

```
int x[8], i;
void main( ) {
for( i = 0 ; i < 8 ; i ++ )
cin >> x[i] ;
for( i = 0 ; i < 8 ; i ++ )
if(x[i] % 2 == 0) x[i] = x[i] * 3 ;
for( i = 0 ; i < 8 ; i ++ )
cout << x[i] ;
}
```

قراءة	■
عدادات	■
معالجة	■
نتائج	■

مثال

Write a program to read a[10] numbers. Then

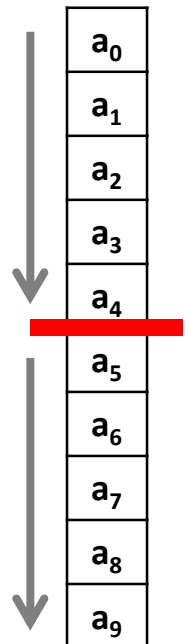
* Reset (صفر) the first half of a?

* Set (اجعل/ضع) the second half to -1 ?

```
int a[10], i;
void main( ) {
for( i = 0 ; i < 10 ; i ++ )
cin >> a[i] ;
for( i = 0 ; i < 5 ; i ++ )
a[i] = 0 ;
for( i = 5 ; i < 10 ; i ++ )
a[i] = -1 ;
for( i = 0 ; i < 10 ; i ++ )
cout << a[i] ;
}
```

ملاحظة: في هكذا نوع من الأسئلة والتي فيها المعالجة تكون حسب مواقع (وسم) المتجه.... أرسم المصفوفة

$$10 / 2 = 5$$



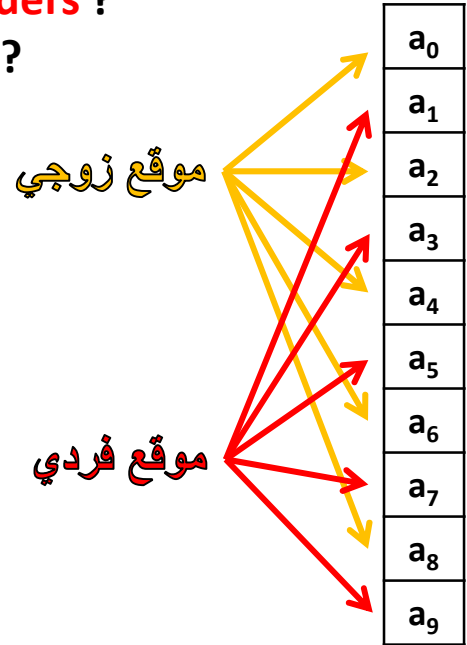
مثال

Write a program to read a[10]. Then

* Find the sum of all elements in the **odd orders** ?

* Set all elements in the **even orders** to 999 ?

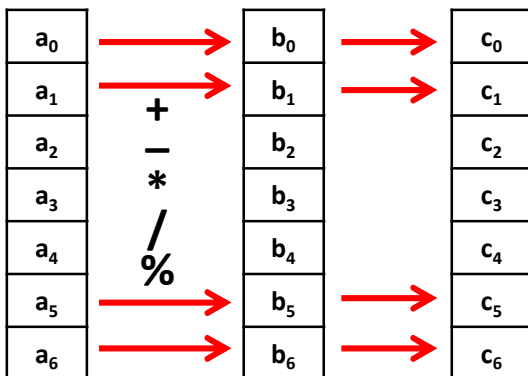
```
int a[10], i, s;
void main( ) {
for( i = 0; i < 10; i++ )
cin >> a[i];
s = 0;
for( i = 1; i <= 9; i = i + 2 )
s = s + a[i];
for( i = 0; i <= 8; i = i + 2 )
a[i] = 999;
cout << s << endl;
for( i = 0; i < 10; i++ )
cout << a[i];
}
```



ملاحظة:

يتم طباعة المصفوفة في حالة اذا تم تغيير قيمها اثناء المعالجة..
بمعنى متى ما تم ذكر عنصر المصفوفة (x[i] أو a[i] أو z[i] أو اي مسمى) وتم ذكره
على جهة اليسار في أي معادلة، معناها أنه قد تم تغيير والتلاعب بقيم المصفوفة.. وعليه
يجب طباعة المصفوفة كاملة في نهاية البرنامج.

العمليات على المصفوفات الأحادية (المتجهات)



ملاحظة:

العمليات على المصفوفات الاحادية أو المتجهات
يجب ان تتساوى احجام المتجهات

$$a_n \begin{matrix} + \\ - \\ * \\ / \end{matrix} b_n = c_n$$

Write a program to find:

$$Y = A_7 + \underbrace{B * F}_M - C$$

```
int a[7], b[7], c[7], f[7], m[7], y[7], i;
void main( ) {
for( i = 0 ; i < 7 ; i ++ )
cin >> a[i];
for( i = 0 ; i < 7 ; i ++ )
cin >> b[i];
for( i = 0 ; i < 7 ; i ++ )
cin >> f[i];
for( i = 0 ; i < 7 ; i ++ )
cin >> c[i];
for( i = 0 ; i < 7 ; i ++ )
m[i] = b[i] * f[i];
for( i = 0 ; i < 7 ; i ++ )
y[i] = a[i] + m[i] - c[i];
for( i = 0 ; i < 7 ; i ++ )
cout << y[i];
}
```

حسب قاعدة الاسبقية في العمليات
فان عملية الضرب والقسمة وباقي
القسمة يتم ايجادها قبل عمليات
الجمع والطرح.

أو ممكن اجراء المعالجة كاملة
والحاسبة سوف تقوم بأجراء
عمليات الاسبقية تلقائياً

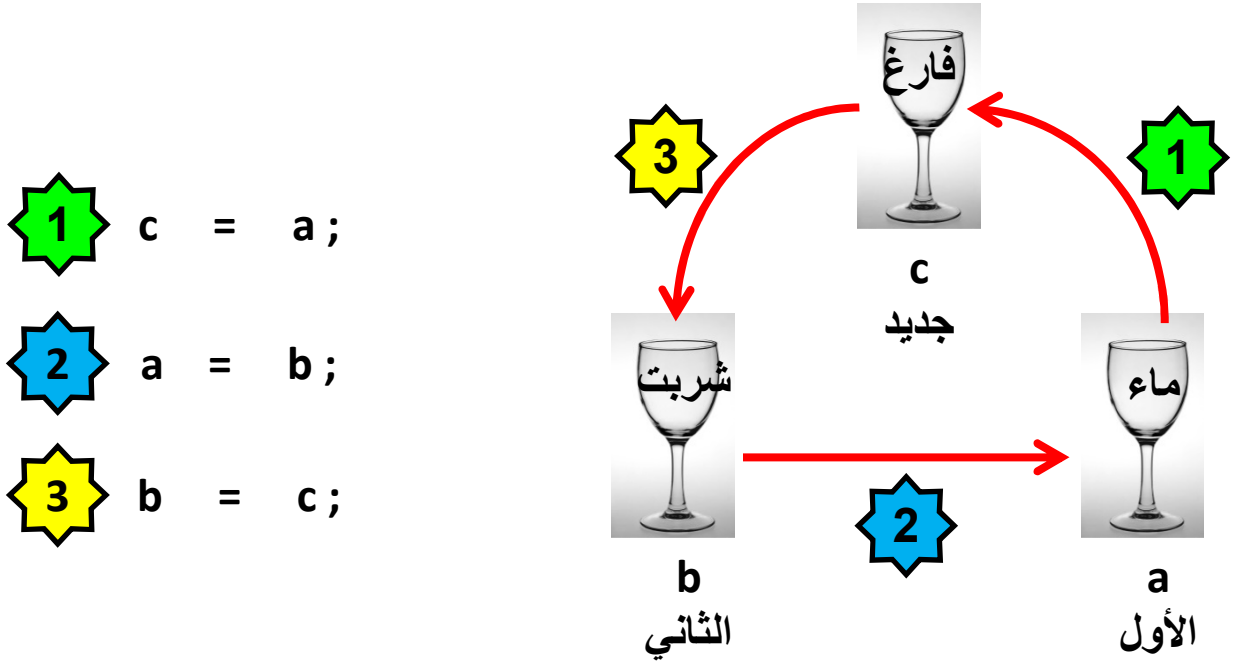
```
for( i = 0 ; i < 7 , i ++ )
m[i] = b[i] * f[i];
```

```
for( i = 0 ; i < 7 ; i ++ )
y[i] = a[i] + b[i] * f[i] - c[i];
```

Swapping elements

التبديل ما بين عناصر المصفوفة

في بعض الأسئلة قد نحتاج (أو مطلوب من ضمن السؤال) التبديل ما بين عناصر المصفوفة. وهذا يذكرنا بطريقة تبديل كلاص الماء بكلاص الشربت.. Swap



قراءة القاعدة من اليسار الى اليمين

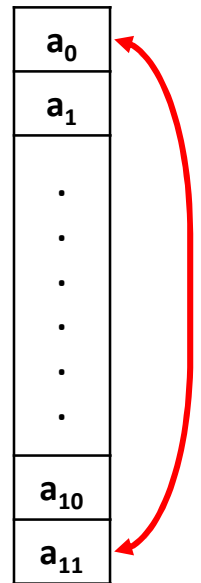
الأول = جديد
الثاني = الأول
جديد = الثاني

مثال

Write a program to read a[12]. Then swap (exchange) the **first** element with the **last** element?

```
int a[12], i, temp ;
void main( ) {
for( i = 0 ; i < 12 ; i ++ )
cin >> a[i] ;
temp = a[0] ;
a[0] = a[11] ;
a[11] = temp ;
for( i = 0 ; i < 12 ; i ++ )
cout << a[i] ;
}
```

$a_0 \longleftrightarrow a_{11}$
الأول الثاني



مثال

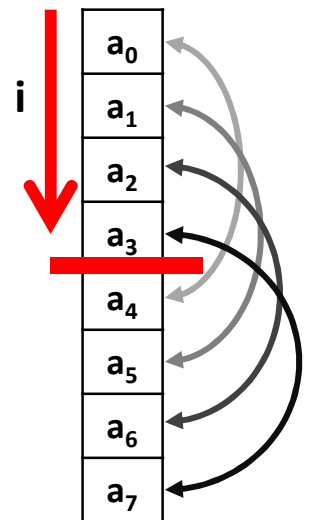
Write a program to read a[8]. Then swap (exchange) the **first half** with the **second half** of it?

```
int a[8], i, temp ;
void main( ) {
for( i = 0 ; i < 8 ; i ++ )
cin >> a[i] ;

for( i = 0 ; i <= 3 ; i ++ ) {
temp = a[i] ;
a[i] = a[i + 4] ;
a[i + 4] = temp ;
}

for( i = 0 ; i < 8 ; i ++ )
cout << a[i] ;
}
```

$a_i \longleftrightarrow a_{i+4}$
الأول الثاني



النصف الأول	$i + \frac{\text{size}}{2}$	النصف الثاني
0	$0 + 4 =$	4
1	$1 + 4 =$	5
2	$2 + 4 =$	6
3	$3 + 4 =$	7

مثال

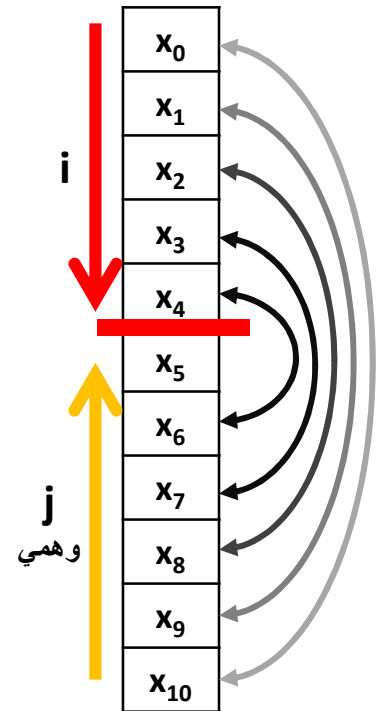
Write a program to read $x[11]$. Then find the transpose of it (x^T) ?

ملاحظة

كل عدد يتزايد (i) ... يقابله عدد يتناقص

فالذي يتناقص هو ($n - 1 - i$)

$x_i \longleftrightarrow x_{n-1-i}$
الأول الثاني



```
int x[11], i, temp ;  
void main( ) {
```

```
for( i = 0 ; i < 11 ; i ++ )  
cin >> x[i] ;
```

```
for( i = 0 ; i < 5 ; i ++ ) {  
temp = x[i] ;  
x[i] = x[10 - i] ;  
x[10 - i] = temp ;  
}
```

```
for( i = 0 ; i < 11 ; i ++ )  
cout << x[i] ;  
}
```

i	j	علاقة i مع j
0	10	$0 + 10 = 10$
1	9	$1 + 9 = 10$
2	8	$2 + 8 = 10$
3	7	$3 + 7 = 10$
4	6	$4 + 6 = 10$

$$i + j = 10$$

$$i + j = 11 - 1$$

$$j = 11 - 1 - i$$

بصورة عامة اذا
حجم المصفوفة = n

$$i + j = n - 1$$

$$j = n - 1 - i$$

Q1) Find the average of the second half of **a[12]**?

ايجاد معدل العناصر للنصف الثاني من **a[12]**؟

Q2) Reset all even elements in the vector **c[10]**?

صفر (اجعلها تساوي صفر) العناصر الزوجية في المتجه **c[10]**؟

Q3) Read **x[10]** and an integer value (**m**). Then find the repetition of **m** in **x**?

قراءة **x[10]** وقيمة صحيحة واحدة مثل (**m**). ثم جد عدد تكرارات العدد **m** في المتجه **x**؟

Q4) Write a program to read **x[10]**. Then find the largest element in the first half and the largest element in the second half. Then exchange their values?

برنامج لقراءة **x[10]**. ثم جد أكبر عنصر في النصف الاول وأكبر عنصر في النصف الثاني من المصفوفة. ثم بدل قيمهم؟

Q5) Write a program to find the result for the following:

$$A = X^T + Y_6 - Z$$

برنامج لاجاد ناتج التعبير؟

Q6) Write a program to exchange the elements in the even orders in the first half of **a[12]** by the elements in the odd orders in the second half?

برنامج لتبديل العناصر ذات التسلسل الزوجي في النصف الأول من المصفوفة **a[12]** مع العناصر ذات التسلسل الفردي في النصف الثاني منها؟

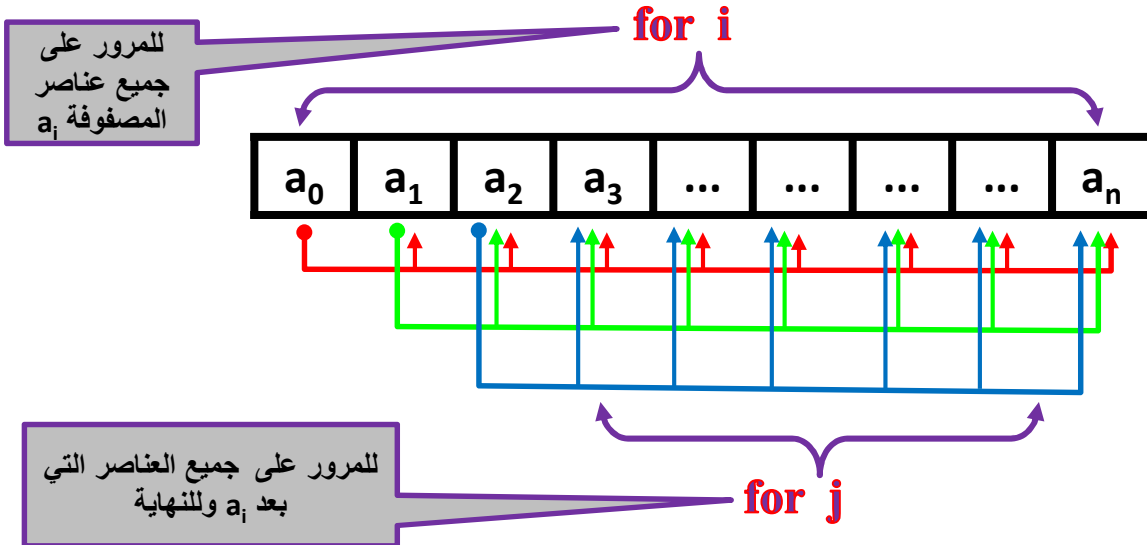
ترتيب عناصر المصفوفة الأحادية

Re-arranging (Sorting)

قد نحتاج الى ترتيب المتجهات **تصاعدياً (Ascending)** أو **تنازلياً (Descending)** حسب متطلبات السؤال... مثلاً ترتيب قائمة الدرجات أو ترتيب قائمة الأسماء أو غيرها. أبسط وأسهل خوارزمية للترتيب هو **الترتيب بالاختيار (Bubble Sort)**.. حيث تعتبر فكرتها سهلة وبرمجتها ايضاً سهلة... وتنص على:

مقارنة كل عنصر من المصفوفة مع جميع العناصر التي تليه وللنهاية.

يعني مقارنة :

$$\begin{aligned} a_0 &\leftrightarrow a_1, a_2, a_3, \dots, a_n \\ a_1 &\leftrightarrow a_2, a_3, \dots, a_n \\ a_2 &\leftrightarrow a_3, a_4, \dots, a_n \\ &\vdots \end{aligned}$$


∴ هذا النوع من الافكار على المصفوفة الاحادية يحتاج الى **for** اثنين..

أي أن i يقارن من 0 الى ما قبل النهاية $n-1$ (لان لو i أخذ الـ n فلا يوجد عنصر بعد الـ n)
وأن j يقارن ما بعد i من $(i+1)$ والى النهاية n

```
for i = 0 → n - 1
for j = i + 1 → n
if ( ai > aj ) { ابدل ai ← aj }
```

قاعدة

مثال : رتب الأعداد التالية لـ $a[5]$

0 1 2 3 4
3, 8, 4, 9, 1

$i = 0, j = 1$ 3, 8, 4, 9, 1 \longrightarrow 3, 8, 4, 9, 1

$\underbrace{\hspace{1.5cm}}_x$

, $j = 2$ 3, 8, 4, 9, 1 \longrightarrow 3, 8, 4, 9, 1

$\underbrace{\hspace{1.5cm}}_x$

, $j = 3$ 3, 8, 4, 9, 1 \longrightarrow 3, 8, 4, 9, 1

$\underbrace{\hspace{1.5cm}}_x$

, $j = 4$ 3, 8, 4, 9, 1 \longrightarrow 1, 8, 4, 9, 3

$\underbrace{\hspace{1.5cm}}_x$

$i = 1, j = 2$ 1, 8, 4, 9, 3 \longrightarrow 1, 4, 8, 9, 3

$\underbrace{\hspace{1.5cm}}_{\checkmark}$

, $j = 3$ 1, 4, 8, 9, 3 \longrightarrow 1, 4, 8, 9, 3

$\underbrace{\hspace{1.5cm}}_x$

, $j = 4$ 1, 4, 8, 9, 3 \longrightarrow 1, 3, 8, 9, 4

$\underbrace{\hspace{1.5cm}}_x$

$i = 2, j = 3$ 1, 3, 8, 9, 4 \longrightarrow 1, 3, 8, 9, 4

$\underbrace{\hspace{1.5cm}}_x$

, $j = 4$ 1, 3, 8, 9, 4 \longrightarrow 1, 3, 4, 9, 8

$\underbrace{\hspace{1.5cm}}_{\checkmark}$

$i = 3, j = 4$ 1, 3, 4, 9, 8 \longrightarrow 1, 3, 4, 8, 9

$\underbrace{\hspace{1.5cm}}_{\checkmark}$

```
for i = 0  $\longrightarrow$  < 4 (n - 1)
  for j = i + 1  $\longrightarrow$  < 5 n
```

```
for( i = 0 ; i < 4 ; i ++ )
  for( j = i + 1 ; j < 5 ; j ++ )
    if( a[i] > a[j] ) { temp = a[i] ;
                        a[i] = a[j] ;
                        a[j] = temp ; }
```

Write a program to re-arrange 14 marks ascendingly?

```
int mark[14], i, j, t;
void main() {
for( i = 0 ; i < 14 ; i ++ )
    cin >> mark[i] ;

for( i = 0 ; i < 13 ; i ++ )
    for( j = i + 1 ; j < 14 ; j ++ )
        if( mark[i] > mark[j] ) { t = mark[i] ;
                                mark[i] = mark[j] ;
                                mark[j] = t ; }

for( i = 0 ; i < 14 ; i ++ )
    cout << mark[i] ;
}
```

Write a program to find the **THREE** maximum/largest numbers among 20 numbers?

```
int a[20], i, j, t;
void main() {
for( i = 0 ; i < 20 ; i ++ )
    cin >> a[i] ;

for( i = 0 ; i < 19 ; i ++ )
    for( j = i + 1 ; j < 20 ; j ++ )
        if( a[i] > a[j] ) { t = a[i] ;
                            a[i] = a[j] ;
                            a[j] = t ; }

for( i = 19 ; i >= 17 ; i -- )
    cout << a[i] ;
}
```

إما ترتيب تصاعدي.. فالأعداد الكبيرة ستكون بالأخير.. عندها نطبع آخر 3 عناصر..
أو ترتيب تنازلي.. فالأعداد الكبيرة ستكون في البداية.. عندها نطبع أول 3 عناصر..

```
for( i = 17 ; i <= 19 ; i ++ )
    cout << a[i] ;
}
```

مثال

Write a program to read 8 characters. Then print them in **alphabit** order?

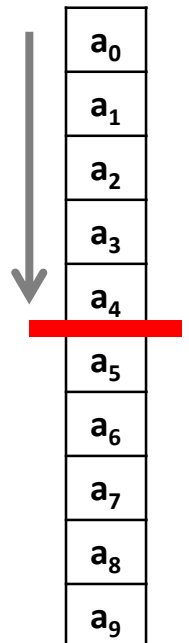
```
char ch[8] , temp ;
int i , j ;
void main() {
for( i = 0 ; i < 8 ; i ++ )
    cin >> ch[i] ;
for( i = 0 ; i < 7 ; i ++ )
    for( j = i + 1 ; j < 8 ; j ++ )
        if( ch[i] > ch[j] ) { temp = ch[i] ;
                                ch[i] = ch[j] ;
                                ch[j] = temp ; }
for( i = 0 ; i < 8 ; i ++ )
    cout << ch[i] ;
}
```

مثال

Write a program to read x[10]. Then re-arrange the **first half** ascendingly?

```
int x[10] , i , j , temp ;
void main() {
for( i = 0 ; i < 10 ; i ++ )
    cin >> x[i] ;
for( i = 0 ; i < 4 ; i ++ )
    for( j = i + 1 ; j < 5 ; j ++ )
        if( x[i] > x[j] ) { temp = x[i] ;
                            x[i] = x[j] ;
                            x[j] = temp ; }
for( i = 0 ; i < 10 ; i ++ )
    cout << x[i] ;
}
```

$$10 / 2 = 5$$



إنشاء مصفوفة جزئية من مصفوفة أخرى

قد نحتاج أحياناً الى تكوين مصفوفة ثانوية عناصرها مأخوذة أو منتقاة من مصفوفة أخرى. هذه الحالة نطلق عليها إنشاء **مصفوفة جزئية** من **مصفوفة معلومة**.

Create an array (sub-array/sub-vector) from another array.

حجم المصفوفة **a**
معلوم هو 10.

```
int a[10], i;
```

```
int b[10], m;
```

```
void main() {  
for( i = 0 ; i < 10 ; i ++ )  
cin >> a[i];
```

حجم المصفوفة **b**
هو غير معلوم لحد
الآن. فهو **m**.

القيمة 0 تعتبر وسم للعنصر
الأول... عليه لا يمكن ان
يبدأ وسم المصفوفة **b** ب 0.

المصفوفة
المعلومة **a**

a ₀
a ₁
a ₂
a ₃
a ₄
a ₅
a ₆
a ₇
a ₈
a ₉

المصفوفة
الجزئية **b**

b ₀
b ₁
b ₂
⋮
⋮
⋮

```
m = -1;
```

هذه دائرة المصفوفة **a** المعلومة/المقروءة

```
for( i = 0 ; i < 10 ; i ++ )
```

```
if ( a[i] ??? تكوينها المراد ) { m ++;
```

```
b[m] = a[i];
```

```
}
```

```
for( i = 0 ; i <= m ; i ++ )
```

```
cout << b[i];
```

```
}
```

طباعة المصفوفة الجديدة والمكونة **b**

Write a program to read **x[10]**. Then create the vector **y** that **مثال**
consists of the even elements only from **x** ?

```
int x[10], i ;
int y[10], m ;
void main() {
for( i = 0 ; i < 10 ; i ++ )
    cin >> x[i] ;
m = -1 ;
for( i = 0 ; i < 10 ; i ++ )
    if (x[i] % 2 == 0) { m ++ ;
                        y[m] = x[i] ; }
for( i = 0 ; i <= m ; i ++ )
    cout << y[i] ;
}
```

مثال

Read a[20]. Then CREATE two vectors, first one consists of
positives of a, and the second one consists of the negatives of a ?

```
int a[20], i, pos[20], m, neg[20], k ;
void main() {
for( i = 0 ; i < 20 ; i ++ )
    cin >> a[i] ;
m = -1 ;
k = -1 ;
for( i = 0 ; i < 20 ; i ++ )
    if (a[i] >= 0) { m ++ ;
                    pos[m] = a[i] ; }
    else { k ++ ;
          neg[k] = a[i] ; }
for( i = 0 ; i <= m ; i ++ )
    cout << pos[i] ;
for( i = 0 ; i <= k ; i ++ )
    cout << neg[i] ;
}
```

Read `a[15]` and `b[8]`. Then CREATE a vector (`c`). Then store the even elements of `a` in the beginning of `c`, and store the odd elements of `b` at the end of `c` ?

```
int a[15], b[8], i, c[23], m ;
void main() {
for( i = 0 ; i < 15 ; i ++ )
    cin >> a[i] ;
for( i = 0 ; i < 8 ; i ++ )
    cin >> b[i] ;

m = -1 ;

for( i = 0 ; i < 15 ; i ++ )
    if ( a[i] % 2 == 0 ) { m ++ ;
                        c[m] = a[i] ; }
for( i = 0 ; i < 8 ; i ++ )
    if ( b[i] % 2 != 0 ) { m ++ ;
                        c[m] = b[i] ; }

for( i = 0 ; i <= m ; i ++ )
    cout << c[i] ;
}
```

Lab Exercises

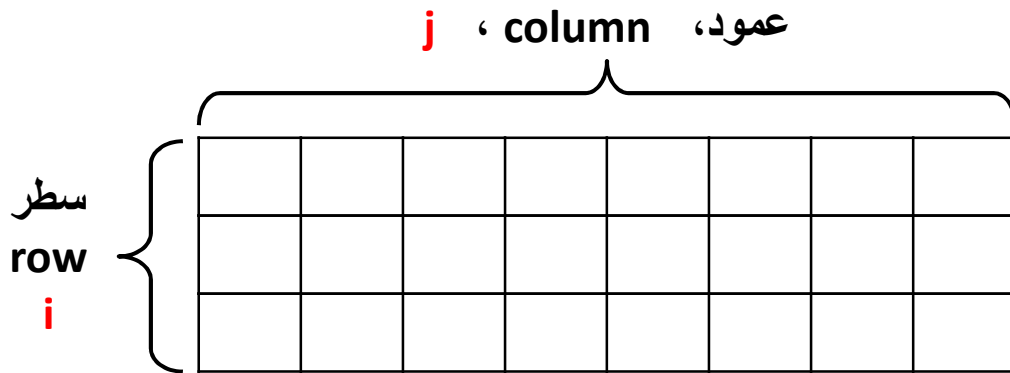
تمارين

Q1) Read `a[14]`. Then CREATE (`b`) that consists at the beginning the elements that divisible by 3 from the second half of `a`. And then consists the elements that divisible by 5 in the first half of `a`?

قراءة المصفوفة `a[14]`. ثم كَوّن المصفوفة `b` والتي تحتوي في بدايتها العناصر التي تقبل القسمة على 3 في النصف الثاني من المصفوفة المقروءة `a`. ومن ثم تحتوي على العناصر التي تقبل القسمة على 5 في النصف الاول من المصفوفة `a` ؟

Two Dimensional Arrays

المصفوفات الثنائية



j

	العمود 0	العمود 1	العمود 2	العمود 3	العمود 4	العمود 5	العمود 6	العمود 7
السطر 0	a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}	a_{06}	a_{07}
السطر 1	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}
السطر 2	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}

i

المصفوفة المربعة

هي المصفوفة التي فيها
عدد الأسطر = عدد الأعمدة

2×2

a_{00}	a_{01}
a_{10}	a_{11}

3×3

a_{00}	a_{01}	a_{02}
a_{10}	a_{11}	a_{12}
a_{20}	a_{21}	a_{22}

4×4

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

تعريف المصفوفة الثنائية

```
type Array_Name [row][column];  
int a[3][8];  
float x[6][6];
```

قراءة المصفوفة الثنائية

A 3x8 grid representing a 2D array. The rows are indexed by 'i' (0, 1, 2) and the columns by 'j' (0, 1, 2, 3, 4, 5, 6, 7). The elements are labeled a_{ij} from a₀₀ to a₂₇.

	a ₀₀	a ₀₁	a ₀₂	a ₀₃	a ₀₄	a ₀₅	a ₀₆	a ₀₇
	a ₁₀	a ₁₁	a ₁₂	a ₁₃	a ₁₄	a ₁₅	a ₁₆	a ₁₇
	a ₂₀	a ₂₁	a ₂₂	a ₂₃	a ₂₄	a ₂₅	a ₂₆	a ₂₇

```
for( i = 0 ; i < 3 ; i ++ )  
  for( j = 0 ; j < 8 ; j ++ )  
    cin >> a[i][j] ;
```

طباعة المصفوفة الثنائية

```
for( i = 0 ; i < 3 ; i ++ ) {  
  for( j = 0 ; j < 8 ; j ++ )  
    cout << a[i][j] ;  
  cout << endl ;  
}
```

ملاحظة:

طباعة النتائج الاعتيادية مثل (مجموع العناصر، عدد العناصر) لا يحتاج الى طباعة المصفوفة لأن قيم المصفوفة تم استدعائها فقط وجمعها أو عدّها.

ملاحظة:

عند تغيير قيم أي عنصر من عناصر المصفوفة، عندها يتم طباعة المصفوفة كاملة.

Read `a[3][4]`. Then find:

- Sum of all positive elements ?
- Sum of all even elements ?
- No. of elements that divisible by 5 ?

```
int a[3][4], i, j ;
```

```
int s1, s2, k ;
```

```
void main() {
```

```
for( i = 0 ; i < 3 ; i ++ )
```

```
for( j = 0 ; j < 4 ; j ++ )
```

```
cin >> a[i][j] ;
```

```
s1 = 0 ;
```

```
s2 = 0 ;
```

```
k = 0 ;
```

```
for( i = 0 ; i < 3 ; i ++ )
```

```
for( j = 0 ; j < 4 ; j ++ ) {
```

```
if( a[i][j] > 0 ) s1 = s1 + a[i][j] ;
```

```
if( a[i][j] % 2 == 0 ) s2 = s2 + a[i][j] ;
```

```
if( a[i][j] % 5 == 0 ) k = k + 1 ;
```

```
}
```

```
cout << s1 << s2 << k ;
```

```
}
```