ملخصات

لجنة سنافر البوليتكنك - الاتجاه الإسلامي

اسم المادة

C++

الطبعة الثانية

شرح مع أسئلة سابقة

تواصل معنا

www.Muslimengineer.info

Muslimengineerpage

/groups/Smurfs.On.The.Way

مقدمة

بسم الله الرحمن الرحيم

نضع بين أيديكم هذا الجهد المتواضع في شرح مادة
مهارات حاسوب 2 (++c)
الطبعة الثانية المنقحة من الأخطاء

حيث يمكن الإعتماد على هذا التلخيص في فهم المادة
ويبقى عليك حل المزيد من التمرينات
حيث تم وضع تمرينات اضافية: اسئلة سنوات سابقة
مع اسئلة مقترحة لعدد من مدرسي المادة
في نهاية التلخيص

نتمنى لكم التوفيق والنجاح في دراستكم
راجعين رضا الله سبحانه وتعالى

لا تنسونا من صالح دعائكم

C++:

- **High level programming language.**
- **Compiler: convert form (c++) to (incline language)**

First Example:

هذا هو أول مثال سوف نتعلمه وسنقوم بشرح المبادئ الأساسية للبرنامج من خلاله:

```
1. // a first program in c++
2. # include <iostream.h>
3. int main ()
4. {
5. cout <<"welcome to c++";
6. return 0;
7. }
```

أوامر البرنامج:

1. // : comment (تعليق):

- أمر التعليق (//) لا يظهر في المخرجات فهو فقط لبيان وظيفة البرنامج أو لإعطاء أي ملاحظة للمبرمج باستخدام البرنامج.
- أمر التعليق غير ضروري.
- يمكن كتابة أمر التعليق في أي مكان في البرنامج وأكثر من مرة.

2. # include < iostream.h>

عبارة عن أمر لاستدعاء مكتبة تحتوي على اقترانات الإدخال والإخراج

cin >> : اقتران إدخال

cout<< : اقتران إخراج (للطباعة)

- ملاحظة: إذا كتبنا في البرنامج أمر (cin) أو (cout) بدون استدعاء المكتبة السابقة يعطي خطأ.

3. int main():

- تمثل بداية البرنامج، يجب أن يحتوي البرنامج على أمر (return 0) في نهايته عند استخدامها.
  أي أن [int main()] و [return 0] يجب أن يكونوا متلازمتان.
- يمكن استخدام [void main()] كبداية للبرنامج وهي لا تحتاج إلى نهاية.

www.muslimengineer.info
facebook.com/muslim.engineer.page

4. { : left brace (main) تمثل بداية حدود الـ

} : right brace (main) تمثل نهاية حدود الـ

- تعرفنا سابقا أن [int main()] أو [void main()] تمثل بداية البرنامج، ولكن.. هذا البرنامج يجب أن يكون داخل الحدود وهي { }.
- أي أمر يقع خارج { } لا ينفذ.
- كل جملة (أمر) يكتب داخل الـ main يجب أن تنتهي بالفاصلة المنقوطة (;).

5. return 0;

- تمثل نهاية البرنامج (لا ينفذ أي شيء بعدها)

أمر الطباعة output (cout):

- نستخدم أمر (cout) لطباعة مخرجات في البرنامج.
- يجب أن يتبع (cout) إشارة (<<) لتصبح : cout<<
- لطباعة جمل أو كلمات يجب أن توضع بين علامات اقتباس " ".

Ex 1:

```
cout <<" welcome to c++";
```

هنا يقوم البرنامج بطباعة welcome to c++ كما هي في البرنامج.

Ex2:

```
cout << 5+3;
```

هنا يقوم البرنامج بإيجاد حاصل جمع (5) مع (3) ويطبع الناتج (8).

ملاحظات:

- جميع الأوامر في برمجية C++هي أوامر ضرورية ما عدا أمر التعليق (//).
- جميع الحروف المستخدمة في كتابة البرنامج هي حروف صغيرة (small letter).

facebook.com/Smurfs.On.The.Way

في البرنامج التالي:

Ex:

```
#include <iostream.h>
void  main()
{
cout << " muslim engineer";
}
cout << " Hi";
```

- هذا البرنامج يعطي error

وذلك بسبب وجود امر الطباعة بعد نهاية البرنامج

Ex:

```
#include <iostream.h>
int main()
{
cout << " muslim engineer";
return 0;
cout << 5;
}
```

- هنا البرنامج يطبع فقط muslim engineer

ولا يطبع الرقم 5 لأنه موجود بعد return

تمثل نهاية البرنامج

- إذا أعطى البرنامج أي خطأ error فإن البرنامج لا يعطي مخرجات.

- **بعض الأوامر الموجودة في أمر الطباعة:**

- **\n** : سطر جديد

Ex:

```
cout << " Muslim\nEngineer";
```

لا يشترط وجود مسافة بينها وبين الجملة المحيطة بها

شاشة المخرجات

```
Muslim
Engineer
```

- **** : لطباعة (\)

Ex:

```
cout << " Muslim Engineer\\";
```

```
Muslim Engineer\
```

- **\"** : لطباعة (")

Ex:

```
cout << " Muslim Engineer\"";
```

```
Muslim Engineer"
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

- \t : يعطي مسافة

يزيح الجملة إلى بداية خانة جديدة.. حيث تقسم شاشة المخرجات إلى خانات كل خانة ثمانية حروف:

Ex:

```
cout << "Muslim\tEngineer";
cout << "1234567\tEngineer";
cout << "1\tEngineer";
cout << "\t\tHi";
```

(على اعتبار ان كل جملة في برنامج منفصل)

```
Muslim  Engineer   → 2
1234567 Engineer   → 1
1       Engineer   → 7
                Hi → 16
```

عدد الفراغات بين الكلمتين

- \a : يصدر صوت beep

- \r : ترجع المؤشر لبداية السطر (تمحي ما قبلها من جمل)

Ex:

```
cout << "Hello\r";
cout << "MuslimEngineer";
```

```
MuslimEngineer
```

- جميع الأوامر السابقة الموجودة في أمر الطباعة للنصوص يجب أن تكتب داخل علامات اقتباس " ".
- هناك أمر يقوم بنفس وظيفة (\n) سطر جديد وهو : endl

- Ex:

```
cout << "Muslim";
cout << endl;
cout << "Engineer";
```

```
Muslim
Engineer
```

- لاحظنا سابقا انه لطباعة 3 مخرجات يجب كتابة (cout) 3 مرات، ولكن يمكن كتابتها مرة واحدة فقط كالتالي:

- Ex:

```
cout << "Muslim";
cout << endl;
cout << "Engineer";
```

```
cout<<"Muslim"<< endl<<"Engineer";
```

- **أمر الإدخال (cin):**
- يستخدم أمر cin لإدخال قيم إلى البرنامج، حيث يطلب من المستخدم إدخال قيمة والضغط على Enter
- يجب ان يتبع (cin) إشارة (<<) لتصبح : cin>> (عكس cout)
- عند استخدام أمر الإدخال (cin) فإننا نقوم بتخزين قيم داخل متغير وهذا المتغير يجب ان يكون معرف في البرنامج من حيث نوعه لذلك نستخدم جملة التعريف .

تعريف المتغيرات:

- يجب أن يبدأ المتغير بحرف وليس رقم:
  - A9b : ✓
  - A10 : ✓
  - 1Ab : ☒
- في نفس المتغير لا يوجد فراغ:
  - AB : ✓
  - A B : ☹
- ان لا يكون المتغير عبارة عن كلمة محجوزة في برنامج c++ مثل:

auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while...

وهذه الكلمات أغلبها أوامر في البرنامج.

- **الصيغة العامة لجملة تعريف المتغير:**

data type variable name ;

حيث من انواع الـ data type :

1. int	:	عدد صحيح	**Ex: -2, -1, 0, 1, 2.....**
2. double	:	عدد غير صحيح	**Ex: -2.3, 4.2, 0.2....**
3. float	:	عدد غير صحيح	
4. char	:	رمز أو حرف	**Ex: *, A, E, -, #, ...**

- **Ex:**

```
#include <iostream.h>
void main()
{ int x, y, res;
cin >> x >> y;
res = x+y;
cout << "the summation is ";
cout << res;
}
```

تعريف ثلاث متغيرات بجملة تعريف واحدة، وفصلنا بين المتغيرات بالفاصلة (,)

امر الإدخال (طلبنا من المستخدم ادخال قيم لـ x و y)

قيمة المتغير res تساوي مجموع المتغيرات x و y

طباعة الناتج (قيمة المتغير res)

www.muslimengineer.info
facebook.com/muslim.engineer.page

توضيح المثال:

اولا نعرف 3 متغيرات من نوع int :

x	y	res

يقوم البرنامج بالتوقف إلى حين إدخال قيم للمتغير x و y على الترتيب

كيفية الإدخال: نقوم بادخال رقم صحيح من لوحة المفاتيح ثم مفتاح الإدخال ↵ Enter

مثال: قمنا بإدخال 10 ↵ 20 ↵ يعني تخزين قيمة 10 في x و 20 في y

x	y	res
10	20	

يقوم البرنامج بإيجاد ناتج جمع الرقمين المدخلين وتخزينه في المتغير res.

x	y	res
10	20	30

res = 10 + 20 = 30

يقوم البرنامج بطباعة الناتج :

```
the summation is 30
```

ملاحظة: نفصل بين المتغير والآخر بفاصلة عادية (,) وعند نهاية المتغيرات نستخدم الفاصلة المنقوطة (;)

```
int x, y, res;
```

ويجوز تعريف المتغير بأسطر منفصلة

```
int x;
int  y;
int res;
```

ملاحظة: اذا عرفنا متغير على انه عدد صحيح وقمنا بإدخال عدد غير صحيح فإن الـ(compiler) يقوم بأخذ العدد الصحيح فقط:

Ex:

```
int x;
cin>>x;
```

ادخلنا (10.6) ⟵ تعتبر 10

لأن المتغير x من نوع int

Ex:

```
double x;
cin>> x;
```

ادخلنا (9.5) ⟵ تعتبر 9.5

ادخلنا (5) ⟵ تعتبر 5.0

تثبيت المتغير:

Ex:

```
int a, b, c;
a=10;
b=20;
cin>>c;
```

يقوم البرنامج بطلب إدخال عدد فقط للمتغير (c)

اما المتغيرات a,b فقد تم تحديد قيمتهم في البرنامج

facebook.com/Smurfs.On.The.Way

- Ex:

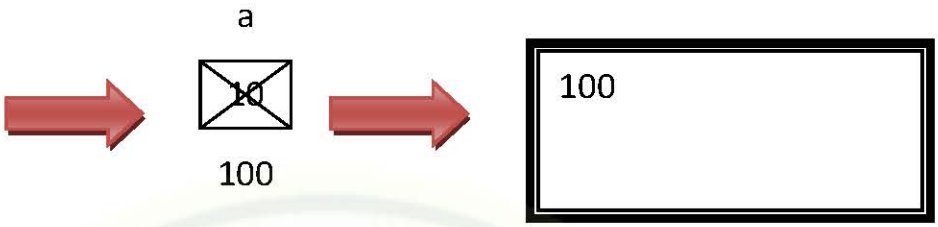
```
int a=10;
cin>>b,c;
```

هنا يقوم البرنامج بطلب ادخال عدد للمتغير b
اما المتغير c فيقوم بأخذ قيمة عشوائية له ولذلك لوجود فاصلة بين b و c.
(حتى يطلب البرنامج قيم للمتغيرين يجب ان تكون الصيغة cin>>b>>c;)

تغيير قيمة المتغير:

- Ex:

```
int a=10;
a=100;
cout << a;
```



العمليات الحسابية:

- الجمع (+).
- الطرح (-).
- الضرب (*).
- القسمة (/).
- باقي القسمة (%) (modulus).

- Ex:

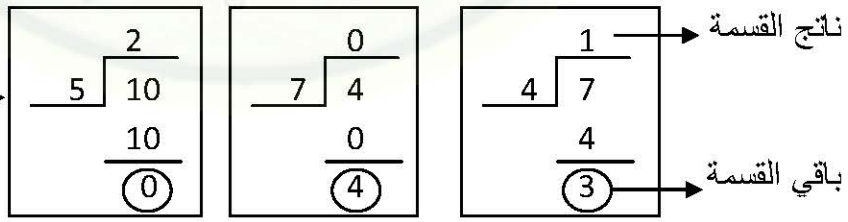
5+3 =8
5-3 =2
5*3 =15
5/3 =1
3/5 =0

اذا كان كلا البسط والمقام اعداد صحيحة فإن الناتج يكون صحيح بدون باقي.

5.0/3 =1.667
5/3.0 =1.667
5. 0/3.0=1.667

اذا كان اي من البسط او المقام اعداد عشرية فإن الناتج يظهر عدد كامل (صحيح + عشري)

7%4 =3
4%7 =4
10%5 =0



- إذا كان المطلوب باقي قسمة (modulus) وكان البسط أصغر من المقام فإن الناتج هو البسط نفسه.

www.muslimengineer.info
facebook.com/muslim.engineer.page

أولويات العمليات الحسابية:

١) الأقواس () من اليسار إلى اليمين
٢) ++ ، -- ، ! ، (سيدرس لاحقا) من اليمين إلى اليسار
٣) * / % من اليسار إلى اليمين
٤) + - من اليسار إلى اليمين
٥) < > >= <= من اليسار إلى اليمين
٦) == != من اليسار إلى اليمين
٧) && (and) (سيدرس لاحقا)
٨) || (or) (سيدرس لاحقا)
٩) += -= *= /= (سيدرس لاحقا)

- مثال: حول ما يلي إلى صيغة (c++):

$y=ax^2 + bx + c$

y=a*x*x + b*x + c

1 2 3

وذلك حسب ترتيب الأولويات يكون ترتيب الحل كالتالي:

عمليات المقارنة:

in math	In c++	Meaning
>	x > y	is x greater than y?
<	x < y	is x less than y?
≥	x >= y	is x greater or equal y?
≤	x <= y	is x less or equal than y?
=	x == y	is x equal y?
≠	x != y	is x not equal y?

- عملية مقارنة (هل a تساوي b ؟) a==b
- خزن قيمة b في a (تخزين) a=b

الجملة الشرطية (if) :

الصيغة العامة:

```
if (condition)
statement
```

- إذا كان الشرط condition صحيح فإن الجملة statement تنفذ.
- إذا كان الشرط condition خاطئ فإن الجملة statement لا تنفذ.
- ملاحظة: في جملة (if) لا يوجد فاصلة منقوطة (;).

facebook.com/Smurfs.On.The.Way

Ex:

```
#include <iostream.h>
void  main()
{ int a,b;
cin >> a >>b;                    //a=10, b=5
if (a>b)
cout <<"a is greater than b";
}
```

جملة تعليق لإعلام المستخدم بإدخال قيمة a=10 و b=5 (على فرض قام المستخدم بإدخال هذه القيم عند تنفيذ البرنامج)

الحل:

if (10>5) → صحيح → ينفذ

```
a is greater than b
```

إذا كان الشرط خاطئ فإن البرنامج لا يطبع شيء.

Ex:

```
#include <iostream.h>
int   main()
{ int x,y;
cout <<"Enter two integers\n";
cin >> x >>y;                    // x=3,   y=7
if (x==y)
cout <<x<<"is equal to"<<y<<endl;
if (x!=y)
cout <<x<<"is not equal to"<<y<<endl;
if (x<y)
cout <<x<<"is less than"<<y<<endl;
return 0;
}
```

```
Enter two integers
3
7
3 is not equal to 7
3 is less than 7
```

من إدخال المستخدم*

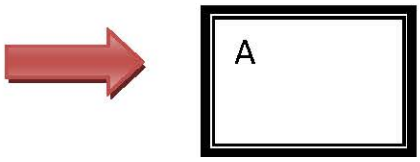
* ملاحظة هامة جدا: هنا افترضنا أن مستخدم البرنامج قام بإدخال القيم x=3, y=7 ..
وأي قيم يدخلها مستخدم البرنامج تظهر على شاشة المخرجات.. لذلك سنعتمد في هذا التلخيص إظهار ما يدخله المستخدم على شاشة المخرجات بدون أن تكون هذه نتائج من نفس البرنامج☺ .

www.muslimengineer.info
facebook.com/muslim.engineer.page

- **طباعة أو تخزين حرف (رمز):**
  - نستخدم الأمر (char) لتخزين حرف وليس عدد

- Ex:

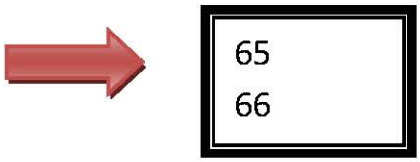
```
char x;
x= 'A';
cout << x;
```



- لتخزين حرف في متغير نستخدم ' ' وليس " ".
- إذا قمنا بتخزين حرف داخل متغير من نوع (int) فسوف يقوم بتخزين القيمة العددية للحرف حسب ترتيب خاص ببرنامج c++:

- Ex:

```
int x,y;
x= 'A';
y='B';
cout << x<<endl<<y;
```



القيمة العددية لحرف A تساوي 65

B=66, C=67,.....

وتختلف القيمة اذا كان الحرف صغير

a=97, b=98,.....

Chapter (2)

- **The if selection structure:**

الصيغة العامة:

```
if (condition)
statement 1
else
statement 2
```

- **Ex:**

```
#include <iostream.h>
void  main()
{ int grade;
cin >> grade;
if (grade >=60)
cout <<"passed";
else
cout <<"failed";
}
```

- اذا كان الشرط صحيح فإنه ينفذ statement 1 ويلغي statement 2.
- اذا كان الشرط صحيح فإنه ينفذ statement 2 ويلغي statement 1.

facebook.com/Smurfs.On.The.Way

- يمكن بطريقة أخرى كتابة جملة (if-else) بجملة واحدة فقط:

```
cout << (grade>=60 ? "passd" : "failed");
```

(condition) (if) (st1) (else) (st2)

- Ex:

```
if (grade >=60)
cout <<"passed\n";
cout <<"Good\n";
else
cout <<"failed\n";
```

☒ هنا البرنامج لا ينفذ، والسبب وجود جملة else غير متصلة بجملة if

✓ الصواب: وضع الجملتين بين أقواس {} :

- Ex:

```
int grade =60;
if (grade >=60)
{cout <<"passed\n";
cout <<"Good\n";
}
else
cout <<"failed\n";
```

- Ex:

```
int grade =60;
if (grade >=60)
{cout <<"passed\n";
cout <<"Good\n";
}
else
cout <<"failed\n";
cout <<"bad\n";
```

هنا الشرط صحيح: ينفذ جملة الـ if ويلغي else

جملة منفصلة، تنفذ حتى لو كان الشرط غير صحيح:

أما في حال تم وضع أقواس {} تحتوي على جملتي الـ if (failed) و(bad) فأن الناتج يكون فقط ما داخل جملة الـ if:

www.muslimengineer.info
facebook.com/muslim.engineer.page

Nested if/else structure:

الصيغة العامة:

```
if (condition 1)
statement 1
else if (condition 2)
statement 2
else
statement 3
```

كيفية الحل:

١) اذا كان (condition 1) الموجود مع if صحيح فإن statement 1 تنفذ وباقي الجمل تلغى.

٢) اذا كان (condition 1) خاطئ و (condition 2) صحيح فإن statement 2 تنفذ وباقي الجمل تلغى.

٣) اذا كان كلا من (condition 1) و (condition 2) خاطئ فإن statement 3 هي فقط التي تنفذ.

Ex:

```
int grade;
cin >> grade;
if (grade >=90)
cout <<"A";
else if (grade >=80)
cout <<"B";
else
cout <<"C";
cout <<"\nGood luck";
```

منفصلة

//grade = 100 → A Good luck

//grade = 80 → B Good luck

//grade = 50 → C Good luck

ملخص:

if-if	if-else	if -else if- else (nested)

الصيغة العامة:
if (condition 1)
statement 1
if (condition 2)
statement 2
	الصيغة العامة:
	if (condition)
	statement 1
	else
	statement 2
		الصيغة العامة:
		if (condition 1)
		statement 1
		else if (condition 2)
		statement 2
		else
		statement 3

الحل: يتحرك خطوة خطوة ويستمر بالتنفيذ للشروط الصحيحة.	الحل: ينفذ شرط واحد فقط إما (if) أو (else) . وفي حال تنفيذ احد الأمرين يلغى الآخر .	الحل: يتحرك خطوة خطوة حتى ينفذ شرط صحيح ثم يتوقف ويخرج إلى خارج loop.

facebook.com/Smurfs.On.The.Way

The switch Multiple selection structure:

- في جمل (switch) يتم تحديد عدد ثابت أو متغير، يبحث البرنامج عن الـ (case) المساوي للـ (switch) ليطبقها.

- مثال: اذا كانت (10) switch فإن أي من الـ case التالية تطبق؟

a- case 12 | b- case 10 | c- case 5 | d- case 7

- في برمجية (c++) يوجد احتمالين، إما وجود (case) مساوية لـ (switch) وإما لا.

1- اذا وجد (case) مساوية لـ (switch):

ينفذ الـ (case) ← اذا وجد (break) في الـ(case): ينهي

← اذا لم يجد (break) في الـ(case): يتابع الى الـ (case) التالية.

2- إذا لم يجد (case) مساوية لـ (switch): ينفذ الـ (default) بغض النظر عن مكانه.

- Ex:

```
#include <iostream.h>
int  main()
{int x;
cout<<"enter an integer:\n";
cin>>x;              // x=2
switch (x)
{ case 1 :
cout << " x is 1";
break;
case 2:
cout << " x is 2 ";
break;
default:
cout<<"value of x unknown";
}
return 0 ;
}
```

قمنا بإدخال قيمة (x) تساوي (2) يخزن في الـ switch العدد 2

نلاحظ استخدام النقطتين (:)

يقوم البرنامج بالبحث عن الـ (case 2) لإنها مطابقة لـ (switch2)

يقوم البرنامج بتنفيذ الجمل الموجودة في (case 2) وهما:
امر الطباعة (cout)
وأمر التوقف (break)

نهاية switch

```
enter an integer: 2
x is 2
```

- وظيفة (break): إنهاء جملة switch بحيث يذهب البرنامج لتنفيذ الجمل بعد ({) الذي يمثل نهاية الـ switch.
- جملة (default) اختيارية، وتنفذ عند عدم تطابق (switch) مع (case).

www.muslimengineer.info
facebook.com/muslim.engineer.page

- الصيغة العامة لجملة switch:

```
switch (variable)
{
case expression 1:
statement 1;
case expression 2:
statement 2;
..
..
default:
statement ;
}
```

- اذا لم توجد (break): يبقى البرنامج بالتنفيذ من الـ case المتطابق حتى بعد الـ (default) ({).

- Ex:

```
switch (10)
{
case 10 :
cout << " Mechanical Team" << endl;
cout << " Muslim Engineer " << break;
}
```

عند تنفيذ أكثر من جملة في الـ (case)
لا داعي لوجود أقواس { }

```
Mechanical Team
Muslim Engineer
```

- Ex:

```
switch (10)
{ case 10 :
cout << " Mechanical Team"; break;
cout << " Muslim Engineer ";
}
```

لا يطبع Muslim Engineer وذلك بسبب وجود (break)

```
Mechanical Team
```

- Ex:

```
switch (10)
{ case 10 :
cout << " x is 1 " << endl;
case 2 :
cout << " x is 2 " << endl;
default  :
cout << " x is unknown " << endl; }
```

هنا تنفذ كل الجمل بسبب عدم وجود (break)

```
x is 1
x is 2
x is unknown
```

- إذا كان الـ (case) المتطابق موجود في آخر البرنامج ولم يكن موجود (break) فإن الـ (case) ينفذ ثم ينهي البرنامج، أي طرق إنهاء برنامج الـ(switch):

1- break

2- case في نهاية البرنامج

- Ex:

```
#include <iostream.h>
void  main()
{int x, y;
cin>> x >> y;          //x=5    y=5
switch (x+y)           ──────►   switch 10  تصبح
{case 10:
cout << "value is 10 " ; break;
case 3:
cout << "value is 3 " ; break;
case 100:
cout << "value is 100 " ; break;
}
cout <<"\n";
cout << x + y;
}
```

```
value is 10
10
```

- Ex:

```
#include <iostream.h>
void  main()
{ char (x):            ──────►   لتعريف حرف أو رمز
int y, z;              ──────►   لتعريف أرقام صحيحة
cout << "Enter an operator";
cin >> x;              //x=+
cout << "\nEnter two integers";
cin << y << z;         // y=2, z=3
switch (x) {
case '+':
cout << "\nThe result is : " << y + z; break;
case '-':
cout << "\nThe result is : " << y -  z; break;
}
}
```

قمنا بإدخال المتغيرات التالية:

تخزن في switch رمز الجمع (+)

نطابق الـswitch مع الـ case

وننفذ الـ case

```
Enter an operator
Enter two integers
The result is : 5
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

Logical Operators المعاملات المنطقية

- تحدثنا سابقا في الأولويات عن بعض هذه المعاملات بالإضافة لعمليات المقارنة.. الآن سندرسها بالتفصيل ونستخدمها في الجمل الشرطية:

Ex:

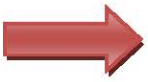
```
#include <iostream.h>
void  main()
{ int x;
if (x >= 5 )
cout << "hello";
else
cout << "welcome"; }
```

استخدمنا عملية مقارنة في جملة شرطية.. إذا كانت قيمة x أكبر أو تساوي 5 عندها يتحقق الشرط True وينفذ الجملة الشرطية
أما إذا كانت قيمة x اقل من 5 عندها يصبح الشرط خاطئ False وينفذ جملة else

- ملاحظة: أي عدد داخل جملة if يكون دائما صحيح إلا إذا كان قيمته صفر فهو خاطئ.

Ex:

```
if (15 )
cout << "hello";
else
cout << "welcome";
```



hello

- لتحويل جمل المقارنة المركبة (مثل $90 < x \leq 100$) إلى صيغة برمجية (c++) نستخدم عدة معاملات:

1- && (and).
2- || (or).
3- ! (not).

$90 < x \leq 100$ ⟶ $x > 90$ <u>&&</u> $x <= 100$ (and)

facebook.com/Smurfs.On.The.Way

- والآن سنوضح طريقة عمل هذه المعاملات عند وضعها بين شرطين في كل حالة يكون الناتج:

1) && (and): - فقط في حال كان الشرطين صحيحان يكون الناتج صحيح

Condition A	Condition B	A&&B
True	True	True
True	False	False
False	True	False
False	False	False

2) || (or) : - إذا كان أي من الشروط صحيح يكون الناتج صحيح

Condition A	Condition B	A ||B
True	True	True
True	False	True
False	True	True
False	False	False

3) ! (not) : - تقوم بعكس الشرط

A	!A
True	False
False	True

- Ex : -

```
int x ;
cin >> x;          // x=80
if (x  >=80 && x<=90)
cout<<" A";
else
cout << "B" ;
```

التوضيح

```
if (80>=80  &&  80<=90)
    (True)        (True)
if (Ture && True)
if True
```

حسب الجدول

ما داخل جملة (If) هو (True)
هذا يعني انه يطبق جملة (If)
ويتجاهل جملة الـ (else).

A

www.muslimengineer.info
facebook.com/muslim.engineer.page

- Ex :-

```
X=80 ;
✓ { if (X > =80 || X< =90)
    cout<< "A" ;
☒ { else
    cout << "B"
```

→ if (80>=80 ||80<=90)
(True) (True)
if (Ture || True)
if True

→ A

- Ex: -

```
Grad =60 ;
☒ { if (!(Grad >=50)
    cout << "Failed" ;
✓ { else
    cout <<" Passed"  ;
```

→ if (! (60 >=50)
if (! (Ture)
if (false)
else يطبق جملة

→ Passed

- Notes: -
  - cout << True ; → 1
  - cout << false ; → 0

يطبع

- Ex : -

cout<< (17|| 0);

(T) (F) ➔ (T|| F) ➔ (T)

➔ cout<<True ;

→ 1

facebook.com/Smurfs.On.The.Way

Ex :-

```
cout << (10 && 0)
```

(T) (F) (T&&F) ➔(F)

➔ cout<<False ;

0

Ex : -

```
if (x==4)
cout <<"A" ;
else
cout <<" B " ;
```

هنا لا يتحقق الشرط إلا بعد إدخال قيمة (X) ثم يفحص الشرط لينفذ عملية المقارنة

Ex: -

```
If (X=4)
cout << "A"
Else
cout<<" B"
```

هنا دائما ينفذ جملة (if) لأنها صحيحة دائما (عملية تخزين)

اذا كانت عملية التخزين تخزن الصفر في المتغير يعتبر الشرط false

اما اذا تم تخزن أي رقم لا يساوي صفر في المتغير يعتبر الشرط True

Ex: -

```
int x= 5 ;
if (x =! 6)
cout <<" yes" ;
else
cout<< " No" ;
```

X=! 6

➔ X = 0

If (0) (False)

No

Ex: -

```
int x= 5 ;
if (x != 6)
cout<<" yes" ;
else
cout<<" No" ;
```

X != 6

(5 ≠ 6) ➔ (True)

If (True)

yes

www.muslimengineer.info
facebook.com/muslim.engineer.page

- **Increment and Decrement operates : -**

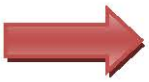
(++X) or (X++) ↓ (- -X) or (X - -) ↓ تعني زيادة أو نقصان بمقدار واحد

- ++ x : pre increment = increment (x) by (1) , then use the new value
عندما تكون الإشارة قبل المتغير عندها يزيد المتغير بمقدار 1 ثم يتم استخدام القيمة الجديدة.

- X++ : post increment = use the current value then add (1)
أما عندما تكون الإشارة بعد المتغير عندها يستخدم المتغير ثم يضيف إليه مقدار 1.

- Ex : -

```
# include < iostream.h>
{ int main ()
1)  int c=5 ;
2)  cout << c <<endl ;
3)  cout << c++<<endl ;
4)  cout<< c<<endl ;
5)  C=5 ;
6)  cout << ++c << endl;
7)  cout<<c ;
    return 0;}
```



```
5
5  ──→  (post)
6
6  ──→  (pre)
6
```

- التوضيح: -

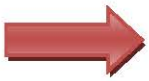
١) بدأت قيمة (c) بـ (5).
٢) استخدمنا أمر طباعة للمتغير (c) الذي قيمته (5).
٣) ++ c:- (post) أي استخدام القيمة القديمة وهي (5) في الطباعة
وبعد الطباعة تم زيادة (1) على (c) والذي قيمته (6).
٤) استخدمنا أمر طباعة (cout) للمتغير (c) والذي قيمته (6) بعد زيادة .
٥) عدّلنا على قيمة (c) لـتصبح (5) بدالا من (6).
٦) C++ ; (pre) قمنا بإضافة (1) إلى (c) لتصبح (c) تساوي (6) ثم قمنا بالطباعة .
٧) استخدمنا أمر الطباعة للمتغير (c) والذي قيمته (6).

- هناك شواذ (استثناء) لقاعدة الـ (pre) والـ (post) حيث في هذه الحالة يطبع دائما (pre) وهذه الحالة هي وجود (++x) أو (x++) في سطر لوحدهما:

facebook.com/Smurfs.On.The.Way

Ex: -

```
X=10 ;
X++ ;
cout << X ;
```



11

Ex : -

```
X=10 ;
++ X;
cout << X;
```

11

X++ أو ++ X موجودة في سطر منفصل لذلك تستخدم قيمة pre دائما في الطباعة

Ex : -

```
X=10 ;
X = X ++ ;
cout << X ;
```

11

Ex :-

```
x= 10 ;
x--;
 - - x;
X= -- x;
cout << x;
```

7

Important Example:-

```
int x=5 , y=0 , z= 3 ;
Z+=++x + y++ - z++ ;
cout << z<< x <<y ;
```

تذكر : -

الأولويات :-

١) ++ ، -- -- (من اليمين إلى اليسار)

٢) + ، - (من اليسار إلى اليمين)

٣) = +

www.muslimengineer.info
facebook.com/muslim.engineer.page

Solve:-

Z+ = ++x + y++ - z++

(pre) (post) (post)

من اليمين إلى اليسار

x	y	z
~~5~~	~~0~~	~~3~~
6	1	~~4~~
		7

Z+ = ++ x + y++ - 3

Z+ = ++ x + 0 – 3

Z + = 6 + 0 – 3

Z+= 3 ⟶ Z= Z+3 ➔ Z= 4 + 3

Z= 7

```
cout << z << x <<y ;
```

761

- Ex : -

```
Char x = ' A' ;
X ++ ;
cout <<x ;
```

B

- ملاحظة : 1+ حرف = حرف الذي يليه

- Ex : -

```
int  a=3 , b=1 , c=5, y =9 ;
C= a/b + --y ;
cout << c <<y ;
```

118

- التوضيح : -

C= a/b + --y

C = a/b +8

C =3/1 +8

C =3+8

C =11

A	B	C	Y
3	1	~~5~~	~~9~~
		11	8

facebook.com/Smurfs.On.The.Way

- ملاحظة على جملة if :

- Ex : -

```
if ('A' < = 'B')
cout<< " Good" ;
else
cout<< " Bad" ;
```

```
Good
```

- في برمجية (c++) تكون الحروف الصغيرة أكبر من الحروف الكبيرة : -

الأكبر z < y < x < ..... < b < a < Z < Y < ..... < C < B < A الأصغر

- The while Repetition structure : -

الصيغة العامة لجملة (while) : -

```
Initialization value
While (condition)
Statement (s)
Increment /decrement
condition
```

- Ex : -

```
int product =2 ;
while ( product < = 1000)
Product =2 *  product ;
cout <<product ;
```

```
1024
```

- في جملة while يجب توفر3 شروط وهي : -

١) Initialization

٢) Condition

٣) Increment /decrement condition

ثم تفحص الشرط الموجود مع (while) إذا كان صحيح نطبق أول جملة تلي (while) ثم نرجع نفحص الشرط مع (while) مرة أخرى، فإذا مازال صحيح نطبق أول جملة تلي (while) مرة أخرى وهكذا حتى يصبح الشرط خاطئ نذهب إلى ثاني جملة تلي (while).

www.muslimengineer.info
facebook.com/muslim.engineer.page

توضيح المثال السابق : -

product
~~2~~
~~4~~
~~8~~
~~16~~
. .
~~512~~
1024

١) عرفنا متغير اسمه product وخزّنا فيه قيمة (2) (initialization)

٢) نفحص الشرط الموجود مع (while) (2<=1000)

٣) بما أن الشرط صحيح نطبق أول جملة تلي (while) وهي (increment)

Product (new)= product(old) * 2

Product =2* 2=4

٤) نرجع مرة أخرى إلى الشرط الموجود مع (while) ونفحصه : (4<=1000) الشرط صحيح

٥) نطبق أول جملة تلي while : product=2*4=8

٦) وهكذا حتى تصل قيمة الـ (product) إلى (1024) فعندها الشرط يصبح خاطئ فنذهب إلى الجملة الثانية التي تلي جملة (while) وهي أمر الطباعة

٧) أمر الطباعة يطلب طباعة أخر قيمة (product) نذهب إلى المربع ونأخذ أخر قيمة مخزنة فيه ونطبعها .

ملاحظة : -

1)يبقى الـ (loop) ينفذ حتى يصبح الـ (condition) خاطئ

2) إذا أردنا تنفيذ أكثر من جملة في أثناء الـ (loop) نضع الجمل بين { }.

Ex: -

```
product =2 ;
while (product <=2)
{cout<< product << endl ;
product =2*product ;}
cout << product ;
```

ينفذ طالما الشرط صحيح

ينفذ طالما الشرط خاطئ

Product
~~2~~
4

```
2
4
```

Notes: -

1) while (0) (false) : ⟶ لا ينفذ أبدا

2) while (any number) (True) : ⟶ ينفذ إلى المالانهاية

3) في كل مرة تنفذ الجمل التي بعد (while) يجب أن تغير الـ (condition) إما بزيادة أو نقصان من أجل عندما نرجع نفحص الـ (condition) مرة أخرى يكون قد تغير .

facebook.com/Smurfs.On.The.Way

Ex :-

```
#include <iostream.h>
void main ()
{int total, GradeCounter, grade, average;
Total =0 ;
GradeCounter =1 ;
while (GradeCounter <= 10)
{cout<< "enter grade: " ;
cin>>grade ;
total =total +grade ;
GradeCounter +=1;}
Average =total /10 ;
cout<<"class average is'' <<average<<\n ;}
```

```
Enter grade : 98
Enter grade : 71
.
.
Class average : 81
```

التوضيح: -

١) عرفنا (4) متغيرات هم (total, GradeCounter , grade , average)

٢) نفحص الشرط (while) : (10 <= 1) صحيح.

٣) ننفذ الجمل بين { } .

ملاحظة :

total = total + grade

الجديدة القديمة

GradeCounter + =1

GradeCounter=grade counter +1

The for Repetition structure:

```
For (initialization value ; condition ; ( increment /decrement)
Statement (s) ;
```

إذا كانت فاصلة عادية (,) يعطي error

www.muslimengineer.info
facebook.com/muslim.engineer.page

تسلسل التنفيذ : -

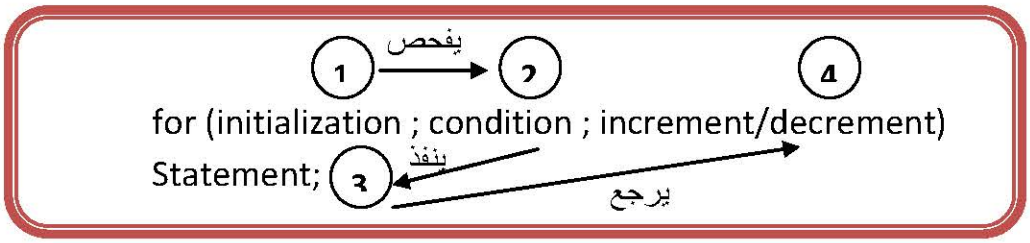
١) يأخذ (initialization value)

٢) يفحص الـ (condition) ← صحيح : ينفذ جملة ((statement (s)

← خاطئ : ينفذ الجملة ما بعد جملة ((statement (s)

٣) إذا كان الـ (condition) صحيح فعند الانتهاء من تنفيذ جملة الـ ((statement(s)) يرجع وينفذ الـ (increment/decrement) ثم يفحص الـ (condition) مرة أخرى .

٤) إذا كان الـ (condition) خاطئ لا يرجع إلى (increment /decrement) (يخرج من الـ loop)



Ex : -

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

```
1
2
3
4
5
6
7
8
9
10
```

التوضيح:

1) قمنا بتعريف المتغير (counter) وأعطي قيمة ابتدائية مقدراها (1) "إذا لم يعطى قيمة ابتدائية يعطيك برنامج (error) (لا ينفذ برنامج)".

2) نفحص الشرط (counter <=10) ← (1<=10) ✓

3) ننفذ جملة الطابعة وهي طباعة مقدار الـ (counter) الذي يساوي (1).

4) نرجع إلى الـ (increment) ← counter++ وهي زيادة على المتغير counter بمقدار واحد ليصبح الـ counter يساوي (2).

5) نفحص الشرط مرة أخرى (2<=10)✓ وننفذ جملة الطباعة وهكذا .....

6) عندما نصل إلى المقدار الـ (counter) يساوي (10) نفحص الشرط (10<=10) ✓ ننفذ جملة ونزيد على الـ (counter) ليصبح (11).

7) نفحص الشرط (11<=10) ☒ خاطئ ننفذ الجملة التي بعد جملة الطباعة وهي (return 0;) أي إنهاء البرنامج.

facebook.com/Smurfs.On.The.Way

- ملاحظة : - لتنفيذ أكثر من جملة عندما يكون الشرط صحيح نضع الجمل بين{ }.

- Ex : -

```
# include < iostream .h >
void main ()
{ int sum =0 ;
for (int number =2 ; number <= 100 ; number +=2)
sum+=number ;
cout <<" sum is :" << sum ;
}
```

التوضيح:

2 <= 100
Sum=sum+ number
sum=0+2=2
4 <= 100
Sum= sum + number
Sum=2+4=6
6 <= 100
Sum= sum + number
Sum=6+6=12
.....etc

Sum	number
~~0~~	~~2~~
~~2~~	~~4~~
~~6~~	~~6~~
~~12~~	~~8~~
..	..
2550	~~100~~
	102

```
sum is : 2250
```

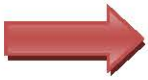
- Note : -

هذه الجمل الثلاث متشابهة : -

1) for (int j=2 ; j < =80 ; j+=5)
2) int j=2;
   for(j; j<=80 ;j+=5)
3) int j;
   for (j=2 ; j < = 80 ; j= j+5)

- Ex : -

```
for (char x ='A' ; x < = 'C' ; x++)
cout << x<< endl;
```



```
A
B
C
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

The (do – while) structure : -

الصيغة العامة : -

```
Do {
Statement (s)
}
While (condition);
```

ملاحظات : -

١) الفرق بين جملة (while) و جملة (do –while)

- في جملة (while) نفحص الشرط أولا ثم ننفذ الجمل
- وفي جملة (do –while) ننفذ الجمل أولا ثم نفحص الشرط

٢) يجب وضع فاصلة منقوطة (;) بعد (while (condition و الا يعطي (error)

Ex : -

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

يطبع فراغ

يرجع إذا كان الشرط صحيح

يستمر إذا كان الشرط خاطئ

Counter		
~~1~~	2< =10	✓
~~2~~	3< =10	✓
~~3~~	4< =10	✓
~~4~~	10<= 10	✓
..	11< =10	☒
~~10~~		
11		

1 2 3 4 5 6 7 8 9 10

يترك سطر جديد

facebook.com/Smurfs.On.The.Way

- Ex : -

```
#include <iostream .h>
void main ()
{ int counter =10 ;
Do {
cout << counter << "\t" ;
} while (++ counter < =10 ) ;
cout << endl ;

cout <<"welcome" ;}
```

```
10 ..... فراغ
welcome
```

- لاحظ انه بالرغم من أن الشرط كان خاطئ منذ البداية إلا أن البرنامج طبع الجملة مرة واحدة وهذا هو أساس عمل جملة (do -while)

The break instruction : -

- Ex:

```
# include <iostream.h>
int main ()
{int n ;
for (n =10 ; n> 0 ; n --)
{cout << n << "," ;
if (n==3)
{
cout << "abort !" ;
Break ;
}}
return 0 ;
}
```

```
10,9,8,7,6,5,4,3,abort!
```

- التوضيح : -

وظيفة (break) : - توقف (loop)

(أنواع loop: for , while , do while , switch)

وفي المثال السابق الأمر (break) ينفذ عندما يكون شرط (if) صحيح لأنه موجود ضمن جملة (if) ويقوم بالخروج من (loop).

www.muslimengineer.info
facebook.com/muslim.engineer.page

The Continue instruction : -

- Ex : -

```
#include <iostream.h>
void main()
{ for (int n=10 ; n>0 ; n -- -- ) {
if ( n== 5)
Continue ;
cout << n<< "," ; }
}
```

10,9,8,7,6,4,3,2,1

- وظيفة (continue):- تستثني جمل بعدها (كل الجمل الموجودة في loop)

(while ,do –while) ➔➔ continue to condition

(for) ➔➔ continue to increment /decrement

- في المثال السابق (continue) ينفذ عندما يكون شرط (if) صحيح لأنه موجود ضمن جملة (if) ويقوم باستثناء (إهمال) الجمل التي بعده في الـ (loop) والجملة التي تم استثناءها هنا هي :

cout<<n<< "," ;

ولذلك لم يطبع العدد 5 في شاشة المخرجات،

وبعد الاستثناء يرجع إلى (increment /decrement) ليعود إلى الـ (loop) من الجديد .

- Ex : -

```
int x =0 ;
while (x!=5)
{x++ ;
Continue ;
cout<<x; }
```

(no output)

0≠5	✓	Continue (لا ينفذ الطباعة)
1≠5	✓	Continue (لا ينفذ الطباعة)
2≠5	✓	Continue (لا ينفذ الطباعة)
3≠5	✓	Continue (لا ينفذ الطباعة)
4≠5	✓	Continue (لا ينفذ الطباعة)
5≠5	☒	➔ exit loop

Chapter (3): - Arrays (المصفوفات)

الصيغة العامة للمصفوفة : -

Data type Array name [size] ;

Data type: (int , float , double ,char ,.....)

- Ex : -

int A [10];

تعني حجز (10) مواقع في المتغير (A) من نوع (integer)

المؤشر →	0	1	2	3	4	5	6	7	8	9
A=	1	2	3	4	5	6	7	8	9	10

- المؤشر يبدأ دائما من الصفر وينتهي بـ (size -1)
- لتحديد عنصر من عناصر المصفوفة نستخدم اسم المصفوفة و رقم مؤشر العنصر

مثال : - لتحديد العنصر الأول في المصفوفة ← A[0]

لتحديد العنصر الأخير في المصفوفة ← A[9]

لتحديد العنصر الثالث في المصفوفة ← A[2]

- لطابعة عنصر من العناصر المصفوفة :

يطبع أول عنصر في المصفوفة cout << A[0] ;

- ملاحظة : - عدد الأرقام المخزنة في المصفوفة => size

- فإذا كان (size) لمصفوفة يساوي (5) فإننا نستطيع إدخال أرقام (عناصر) لهذه المصفوفة بمقدار (5) عناصر أو أقل ولا يجوز إدخال (6) عناصر أو أكثر .

- لإدخال عناصر داخل المصفوفة هناك طريقتين : -

١) باستخدام (loop) ← (..for) [(0) - (size -1)] .

٢) باستخدام التخزين المباشر [,].

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex : -

```
#include <iostream.h>
void main ()
{ int i , N[10];
for (i=0 ; i<10 ; i++)
N[i] =0 ;
cout << "Element " <<"\t" <<"value" << endl;
for(i=0 ; i< 10 ; i++)
cout<<i<< "\t"<< N[i] <<endl;}
```

Loop { for (i=0 ; i<10 ; i++) N[i] =0 ;

Loop { for(i=0 ; i< 10 ; i++) cout<<i<< "\t"<< N[i] <<endl;}

تنفذ عندما يكون الشرط خاطئ

Element	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

التوضيح : -

1) قمنا بتعريف متغيرين أحدهما متغير عادي والأخر مصفوفة N[10].

2) استخدمنا جملة (for) الأولى لإدخال العناصر للمصفوفة N[10] ونبدأ الـ (loop) من الصفر إلى (9)

(size -1)

	0	1	2	3	4	5	6	7	8	9
N	0	0	0	0	0	0	0	0	0	0
	N[0]	N[1]								N[9]

i

~~0~~

~~1~~

2

0<10 ✓

N[0]=0 → خزن في الموقع N[0] المقدار (0)

1<10 ✓

N[1]=0 → خزن في الموقع N[1] المقدار (0)

2<10 ✓

N[2]=0

...

9<10 ✓

N[9]=0

10<10 ☒

٣) استخدمنا جملة (for) الثانية لطباعة عناصر المصفوفة n[10] ويبدأ الـ (Loop) من الصفر إلى (9) ← (size-1)

facebook.com/Smurfs.On.The.Way

Ex: -

```
#include <iostream.h>
void main ()
{ int n [10] ={32, 27 , 64 ,18 ,95, 14, 90, 70, 60, 37};
cout<< "Element " <<"\t" << "value" << endl ;
for ( int i=0 ; i< 10 ; i++)
cout<< i<<"\t" << n[i] << endl ;}
```

طريقة أخرى لتخزين العناصر داخل المصفوفة

جملة (for) لطابعة المصفوفة

Element	Value
0	32
1	27
2	64
3	18
4	95
5	14
6	90
7	70
8	60
9	37

Notes : -

1) int n [10] ={0} ; جميع المواقع تخزن صفر

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

2) int n[5] ={32,27,64,18,95,14};

يعطي (error) لان عدد عناصر المخزنة > size

5 < 6

3) int n[5] ={32 ,27 , 64 , 18};

	0	1	2	3	4
n	32	27	64	18	0

عدد العناصر المخزنة > size

4 < 5 ⟵ يخزن آخر عنصر المقدار صفر

www.muslimengineer.info
facebook.com/muslim.engineer.page

4) int n [5] ={1};

يخزن في أول موقع المقدار (1) أما بقية المواقع فيخزن المقدار صفر

	0	1	2	3	4
n	1	0	0	0	0

5) int n [ ] = { 1, 2 , 3 , 4 , 5}

- إذا لم يحدد المبرمج الـ (size) ⟵ (صحيح) (ليس error)

يقوم البرنامج بتحديد الـ (size) من خلال عدد العناصر المخزنة وهنا عدد العناصر المخزنة size =5

Constant variable : -

Ex : -

```
# include <iostream.h>
int main ()
{const int x=7 ;
cout << x<< endl;
return 0 ; }
```

⟶ 7

- ملاحظة : - إذا عرّف متغير باستخدام (constant) يجب إعطائه قيمة ابتدائية وإلا يعطي البرنامج (error)، ولا يجوز تغيير قيمة المتغير .

Ex : -

```
Const int x=5 ;
X++ ;
cout <<x ;
```

Error

لا يجوز تغير قيمة (x)

Ex :- const int x ;

```
X=10 ;
cout <<x ;
```

Error

يجب إعطاؤه قيمة ابتدائية في نفس السطر

Ex : -

```
Const int x =7 ;
cout << x+x ;
cout << x+2 ;
cout << x+=x ;
```

cout << x+x ; ⟶ 14

cout << x+2 ; ⟶ 9

cout << x+=x ; ⟶ Error

facebook.com/Smurfs.On.The.Way

ملاحظة : - عندما قمنا بتحديد الـ (size) للمصفوفة في الأمثلة السابقة استخدمنا الأرقام:

```
int n [5] ;
```

- ولكن يجوز تحديد الـ (size) للمصفوفة باستخدام المتغير على أن يكون المتغير من نوع (const):

```
Const int x = 5 ;
int n [x] ;
```

Ex : -

```
#include  <iostream.h>
void main ()
{ const int arraysize  =10 ;
int j , s [arraysize] ;
for (j=0 ; j < arraysize ; j++)
S[j] =2+2*j ;
cout << "Element " << "\t" << "value" <<endl;
for (j=0 ; j< arraysize ; j++)
cout<< j<<"\t" << s[j] <<endl;
}
```

Element	Value
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20

	0	1	2	3	4	5	6	7	8	9
S	2	4	6	8	10	12	14	16	18	20

التوضيح : -

j	
~~0~~	S[j] =2+2*j
~~1~~	S [0]=2+2*0=2+0=2
~~2~~	S [1]=2+2*1=2+2=4
.	.
.	.
~~9~~	.
10	S[9] =2+2*9=2+18=20

عندما يوجد جملتين (for) نبدأ من جملة (for) الأولى (i=0) ثم ننتقل إلى (for) الثانية ونطبقها كاملة (j=0 ➔3).

1) for (int i=0 ; i<5 ; i++)

2) for(int j =0 ; j<4 ; j++)

- ثم نرجع مرة أخرى إلى (for) الأولى (i=1) ثم ننتقل إلى (for) الثانية (j=0➔ 3)
- ثم نرجع مرة أخرى إلى (for) الأولى (i=2) وهكذا حتى الانتهاء من (for) الأولى (يصبح شرطها خاطئ).

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex : -

```
#include <iostream.h>
void main() {
Const int arraysize=10;
int n [arraysize] =[3,7,9,5,4,11,8,6,13,1];
cout<< "Element " <<"\t" <<"value" <<"\t"<<"shape" <<endl;
for (int i=0 ; i<arraysize ;i++)
{
cout<<i<<"\t"<<n[i]<<"\t" ;
for (int j=0 ;<n[i] ; j++)
cout <<"*";
cout<<endl;
}
}
```

Loop(2)

Loop(1)

Element	Value	shape
0	3	***
1	7	*******
2	9	*********
3	5	*****
4	4	****
5	11	**********
6	8	********
7	6	******
8	13	**********
9	1	*

تخزين حروف في المصفوفة : -

Ex : -

Char string []="First" ;

(data type) (arrayname) (5 char +\0= 6 char)

Null character

	0	1	2	3	4	5
string	F	i	r	s	t	\0 → (فراغ)

ملاحظة :-

cout<<string [2]; → r

cout<<string [5]; → \0(space)

} لطباعة عناصر من المصفوفة

cout<<string ; → first لطباعة المصفوفة كاملة

facebook.com/Smurfs.On.The.Way

Ex : -

```
char string [  ]={' f ', ' i ', 'r ' ,' s ', ' t ', '\ 0'} ;
```

اختيارية إضافتها ↓ عدد الحروف < size

- طرق تخزين حروف في المصفوفة :-

١) طريقة مباشرة باستخدام " " ← "first"

٢) طريقة مباشرة باستخدام { } ← {' f ', ' i ', ' r ', ' s ', 't '}

٣) باستخدام أمر الإدخال (cin) بحيث يجب أن يكون عدد الحروف المدخلة اقل من (size) بمقدار واحد(1) على الأقل .

- لإدخال (تخزين) حروف في المصفوفة لا نحتاج إلى جملة (Loop).
- عند استخدام الطريقة الثالثة في إدخال حروف في المصفوفة باستخدام أمر الإدخال (cin) تعتبر المفاتيح (Tab,Enter,Space) أوامر لنهاية الإدخال.

مثال : - إذا أدخلنا من لوحة المفاتيح الحروف التالية : -

ABCD AB

فإن البرنامج يقوم بأخذ الحروف (ABCD) فقط .

EX :-

```
Char x[20] ;
```

- حجم المصفوفة يساوي (20) يعني يجب إدخال (19) حرف فقط.

- لو أدخلنا (20) أو (21) حرف سوف يعطي البرنامج (error).
- لو أدخلنا (18) أو (17) حرف لا يعطي (error) وإنما يقوم بتخزين (\0) مكان الموقع الفارغ.

Ex : -

Char x [5] ="str";

	0	1	2	3	4
x	s	t	r	\0	\0

Ex : -

```
#include <iostream.h>
void main ()
{ char string1 [20] ,string2 [ ] =" Mechanical Team ";
cout << "Enter a string:" ;
cin >> string1;                    //Muslim Engineer
cout<< "string1 is :" << string 1<< "\n string 2 is : " ; << string 2<<"\n";
for (int i=0 ; string1 [i] !=  ' \ 0 ' ; i ++)
cout << string1 [i] << '   ' ;
cout <<endl ;
cin >> string1 ;
cout<<string1<<endl;}
```

لاسترجاع الكلمة الثانية من الإدخال

```
Enter a string: Muslim Engineer
String1 is : Muslim
String2 is : Mechanical Team
M u s l i m
Engineer
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

- التوضيح : -

١) قمنا بتخزين مصفوفتين حروف.

٢) المصفوفة الأولى (string1) يتم إدخال الحروف إليها باستخدام (cin) وقمنا بإدخال (Muslim Engineer) وبما انه يوجد (space) بين الكلمتين فإن البرنامج يقوم بأخذ الكلمة الأولى فقط (Muslim) .

٣) عند طباعة المصفوفة (string1) يقوم بطباعة (Muslim) فقط.

٤) عند طباعة المصفوفة (string2) يقوم بطباعة (Mechanical Team) كاملة.

- ملاحظة: هناك فرق بين الـ (space) الموجود في كلمة (Muslim Engineer) و الـ (space) الموجود في كلمة (Mechanical Team)، ففي كلمة (Muslim Engineer) بسبب استخدمنا أمر (cin) فهنا (space) تعني نهاية الإدخال أما في كلمة (Mechanical Team) فهنا الـ (space) عبارة عن حروف فلو سأل كم عدد الحروف في كلمة (Mechanical Team) عبارة عن (15) حرف.

- ملاحظة: تعرفنا سابقاً انه عند استخدام الأمر (cin) فإن المفتاح (Tab, Enter, Space) تمثل نهاية إدخال و يقوم البرنامج بإهمال الكلمة الثانية بعد (space) ولكن إذا أردنا طباعة هذه الكلمة (المهملة) فإننا نقوم بما يلي : -

```
cin>> string 1;
cout << string1;
```

بالرغم من استدعاء أمر الإدخال (cin) فان البرنامج لا يطلب من المستخدم إدخال حروف وإنما يقوم باسترجاع الكلمة الثانية مباشرة وطباعتها .

Multiple subscripted Array (Matrix) : -

A [i] [j] ➔➔ A[0] [0]: first element in array

Array name — (الصف) row — العمود Column

Ex : -

A[3][4];

		C0	C1	C2	C3	
A	r0	A[0][0]	A[0][1]	A[0][2]	A[0][3]	عدد الصفوف = 3
	r1	A[1][0]	A[1][1]	A[1][2]	A[1][3]	عدد الأعمدة = 4
	r2	A[2][0]	A[2][1]	A[2][2]	A[2][3]	

Ex : -

int b[2][2] ;

		C0	C1
b	r0	b [0] [0]	b [0][1]
	r1	b [1] [0]	b [1][1]

عدد الصفوف =2

عدد الأعمدة = 2

Cout<<b[1][1];

- طرق إدخال العناصر إلى (matrix) :
  ١) طريقة مباشرة باستخدام { }.
  ٢) باستخدام (for).

Ex : -

int b [2] [2] = {{1,2} , {3,4}};

cout << b[0][1];

- تسمى هذه الطريقة في التعبئة طريقة الأزواج المرتبة وتقوم هذه الطريقة على تعبئة الـ (matrix) صف صف.

		C0	C1
b	r0	1	2
	r1	3	4

b[0][1]

Ex : -

int b [2] [2] = {{ 1} , {3,4}};

		C0	C1
b	r0	1	0
	r1	3	4

الأماكن الفارغة من الـ(matrix) تعتبر صفرا

Ex :

int array [2] [3] = {1,2,3,4,5} ;

	C0	C1	C2
r0	1	2	3
r1	4	5	0

هنا تم استخدام الأقواس فقط.

Ex : (

int b[2][3] ;

for (int i=0 ; i<2 ; i++) ⟶ الصفوف (0 – 1)

for (int j=0 ; j<3 ; j ++) ⟶ الأعمدة (0 – 2)

cin >> b[i][j] ;

باستخدام جملة الـ (For)

- لاحظ دائماً حدود الـ (for) تبدأ من الصفر إلى (size-1).

www.muslimengineer.info
facebook.com/muslim.engineer.page

Chapter (4) : Function

- أنواع الـ(function)

١) نوع يقوم المستخدم بتعريفه: (use defined function)

٢) نوع معرف في البرنامج (جاهز) : (Built – in function)

Built – in function: math function الإقترنات الرياضية

- قبل استخدام (math function) يجب تعريف المكتبة الخاصة بها : -

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

- الإقترنات الرياضية تتعامل مع (double variable) وليس (int) : -

```
Let  x: double
```

- Same of Math Function : -

1) ceil (x) : يقرب العدد إلى أكبر عدد صحيح

   cout<< ceil (9.2) ⟶ 10.0

2) Floor (x) : يقرب العدد إلى أصغر عدد صحيح

   cout<< floor (9.2) ⟶ 9.0

   cout<< floor (-9.8) ⟶ -10.0

3) Cos (x) : جتاس

   cout <<cos (0.0) ⟶ 1.0

   ↓ الزاوية (Rad)

4) Exp (x) : e^x

   cout<<exp(1.0) ⟶ 2.71 (e^1 = 2.71)

5) Fabs (x) : absolute value القيمة المطلقة

   cout<< fabs (-4.5) ⟶ 4.5

ملاحظة : abs (x)
يحول لقيمة مطلقة بدون عدد كسري
Cout<<abs(-4.5) ⟶ 4

6) Pow (x ,y) : X^y

   cout << pow (2,4) ⟶ 16.0 ($2^4 = 16$)

7) Sqrt (x) : $\sqrt{x}$

   cout << sqrt (9.0) ⟶ 3.0

facebook.com/Smurfs.On.The.Way

8) Fmod (x,y) : يعطي باقي قسمة (x) على (y)

cout << f mod (13.657 ,2.333) ⟶ 1.992

```
        5
2.333 | 13. 657
        11.665
        -------
        [1.992]
```

Ex : -

cout<< pow (2 , sqrt (9.0))

Pow(2,3) ➔ $(2^3 = 8)$ ⟶ 8

Use Defined Function :-

- في الإقترانات السابقة كنا نستخدم إقترانات معرف اسمها ووظيفتها مسبقا في مكتبة البرنامج، إما الآن فسنتعرف كيفية تعريف إقترانات جديدة.. نحدد اسمها ووظيفتها ونوع البيانات التي تستقبلها وترسلها ثم نقوم باستخدامها في البرنامج.
- يجب أن تحتوي الإقترانات المعرفة من قبل المستخدم على 3 عناصر:

1) Function prototype
2) Function call
3) Function definition

- وظيفة (function prototype) : - إبلاغ الـ (compiler) بوجود (function) في البرنامج من حيث اسمه ونوعه.
- وظيفة (function call) : - مكان استدعاء الـ (function) (اسم الـfunction).
- وظيفة (function definition) : تعريف الـ (function) [الوظيفة التي يقوم بها الـfunction].

Ex : -

```
#include <iostream.h>
int square  (int) ;          ⟶  1) function protype
int main ( )
{ int  x ;
cin>> x;
cout <<square  (x);          ⟶  2) function call
return 0 ;
}
int square (int y )          ⟶  هنا بدون فاصلة
{
return y*y ;                 ⟶  3) function Definition
}
```

يسمى main

www.muslimengineer.info
facebook.com/muslim.engineer.page

- سنقوم الآن بتعريف كل جزء من أجزاء الإقتران على حدى، ثم توضيح آلية عمله وإعطاء أمثلة متنوعة على الموضوع.
- يتكون (function prototype) من:

```
return value type      function name (parameters);
```

(double ,int, void , float)

- يقوم (return value type) بتحديد نوع القيمة الراجعة إلى (main):

مثال : int ⟵ نوع القيمة الراجعة integer (عدد الصحيح).

double ⟵ نوع القيمة الراجعة عدد غير الصحيح .

void ⟵ لا توجد قيمة راجعة .

- (function name) وهو اسم الاقتران الذي سوف يستدعى في الـ (main).

- (parameter) يقوم بتحديد نوع القيمة المرسلة إلى (function Defibition).

مثال : int نوع القيمة المرسلة integer (عدد الصحيح).

float نوع القيمة المرسلة عدد غير الصحيح .

void لا توجد قيمة مرسلة .

- الذي يميز الـ (function prototype) عن (Function Defintion) هو وجود فاصلة منقوطة (;).

Ex : -

1) int square (int) ;

عبارة عن اقتران اسمه (square) يقوم بإرسال واستقبال قيمة من نوع (integer).

2) int square (void); ≡ int square ( );

عبارة عن اقتران اسمه (square) يقوم باستقبال قيمة من نوع (integer) ولا يرسل أي قيمة.

3) void square (int) ;

عبارة عن اقتران اسمه (square) يقوم بإرسال قيمة من نوع (integer) ولا يستقبل أي قيمة.

4) void square (void);

عبارة عن اقتران اسمه (square) لا يقوم بإرسال ولا استقبال أي قيمة .

facebook.com/Smurfs.On.The.Way

- Function cell :

نقوم باستدعاء الـ (function) عن طريق اسمه بإضافة إلى parameter

Square (x)

اسم الاقتران ← Square ، parameter ← (x)

- ملاحظة : -

١) عندما نعرف الـ (function prototype) على أنه لا يوجد قيمة راجعة:

```
void square (int);
```

ففي هذه الحالة لا يجوز استدعاء الـ (function) باستخدام أمر الطباعة (cout) ولكن يتم استدعاؤه لوحده: square (x)

٢) عندما نعرف الـ (function prototype) على انه لا يوجد قيمة مرسلة:

```
int square (void) ;
```

ففي هذه الحالة لا يوجد (parameter) في الـ (function): cout << square ( ) ;

- آلية الحل : -

1) نقوم بالتأكد من وجود (function) في البرنامج من خلال (function prototype) المعرف قبل int main ;.

2) نبدأ حل البرنامج بالخطوات العادية إلى ان نصل إلى (function call):

```
cout<< square (x) ;
cout<< square (10) ;
```

٣) عندما نصل إلى (function call) نأخذ قيمة الـ (parameter) (10) ونرسلها إلى (function definition) ونخزنها في الـ(parameter) التابع لـ(function definition):

int square (int y) ➔➔ function definition

ليصبح int y = 10

٤) ندخل إلى (function definition) وننفذ الجمل الموجودة فيه وعندما نصل إلى (return) نُرجع القيمة المجاورة لـ (return ◯;) إلى محل استدعاء الـ (function) في الـ (main).

- في المثال السابق return y*y ;

Y*y =10 *10 =100 ➔➔➔ return (100) ;

- أي أننا نلغي الـ (function) في الـ (main) ونضع بدلا منه القيمة الراجعة المجاورة لـ (return).

- ملاحظة : - إذا كان الـ (function) لا يُرجع قيمة ((void square (int)) فلا يوجد (return) في الـ(function definition).

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex : -

```
#include <iostream.h>
int square (int) ;
void main ( )
{for (int x=1 ; x<= 10 ; x++)
cout<< square (x) << "  " ;
cout<< endl;
}
int square (int x)
{return x*x ;
}
```

يستقبل ويرسل بيانات من نوع (int)

1 4 9 16 25 36 49 64 81 100

ملاحظة :- المتغير (x) المعرف في الـ (main) ← local in main

يختلف عن المتغير (x) المعرف في الـ (function) ← local in function

Ex : -

```
#include <iostream.h>
int addition (int ,int );
void main( )
{ cout<< addition (5,3);
}
int addition (int a , int b )
{
int x;
X=a+b ;
return x ;}
```

- يستقبل من الـ(main) قيمتين من نوع (int).
- وجود int في بداية الـ (prototype)، يعني يرجع قيمة، يعني وجود (return) في النهاية.

8

ملاحظات :

١) الترتيب مهم في الـ (parameter) في (function prototype).

```
int x ( double , float , char ) ;
cout<< x ( a , b, c)
```

٢) يجوز الاستغناء عن الـ (function prototype) وفي هذه الحالة نعرف الـ(function definition) كاملاً قبل الـ (int main)، كما في المثال التالي:

facebook.com/Smurfs.On.The.Way

Ex : -

```
#include <iostream.h>
int square  (int y )
{
return  y*y ;
}
int main ()
{
cout<<square (5.5) ;
return 0;
}
```

Function definition

main

لايوجد فاصلة منقوطة ;

نبدأ الحل من هنا

25

int y=5 ✓ int y=5.5 ☒

٣) يجب ملاحظة الترتيب :-

Function prototype → int add (int , int);

Function call → cout<< add (x ,y)

Function definition → int add (int a, int b)

☒ int add (int a,b) is Error .

٤) في الـ (function definition) : -

1) void square (int y)

لا يوجد return

2) void square (int y)

```
{
return ;
}
```

ليس (error) وذلك بسبب عدم وجود قيمة راجعة

٥) إذا كان الاقتران (void) يتم استدعاء الاقتران عن طريق كتابة اسمه وقيمته :

مثال : x(7);

ولا يوجد طريقة أخرى الاستدعاء الاقتران.

Ex : -

```
void x (int y)
{ cout<< "school" << endl;
}
void main  (  ) {
cout<<x(7) ; }
```

(error)

لا يجوز الاستدعاء عن طريق الطباعة

www.muslimengineer.info
facebook.com/muslim.engineer.page

٦) إذا لم يحدد نوع القيمة الراجعة تعتبر (int).

Ex :-

```
square (int y)
{
return y*y ;
}
```

(int) تعتبر

Ex : -

```
# include <iostream.h>
int maximum (int ,int ,int ) ;
void main ( )
{ int a ,b ,c ;
cout << "enter three integers " ;
cin>> a>> b>>c ;          // a=22 , b =85 , c =18
cout <<" maximum is :" << maximum (a , b , c ) ;  }
int maximum (int a , int b , int c )
{int max =a ;
if (b >max )
max =b ;
if ( c > max )
max =c ;
return max ; }
```

```
Enter  three integer
22      85     18
Maximum is  : 85
```

Function prototype : وظيفة (يجب وضع فاصلة منقوطة ;)

1) Name of function.
2) Type of data returned by the function.
3) The number of parameter.
4) The type of parameter and the order in which there parameter are expected .

Note : -

Defining a function parameter again as a local variable in the function is a syntax error .

Ex :-

```
void max (int x , int y , int z )
{
int x ;
}
```

سوف يعطي البرنامج (error) لأنه لا يجوز تعريف المتغير أكثر من مرة في نفس الـ(local) مرة في الـ main ومرة في الـ function ,

facebook.com/Smurfs.On.The.Way

Header files :

1) #include <iostream. h>
2) #include <math.h>
3) #include <stdlib.h> ⟶ :random [0-32767]

Ex : -

```
# include <iostream.h>
#include <stdlib.h>
void main ( )
{ int I ;
I=rand ( ) ;
cout <<I ;
}
```

رقم عشوائي

- في كل مرة تعمل فيها (run) للبرنامج يطبع رقم عشوائي مختلف .

Ex : -

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

برنامج لتحديد رقم يدخله المستخدم
إذا كان فردي أو زوجي

www.muslimengineer.info
facebook.com/muslim.engineer.page

Scope Rule : هو عبارة عن وضع أوامر معينة داخل { } داخل البرنامج

Ex : -

```
#include <iostream.h>
void a (void) ;
void b (void) ;
void c (void) ;
void d (void) ;
int x =1 ;
int main ( )
{ int x =5 ;
cout<<x<<endl;
{int x =7 ;
cout <<x; }
cout <<x ;
a( ) ; 25  26
b ( ) ; 50  51
c ( ) ; 1    10
a ( ) ; 25  26
b ( ) ; 51  52
c ( ) ; 10   100
d ( )  ;100
cout << x<<endl; }
void a (void )
{ int x =25 ;
cout <<x ;
++ x;
cout << x << endl;}
void b (void)
{ Static int x =50;
cout<<x ;
++x ;
cout << x;}
void c (void)
{cout<<x;
X*=10;
cout<<x<<endl;}
void d (void )
{cout<<x ;}
```

الإقترانات

- تعريف المتغير قبل (int main) يسمى (global variable) ويتعرف عليه كل البرنامج (functiom + main).
- عند وجود (local) و (global) وبنفس الإسم تكون الـ(local) أعلى أولوية (تنفذ)

New scope

Function call

int x =1 ; → (global variable) يسمى

cout<<x<<endl; → 5

{int x =7 ; cout <<x; } → 7

cout <<x ; → 5

cout << x<<endl; } → 5

{ int x =25 ; → Local in function

cout <<x ; → 25

++ x; → 26 تصبح (x):

cout << x << endl;} → 26

{ Static int x =50; → Local in function

cout<<x ; → 50

++x ; → 51 تصبح (x):

cout << x;} → 51

{cout<<x; → 1 Global (x=1)

X*=10; → تصبح (x) تساوي (10)

cout<<x<<endl;} → 10

{cout<<x ;} → 100 (Global)

facebook.com/Smurfs.On.The.Way

ملاحظات :-

1)

static	constant
لا يجب إعطاء قيمة ابتدائية وتعتبر (صفر)	يجب إعطاء قيمة ابتدائية
يمكن تغير القيمة	لا يمكن تغير القيمة

٢) (Global) و (Static) تحافظ على أخر قيمة مخزنة فيها عند استدعائها أكثر من مرة .

٣) أولويات التنفيذ: Global < Local < new scope

٤) void a (void) ;

void ← داخل الـ (function definition) لا يوجد (return)

(void) ← عند استدعاء الـ (functiom) داخل الـ (main) لا يوجد (parameter)

a ( ) ;

No parameter

Recursion:

int ⟶ ± 32768

Long int ⟶ ± 2147483647

Unsigned long int ⟶ (0 – 4294967294)

- Unsigned long int = unsigned long طول الفترة متساوي

Ex : -

لإيجاد مضروب أي عدد

مضروب الـ 4 يساوي= 4*3*2*1

```
Long int num ;
cin >>num ;
int fact =1 ;
for (int counter =num ; counter > = 1 ; counter-- )
Fact* = counter ;
```

❖ فكرة عمل الـ (Recursion) : -

4!=4*3!

3*2!

2*1!

1*0!

0!=1 ⟶ (base case)يسمى

1*1=1

2*1=2

3*2=6

4*6=24

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex :-

```
#include <iostream.h>
Unsigned long int function (unsigned long int ) ;
int main ( )
{for (int i=0 ; i<=10 ; i++)
cout<< i<<factorial(i) <<endl;
return 0 ;}
Unsigned long factorial (unsigned long num )
{if (num <=1)
return 1 ;
else
return num * factorial (num + 1)}
```

❖ التوضيح : -

الرقم المدخل	المضروب
0	F(0)=1
1	F(1)=1
2	F(2)=2*factorial(1) =2*1=2
3	F(3)=3*f(2)=3*2=6

F (n)!=n* f(n-1)!

Ex: -

10+9+8+7+.....+0

$$\sum 10 = 10 + \sum 9$$

$$\sum 9 = 9 + \sum 8$$

```
int sum (int n)
{if (n==0)
return 0;
else return n+sum (n-1)
}
```

❖ ملاحظة : -

١) في الـ(recursion) يوجد اثنان (return) في الـ(function definition)

٢) في الـ(recursion) يتم استدعاء الـ(function) مرتين : -

a) في الـ (main).

b) في الـ (function definition).

facebook.com/Smurfs.On.The.Way

Reference and reference parameter:

- الـ(reference)(المرجع)هو عبارة عن متغيرين مرتبطين معاً فإذا تغير المتغير الأول تغير المتغير الثاني تلقائياً والعكس صحيح .
- إشارة الـ(reference) نعتبرها هنا هي : (§)

مثال : §Y =X

Y is reference to x

Ex : -

```
#include <iostream.h>
void main ( )
{ int x =3 , §y =x ;
cout<< x<<y ;
cout<< endl;
Y=8;
cout<< x<< y;
}
```

```
33
88
```

- ملاحظات : -

١) بما أن (Y) مرجع لـ (x) فإن قيمة (Y) و(x) متساوين.

٢) عند تغير إحدى المتغيرين يتغير الأخر تلقائياً.

٣) يجب أن يعطى المتغير المستقل (ليس المرجع) قيمة ابتدائية:
نجد في المثال السابق المتغير المستقل هو (x) وقد أعطي قيمة ابتدائية مقدراها (3).

٤) يجب أن يكون المتغير المستقل (x) و المرجع (y) من نفس النوع.
إذا كان (x) من نوع (int) يجب ان يكون (y) من نوع (int) وهكذا..

Ex :-

```
#include <iostream.h>
void main ( )
{  int x=3 , §y ;
cout<<x<<y;
Y=8 ;
cout<< x<<y;
}
```

→ Error (Reference variable (y) must be initialized)

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex :-

```
#include<iostream.h>
int  SquareByValue (int);
void SquareByReference (int§);
void main ( )
{ int x =2 , z=4 ;
cout<< x<< SquareByValue (x) ;
cout<<x
cout<<z
Square By Reference (z) ;
cout<<z;}
int SquareByValue (int  a)
{ return a*=a;
}
void Square By Reference (int § cref)
{
Cref *=cref
}
```

```
242416
```

ملاحظات : -

في الـ (functions) يوجد نوعين هما : -

a) Call by value
b) Call by reference

في النوع الأول (call by value) نأخذ الـ (parameter) من الـ (function) ونرسل نسخة عنه (لا نرسل الأصل) لذلك أي تغير على قيمة الـ (parameter) في الـ (function definition) لايُأثر على قيمةالـ(parameter) في الـ(main)

في النوع الثاني (call by reference) نأخذ الـ (parameter) من الـ (function) ونرسل الأصل إلى (function definition) لذلك أي تغير على قيمة الـ (parameter) في (function definition) سوف يُأثر على قيمة الـ (parameter) في الـ (main).

سؤال: كيف نعرف ان الـ (function) هو (call by value) أم (call by reference)؟

الـ (function) من نوع (call by reference) يحتوي الـ (parameter) لديه على إشارة §.

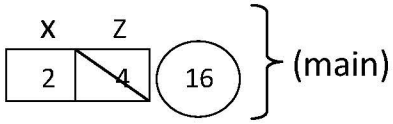
facebook.com/Smurfs.On.The.Way

توضيح المثال السابق:

(function call) square by value (x) ⟶ square by value (2);

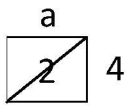
(function definition) int square by value (int a)

int a=2



return a*=a ➔ a=a*a ➔ a=2*2=4

أي تغير على قيمة (a) لا يتأثر قيمة (x)

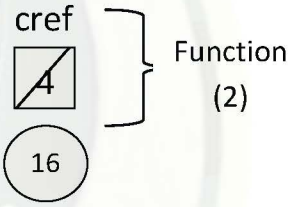


أصبحت (a) تساوي (4) لكن قيمة (x) بقيت (2).

(function call) square by reference (Z) square by reference (4);

(function definition) int square by reference (int §cref)

int §cref =z=4



أي تغير على قيمة (cref) سوف يغير قيمة (z)

Return cref*=cref ➔ cref=cref*cref ➔ cref=4*4=16

أصبحت قيمة (cref) (16) وتلقائياً أصبحت قيمة (z) (16)

www.muslimengineer.info
facebook.com/muslim.engineer.page

Static array:

- تعلمنا سابقاً إن جملة (static) لها خاصيتين :

١) تحافظ على أخر قيمة مخزنة فيها عند استدعائها أكثر من مرة .

٢) إذا لم تعطى قيمة ابتدائية تعتبر (صفر).

Ex:-

```
#include <iostream.h>
void staticArray1(void);
void Autoarray2 (void);
void main ( )
{StaticArray1( );
 AutoArray2( );
 StaticArray1( );
 AutoArray2( );
}
void staticArray1(void)
{static int array 1[3];
for (inti=0 ; i<3 ; i++)
cout<< array [i]<< "     ";
for (i=0 ; i<3;  i++)
cout<<(array [i]+=5)<<"     ";
}
void AutoArray2(void)
{int I; array2[3]={1,2,3};
for (i=0 ; i<3;i++)
cout<<array2[i]<<"     ";
for(i=0 ; i<3; i++)
cout<<(array2[i]+=5)<<"     ";
}
```

- التوضيح: -

- قمنا باستدعاء الـ (function) : - static Array1( )

نذهب إلى (function definition) ونلاحظ تعريف المصفوفة من نوع (static) حجمها (3).

	0	1	2
Array 1	0	0	0

- ونلاحظ أن المصفوفة لم تعطى قيمة ابتدائية لذلك تعتبر (صفر) لانها static

facebook.com/Smurfs.On.The.Way

- تقدم جملة (for) الأولى بطباعة محتويات المصفوفة :

ناتج الطباعة ⟵ 0 0 0

- تقدم جملة (for) الثانية بزيادة مقدار (5) على كل عنصر من عناصر المصفوفة وطباعتها :

ناتج الطابعة ⟵ 0 0 0

- نرجع إلى الـ(main) وقمنا باستدعاء الـ (function) : AutoArray2( )

نذهب إلى (function definition) ونلاحظ تعريف مصفوفة وحجمها (3)

	0	1	2
Array2	1	2	3

- تقوم جملة (for) الأولى بطباعة محتويات المصفوفة :-

ناتج الطباعة ⟵ 1 2 3

- تقوم جملة (for) الثانية بزيادة المقدار (5) على كل عنصر في المصفوفة وطباعتها :-

ناتج الطباعة ⟵ 6 7 8

- نرجع إلى (main) وقمنا باستدعاء الـ(function) : static Array ( )

وهنا تبدأ المصفوفة بالقيمة المخزنة مسبقاً

	0	1	2
Array 1	5	5	5

ونقوم بالزيادة عليها بمقدار (5) :-

	0	1	2
Array1	10	10	10

- نرجع إلى (main) وقمنا باستدعاء الـ (function) Auto Array 2 ( )

وهنا تبدأ المصفوفة من القيم المخزنة من البداية تعريف البرنامج.

	0	1	2
Array2	1	2	3

وبعد الزيادة

	0	1	2
Array2	6	7	8

www.muslimengineer.info
facebook.com/muslim.engineer.page

Ex : -

```
#include<iostream.h>
int main (  )
{cons tint array 1[3] ={10,20,30};
Array1[0]+=10;
Array1[1]*=20;
Array1 [2]-=5;
for (int i=0 ; i<3;i++)
cout<<array[i];
return 0;
}
```

Error (const) لا يمكن التعديل على قيم متغير من نوع

Ex:-

```
#include <iostream.h>
void modifyArray (int [ ] ,int );
void modifyElement (int);
void main ()
{ cons tint arraysize =5;
inti , a [arraysize}={0,1,2,3,4};
for (i=0 ; i<arraysize ; i++)
cout<< a[i];
cout<<endl;
Modify array (a,arraysize);
for (i=0 ; i<arraysize ; i++)
cout<<a [i];
cout<<a[3];
Modify Element (a[3]);}
 void modifyarray (int b[  ] , int size)
{ for (int j=0; j<size ; j++)
B [j]*=2;
}
void modifyElement (int e)
{   cout<< (e*=2);
}
```

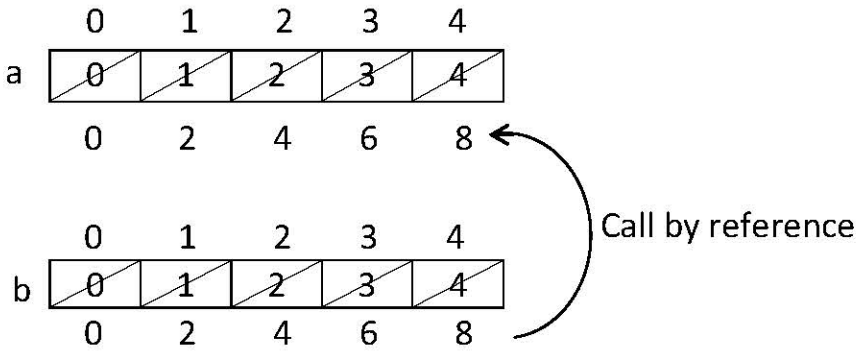
Call by refrance

طباعة العنصر الرابع من المصفوفة (a)

Call by value

```
01234
02468
6
12
```


facebook.com/Smurfs.On.The.Way



- عند استدعاء اسم المصفوفة فقط modifyArray (a , arraysize) تعتبر ← call by reference
- عند استدعاء اسم المصفوفة + عنصر فيها modify Element (a[3]) تعتبر ← call by value

Pointers:

- Pointers contains address of variable that has specific value

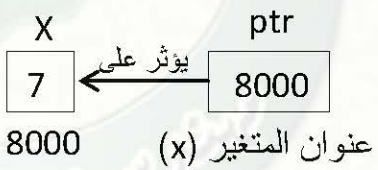
  int *prt ;

  Prt is a pointer to integer value.

Ex :-

```
int x=7;
int *prt =§x ;
```

عنوان المتغير(x)



- عملية تخزين عنوان المتغير (x) في المتغير (prt).
- يجب أن يكون كل من المتغير وعنوانه من نفس النوع :

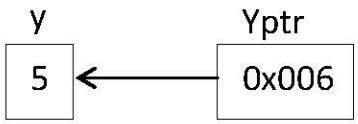
```
int y=5;
int *x =§y;
```

- نعرف ان المتغير عبارة عن مؤشر من خلال الاشارة (x) .

Ex : -

```
#include<iostream.h>
void main ()
{ int y =5 ;
int *yptr ;
Y ptr=§y;
}
```

١. قمنا بتعريف المتغير(y) وخزن فيه المقدار (5)
٢. قمنا بتعريف المؤشر (y ptr)
٣. قمنا بتخزين عنوان المتغير (y) في المؤشر (yptr)



www.muslimengineer.info
facebook.com/muslim.engineer.page

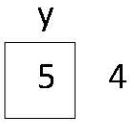
- لو كان في البرنامج الأمر التالية: -

1) cout<< y; ⟶ 5
2) cout<<*yptr; ⟶ 5
3) cout<<yptr; ⟶ 0x006
4) cout<<y+*yptr ; ⟶ 10

- ملاحظة: لطباعة المحتوى العنوان نستخدم (*yptr)
ولطباعة العنوان فقط نستخدم (yptr) من دون (*)

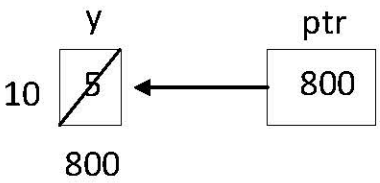
- ملاحظة : لتغير محتوى المتغير هناك طريقتين :
١) الطريقة المباشرة :

Ex :-
```
int y=5;
Y=4 ;
```



٢) الطريقة غير مباشرة :باستخدام المؤشر

Ex:-
```
int y=5;
int*prt=§y;
*prt=10;
```



Ex :-
```
#include<iostream.h>
void main (    )
{ int a;
int *ptr;
A=7;
Ptr=§a;
cout<<§a;
cout<<ptr;
cout<<*ptr;
}
```

⟶ تخزين عنوان (a) في (ptr)
⟶ طباعة عنوان (a)
⟶ طباعة عنوان (a)
⟶ طباعة محتوى الاقتران

```
Any address
Any address
7
```


facebook.com/Smurfs.On.The.Way

Ex :-

```
void main ()
    { int x, y;
    int *p=§x ;int*q=§y
    *p=35; *q=98; *p=*q;
    cout<<x<<"     "<<y<<endl;
    cout<<*p<<"    "<<*q<<endl;
    }
```

98	98
98	98

التوضيح

1) x y

2) p q (*p=§x / *q=§y)

3) *p =35 , * q =98

محتوى العنوان P=(35) X=98 لان(P) يؤشر على(x)

محتوى العنوان q=(98) Y=98 لان (q) يؤشر على (y)

x	y
354	98

4)*p=*q ⟶ تخزين محتوى العنوان (q) ⟵ (98) في محتوى العنوان (p)

X=y ➔ y=98 , x=98

www.muslimengineer.info
facebook.com/muslim.engineer.page

- Ex:-

```
int num=9;
int *ptrnum= §num
cout<<num;          ⟶  9
cout<<ptrnum;       ⟶  Any address
cout<<*ptrnum ;     ⟶  9
```

- Ex:-

```
int x =10 , y=20 ;
int *ptrx=§x , *ptry=§y;
Ptr x= ptr y;
*ptr x++;
cout<<y ;
cout<<§ *ptr;
cout<<*§
```


facebook.com/Smurfs.On.The.Way

أسئلة مقترحة وسابقة لمادة الفيرست مع الحلول

www.muslimengineer.info
facebook.com/muslim.engineer.page

Programming c++

```
#include <iostream.h>
void main ()
{
int t;
 if ( t ==1)
        cout<<"jan"<<endl;
 else if ( t==2)
        cout<<"feb"<<endl;
 else if ( t==3)
        cout<<"march"<<endl;
 else if ( t==4) cout << "april"<<endl;
}
```

t=1 jan | t=2 feb | t=3 mar | t=4 april

Configuration: samer22 - Win32 Debug----------
Compiling...
samer22.cpp
C:\Documents and Settings\samer\My Documents\samer22.cpp(5) : warning C4700: local variable 't' used without having been initialized

samer22.obj - 0 error(s), 1 warning(s)

```
"C:\Documents and Settings\samer\My Do
-858993460
Press any key to continue
```

Why did the compiler give (-85899……etc) number ????? rubbish data
Cout var has no value.

Conditional operator

- ✓ The conditional operator can be written as ? : (ternary operator)
- ✓ Expression1 ? expression2 : expression 3

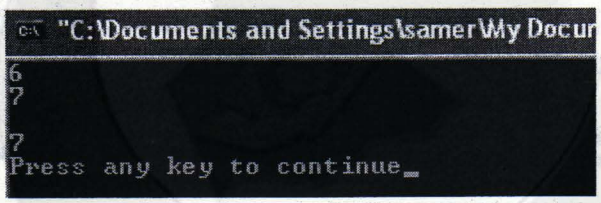
Example 1

If (a >= b) max = a ;
else max = b ; ==========➔ max = (a>=b) ? a : b ;

Exmple 2

```
#include <iostream.h>
void main ()
{
int a,b,max;
cin >> a>>b;
cout<<endl;
max=(a>=b)||(a<6) ? a : b ;

cout << max << endl;
}
```



Switch structure

✓ **General syntax of the switch statement is :**

```
Switch ( expression )
{
Case value 1:
Statement 1
Break;
Case value 2:
statement 2
break;
 .
 .
default :
statements
}
```

Switch case analysis :

1. test the expression , if the value for case 1 , then apply statement 1 ….break .
2. if the test of the expression for case 1 is false , go to test case 2 …..
3. if there is no case is true , then the compiler will execute the default case …

example 1

```
#include <iostream.h>
void main ()
{
int t;
cin >>t;
 switch (t)
 {
 Case 1:
  cout<<"jan"<<endl;
 break;
 case 2:
  cout<<"feb"<<endl;
  break;
  case 3 :
 cout<<"march"<<endl;
 break;
 case 4:
 cout << "april"<<endl;
 break;
 default :
 cout <<" error"<< endl;
 }
 }
```

just int → char / Boolear

error capital

⟹ Const int x = 3 ; ⟹ Here x always = 3

x++ ⟹ error

x = x+3 ⟹ error

cin >> x ⟹ error

⟹ int x = 3 ;

cout << 10 / (double)x ; ⟹ 3.33 } explicit data Conversion

double(x)

⟹ char x = 'A' ;

cout << x << x+ ⟹ A 68 } implicit data ...sion

facebook.com/Smurfs.On.The.Way

What is the output:

```
#include <iostream.h>
void main()
{
int a=5;
int b=8;
b=((a/4 || a>5)?a--:--a);
cout << b <<endl<< a<<"\n";
}
```

```
5
4
```

```
#include <iostream.h>
void main()
{
int a=5;
int b=8;
b=((a/4 || a>5)?--a:++a);
cout << b <<endl<< a<<"\n";
}
```

```
4
4
```

```
#include <iostream.h>
void main()
{
int a=5;
int b=8;
b=((a!=5 || a>5)?--a:++a);
cout << b << a<<"\n";
}
```

```
66
```

```
#include <iostream.h>
void main ()
{
int t;
cin >>t;
 switch (t)
{
case 1:
 cout<<"jan"<<endl;
case 2:
 cout<<"feb"<<endl;
 break;
 case 3 :
cout<<"march"<<endl;
break;
} }
```

```
t = 1
jan
feb
=========
t = 2
feb
=========
t = 6
nothing
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

```
#include <iostream.h>
void main ()
{
char z= 'c';
switch(z){
case 'n' :
cout<<"you are welcome\n";
case 'q':
case 'p':
case 'c':
cout<<"in\n"; cout<<"ok\n";break;
cout<<"balqa applied university"; break;
default:
cout<<"jordan";
}
 }
```

```
in
ok
```

```
#include <iostream.h>
void main ()
{
int b=4,c=23.9;
int f=6.1;
cout<<c/f<<"\n";
cout<<b%5*4-6*2;
```

```
3
4
```

```
#include <iostream.h>
void main ()
{
double s=4.77;
cout<< s<<endl;
cout<< s++<<endl;
cout<< ++s<<endl;
cout<< s--<<endl;
cout<< --s<<endl;

 }
```

```
4.77
4.77
6.77
6.77
4.77
```

```
cout << s+=4
cout << s-=5

cout<< z%v<<endl;
cout<< v%z<<endl
```

```
No answer
```


facebook.com/Smurfs.On.The.Way

```
#include <iostream.h>
void main ()
{
int s=6.77,t=4.6,z=3;
double v=3.6;
cout<< t<<t%s<< endl;
cout<< z%t<<endl;
cout<< z/v<<endl;


cout<< z%v<<endl;
cout<< v%z<<endl

 }
```

```
44
3
0.83333

ERROR
ERROR
```

```
#include <iostream.h>
void main ()
{
if ( 6<2*5)
cout <<"HI";
cout <<"ALL";
 }
```

??

** static_cast<int> (…….)

Example :

- ✓ Static_cast <int>(7.9) = 7
- ✓ Static_cast <int>=3
- ✓ Static_cast<double>(25)=25
- ✓ Static_cast<char>(65)='A'
- ✓ Static_cast<int>('8')=56.

---

www.muslimengineer.info
facebook.com/muslim.engineer.page

```
double x=2;
x=10/4.0 + 9*x;                    20.5

cout<<x;
```

---

long z=3.2; (long alone without any datatype means long int) so 3 -1=2

cout<<z-1 ; **2**

```
int x=10;
const int y=5;
cout<<y+5;                         10no
if(y==x)
cout<<"yes";
else
cout<<"no";
```

```
# include<iostream.h>
    void main()
     {      int x=3,y=2, z=4;
      switch(x+y++)
      {
      case 1:
      cout<<"  "<<".";
      break;
     case 5:
     cout<<++y<<z++<<"\n";
     case 4: cout <<"break";
      case 2: cout<<x%z<<"\t";
      default:
      cout<<"Done";
      break;          }
     }
```

44 **Break 3** **Done**

MEng.Samer A.Hamed

```
# include<iostream.h>
    void main()
     {
int z=8,a=10,A=9;
if (z>=A && z/2)

{
cout <<a;
 a++;
 A++;}
else if (z<a && z/3)
 if (a/A || a%A)
 cout<<A;
     }
```

9

Pre- First Exam solution

```
#include <iostream.h>
void main ()
{
int x=15, y=3;
cout<<x/y;
cout<<--x*y;
cout<<++x;
```

54215

```
int x = 3, z = 5, k=1 ;
      k=++x - z--;
      cout<<z;
```

4

```
int m=3, n=14;
if(m>n)
cout<<"A"<<endl;
else
cout<<"C";
cout<<"D";
```

CD

```
float x=14;

cout<<x%3;
```

ERROR

```
int a = 6,z=2;
z += a++;
z++;
cout << z;
```

9

MEng.Samer A.Hamed

facebook.com/Smurfs.On.The.Way

```
int n = 10;

    if(n)
    n = n + 2;
    if(n)
    n = -n;
    n = n + 12;
    cout<< n;
```

0

```
int x=7, y=18;

cout<<x y;
```

ERROR

```
int a =3, b= 2,c = 5;

if (a > b)

  a = 4;

if ( b > c)

   a = 5;

else

  a = 6;

  cout << "a" << endl;
```

a

```
  cout<<"Hi \\ "<<"cout<<\"Hi\";";
```

Hi \ cout<<"Hi";

```
if( 3%6)
cout<<3;
else
cout<<6;
```

3

MEng.Samer A.Hamed

www.muslimengineer.info
facebook.com/muslim.engineer.page

```
int n=13;

if(n>4.0)
cout<<n/2;
else
cout<<n*2;
cout<<n+2;
```

615

```
int x=10;

if(x=>9){

cout<<"Hello"; }

else

cout<<"Bye";
```

ERROR

```
int x = -3, y = 4, z = 5;

        if (x < 0)

    if (y > 0)

    z = 2;

     else

    z = 3;

   cout<< z;
```

2

```
char c= 'a';

switch(c){

case 'a' :cout<<"0";

case 'b':case 'c':case 'd':

cout<<"1";break; cout<<"2";

cout<<"3"; break;

default: cout<<"4";}
```

01

```
#include <iostream.h>
void main ()
{
if ('a'>'b'||66>static_cast<int>('A'))                    #*#
cout<<"#*#"<<endl;
 }
```

```
#include <iostream.h>
void main ()
{
if (5<3)
cout<<"*";
else                                                     $
if (7==8)
cout<<"&";
else
cout<<"$";
 }
```

```
#include <iostream.h>
void main ()
{int n;
cin>>n;
switch (n%6)
{                                          n=6 → the output is space...
case 1:                                    n=8 → the output is 8
case 2:
case 3:
case 4:
case 5:
    cout<<n;
    break;
case 0:
    cout<<endl;
    break;

 }
}
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

```
#include <iostream.h>
void main ()
{
    int a;
cin>>a;
if(a>0)
switch (a)
{
case 1:a=a+3;
case 3:a++;
    break;
case 6:a=a+6;
case 8:a=a*8;
    break;
default:
    a--;
}
else
a=a+2;
cout<<a;
}
```

```
#include <iostream.h>
void main ()
{
    const int a=5
            int g=5;
            ;
   cin>>a;
    a++
    a=g+2 \\ i cannot do any changes on a
}
```


facebook.com/Smurfs.On.The.Way

أسئلة مقترحة وسابقة لمادة السكند مع الحلول

www.muslimengineer.info
facebook.com/muslim.engineer.page

```
#include<iostream.h>
void main(){
        int f=3;
        do{f=f*4;}while (f<=20);
        cout<<f;
}
```

48

For the following loop , how many times the "all" word will be printed:

```
#include<iostream.h>
void main(){
        int f=3;
        do{cout<<"all";
        f++;}
        while (f<=9);
}
```

7 times

```
#include<iostream.h>
void main(){
        int f=6;
        while (f>=3) {
                cout<<2*f<<"";
                --f;   }
```

12 10 8 6

```
#include<iostream.h>
void main(){
        int a[5]={3,2,6};
        for (int i=0;i<4;i++)
        a[a[i]]=a[i];
        for(i=0;i<4;i++)
            cout<<a[i]<<' ';  }
```

3 2 2 3

```
        #include<iostream.h>
void main(){
        int a[9]={7,5,6,4,2};
        int i=2;
          int x=a[i+3];
        int y=a[i]+3;
        cout<<x<<" "<<y; }
```

0 9

```
#include<iostream.h>
void main(){
    int x=6;
    do{ x=x-1;
    cout<<x<<" ";
    }
    while (x>=2); }
```

5 4 3 2 1

```
#include<iostream.h>
void main(){
        int x[5]={4,5,3,5,7};
        for (int i=1;i<=3;i++){
                for (int j=1;j<=3;j++)
                        cout<<++x[j]<<" ";
                cout<<endl; } }
```

6 4 6

7 5 7

8 6 8

```
#include<iostream.h>
void main(){
                int a;
                for (a=3;a<5;a--) {
                        if (a<2)  break;
                        else
                if (a==3)   continue;   a=-1; }
                cout<<a;   }
```

-2

```
{ int  a=5;   int b=3;
while (b<9)   {
        int c=3;   b+=c++;  b++;    cout<<c;
        break;}
a=3*b;
cout<<b<<a; }
```

4 7 2 1

```
  #include<iostream.h>
void main(){
  int a[]={2,3,3,4,4,3,2,2,1};
  int b[]={5,6,7,8,2,3,5,2,1};
  char c[]="OneMoreTry";
  for (int i=0;i<5;i++)
  { int d=a[i];
  int e=b[d];
  cout<<c[e];  }}
```

Trree

```
#include<iostream.h>
void main(){
    for (int i=1;i<=9;i+=3);
        cout<<i-1 ;    if(i>=9)        cout<<i;  }
```

9 10

```
#include<iostream.h>
void main(){
        for (int i=1;i==6;i++)
        {cout<<"*";
        i+=2;}}
```

Nothing

```
#include<iostream.h>
void main(){
        int i=0,z=2,c=5;
        while (i<10)
        { i+=z;
        switch (i)
        {
        case 2:c=c+1;
        case 4:c=c+2;
        default :c=c+3;
}}
cout<<c;
}
```

25

ARRAYS I

✓ **Initializing array:**

1. **int a[4]= {1,3,2,4};**

hint: int [4]={3} , here the a[0]=3 and (a[1],a[2],a[3])=0

2. **int a[4];**
for (i=0;i<=3;i++)
cin>> a[i];

3. **int a[4];**
for (i=0;i<=3;i++)
a[i]=2+I;

4. **a[0]=2, a[3]=9 …….etc**

5. **int [ ]= { 1,2,3 };**

---

printing

✓ **int a[5]={1,3,4,6,7};**
cout << a[0]<< a[1]<< a[2]<<a[3]<< a[4];

1 3 4 6 7

```
int a[5];
    for (int i=0;i<=4;i++)
        cin>> a[i];
    for (i=0;i<=4;i++)
        cout<<a[i];
```

طباعة عناصر المتجه

```
#include <iostream.h>
void main()
{
    int a[4]={8,2,5,9};
        for (int i=0;i<=3;i++)
    {
        for (int j=0;j<=3;j++)
            cout<<++a[j]<<" ";
    cout<<endl;
    }
}
```

```
9  3  6  10
10  4  7  11
11  5  8  12
12  6  9  13
```

String array :

\0 : stores the null character.

Example:

```
char a[5]={'a','b','u','m','\0'};
cout<<a;
```

abum

```
char a[5]={'a','b','\0','m','u'};
cout<<a;
```

ab

```
char a[5]={'\0','b','a','m','u'};
cout<<a;
```

nothing

multidimensional arrays

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Column subscript
Row subscript
Array name

- A multidimensional array can be initialized in its declaration much like a one-dimensional array.
- The values are grouped by row in braces.
- If there are not enough initializers for a given row, the remaining elements of that row are initialized to 0.

```
int list
[4][3]={{2,3,1},{15,25,13},{20,4,7},{11,18,1
4}};
```

```
int a [2] [2]= {1,3,4,5};
int a [2] [2]= {1,3,4,5,6};
int a [2] [2]= {1,3,4};
      ##################################
int list [3][3]={{1,2,3},{1,1,1},{0,0,0}};
int s1=0,s2=0;
for(int i=0;i<3;i++)
for (int j=0;j<3;j++)
if(i!=j)
s1+=list[i][j];
else
s2+=list[i][j];
cout<< s1<<" "<<s2;
```

```
int a[5][4];
for (int i=0;i<=4;i++)
for (int j=0;j<=3;j++)
cin>> a[i][j];
for (i=0;i<=4;i++)
for (int j=0;j<=3;j++)
cout<<a[i][j];
```

????

شرح وتمارين سامح حامد

Switch selection :-

* switch case analysis :

1. Test the expression ... If the value for case 1, then apply statement 1 ... break

2. IF the test of expression for case 1 is false, go to test case 2 .....

3. If there is no case is true, then the compiler will execute the default case...

Example 1

```
#include <iostream.h>
void main( )
{ int t;
cin >> t;
switch (t)
{
case 1:  cout << "jan" << endl;
break;
case 2:
cout << "feb" << endl;
break;
default :
cout << " error" ;
}}
```

t = 1
jan

t = 2
feb

t = 3
error

facebook.com/Smurfs.On.The.Way

Example #2

```
int a;
cin >> a;
if (a>0)
switch (a)
{
  case 1:   a = a+3;
  case 3:   a++;
  break;
  case 6:   a = a+6;
  case 8:   a = a*8;
  break;
  default :
   a--;
}
else
a = a+2;
cout << a;
}
}
```

a=1 5

a=2 1

?

a=3 4

a=6 96

a=0 2

Example #3

```
char z = 'c';
switch(z) {
case 'n':
cout << " you are welcome\n";
case 'q';
case 'p';
case 'c':
cout << "in\n" ; cout << "ok\n"; break ;
cout << "BAU"; break;
default
cout << "jordan";
}}
```

in ok

?

www.muslimengineer.info
facebook.com/muslim.engineer.page

Example # 4

```
int t;
Cin >> t;
switch (t)
{
Case 1:  cout << "jan" << endl;
Case 2:  Cout << "feb" << endl;
break;
Case 3:
Cout << "march";
break;
}
}
```

t=1
jan
feb

t=2
feb

t=3
march

t=5
nothing

Example #5

```
int a, b
Cin >> a >> b;
switch (a-b) {

   Case 3: cout << " Hi" << endl; break;
   Case 4: Cout << "Why" << endl; break;
   Case 5: Cout << "bye" << endl;   "   ;
}
   Cout << "Salam" << endl;
   Cout << "3alikom" << endl;
   Cout << a+b;
}
```

a=5 , b=3
Salam
3alikom
8

a=9 b=5
why
Salam
3alikom
14

facebook.com/Smurfs.On.The.Way

Example #6

```
int X=3, Y=2, Z=4;
switch( X+Y++)
{
Case 1:
Cout << " A" << "B";
break;

Case 5:
Cout << ++Y << Z++ << "\n";

Case 4:- Cout << "C";

Case 2 : Cout << X % Z << "\t";

default :-
Cout << " Done";
break;
}
```

44 C 3 Done

www.muslimengineer.info
facebook.com/muslim.engineer.page

Computer Skills 2

Loops and arrays Assignments

`double x=1;`
`char c='a';`
`for(int i=-2; i<1; i+=1)`
`{`
`c++;`
`x++;`
`if(c!='c')`
`break;`
`else`
`cout<<c;`
 	
`cout<< !x<<",">>x<<endl;`
	Output
	b0,2

`int a=5;`
`while(a >= 0)`
`{`
`a--;`
`cout<<"\t*";`
`}`
	Output
	* * * * * *

facebook.com/Smurfs.On.The.Way

void main()
{
int x=2;
x+=0.1;
for(int i=0;i<2;i++)
{
switch(x++)
{
case 2:cout<<2<<" ";break;
case 3: if(x==3) cout<<x; else cout<<!x; break;
default: cout<<"bye\n";
}
}
}
	Output
	2 0

for(int i=4; i>0 ;)
{
i--;
cout<<!(!i);
if(!(!i))
break;
	Output
	1

www.muslimengineer.info
facebook.com/muslim.engineer.page

void main()
{
int a[3]={8,2,5};
for (int i=0;i<=2;i++)
{
for (int j=0;j<=2;j++)
cout<<++a[j]<<" ";
cout<<endl;
}
}
	Output
	9 3 6
	10 4 7
	11 5 8

for(int i=0;i;i++) cout<<i;	Output nothing
for(int i=1;i;i++) cout<<i;	Output Infinite loop

int x=2;
for(; x<4;x++)
cout<<++x;
	Output
	3

for(int x=-5; x<=5;x++)
if((x-1)*(x-2)*(x-3)==0)
cout<<x;
	Output
	123

facebook.com/Smurfs.On.The.Way

int i=0,x=0;
 	
while(x<=6)
 	
{
 	
i++;
2
x=i+4;
 	
}
 	
cout<<"x="<<x;
	Output
	X = 7

int x,n=1;
 	
for(int i=0; i<n;i++);
 	
x=1/(2*i-1);
 	
cout<<"x="<<x;
	Output
	X = 1

for(int j=0; j<4,j++)
 	
{
 	
cout<<"$";
 	
if (j%3>=2)
 	
cout<<"+";
 	
else
 	
cout<<"-";
 	
}
	Output
	
	$-$-$+$-

int a[3]={16,1,2};
cout<<a[0]<<endl;
if(a[0]>a[1])
cout<<a[1];
 	
else
cout<<a[2];
	Output
	16
	1

www.muslimengineer.info
facebook.com/muslim.engineer.page

`int a[3]={16,1,2};`
`switch(a[1])`
`{`
`case 1: cout<<1;`
`case 2: cout<<2;`
 	
`default: cout<<9;`
`}`
	Output
	129

`char cc[3]={'a','b','2'};`
`for(int i=2;i>=0;i--)`
`cout<<cc[i];`
	Output
	2ba

`float x = 10.0;`
 	
`for (int j=0; j<5; j++)`
 	
`{`
 	
`x /= 2.0;`
 	
`if (x > 2.0)`
 	
`continue;`
 	
`cout << "computing\n";`
 	
`}`
	Output
	computing
	computing
	computing

`double k[3]={1.1,2.2,3.3};`
`int i=0;`
`while(k[i]<3)`
`{`
`cout<< k[i];`
`i++;`
`}`
	Output
	1.12.2

`double k[3]={1.1,2.2,3.3};`
`int i=0;`
`do{`
`cout<< k[i];`
`i++;`
`}`
`while(k[i]<3);`
	Output
	1.12.2

facebook.com/Smurfs.On.The.Way

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

int b[3]={4,5,6};

for( i-- ;i>=0;i--)
{
a[i]=b[i];
if(a[i]%2==0)
cout<<a[i];
}
```

Output
00064

```
#include<iostream.h>
void main()
{

int list [3][3]={{1,2,3},{1,1,1},{0,0,0}};
int s1=0;
int s2=0;

for(int i=0;i<3;i++)
  for (int j=0;j<3;j++)
     if(i!=j)
         s1+=list[i][j];
     else
         s2+=list[i][j];

cout<< s1<<"    "<<s2;

}
```

Output

7 2

www.muslimengineer.info
facebook.com/muslim.engineer.page

أسئلة مقترحة وسابقة لمادة الفاينال مع الحلول

facebook.com/Smurfs.On.The.Way

User defined functions

These functions should have :

1. **prototype.**
2. **body.**
3. **call(invoke)**

example

```
#include <iostream>
Using namespace std;        →  header for the functions
in each program .

int addition (int,int);        →  function prototype.
// int addition : the value which is returns from the
function below

int main ( ){

int z;              →  z type will be as the int type
z=addition (5,3)            →  function call

cout << " the result is " << z<< endl;

return 0
}
 int addition ( int a,int b) {      →  function body …
int r;

r= a + b ;
return r ;
}
```

The result is 8

www.muslimengineer.info
facebook.com/muslim.engineer.page

Example

```
#include <iostream>
Using namespace std;
int addition (int,int)
int main ( ){

int z;
int x,y;
cout << " enter two values " << x<<y;
cin>> x >>y;
z= addition( 5,3);//    call 1: using integers.
cout<<"the result is "<<z<<endl;
cout<< addition(x,y)<<endl;  // call 2 using variables.
cout<< addition(x+1,y>0)<<endl; // call 3 expression
// y>0 … parameter passing  … true or false. ( 1 or 0 )
 Return 0;
}

int addition (int a , int b ){  // body

int r;
r= a+b;
return r ;
}
```

```
Enter two valus 10 20
The result is 8
30
12
```


facebook.com/Smurfs.On.The.Way

Example

```
# include < iostream>
Using namespace std;

int addition ( int a , int b );

return a + b ;
}
// here we can skip prototype and transfer all body
below the main  to above the main ( here instead of
defining z ,r....etc we can use return a + b .

int main ( ) {

int x ;
x= addition (10 , 20 );
cout << x << endl;
return 0;
}
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

Example

```
# include < iostream>
Using namespace std;

int addition ( int a , int b );
char fun1(int);
int main( ){
int x=5 ,y=5;
cout<<addition (x,y)<<endl;
cout <<fun1(1)<<endl;
return 0;
}
int addition(int a,int b){
return a+b;
}
char fun1(int val){
int z;
z= addition(val,val+1);
cout <<z<<endl;
return'f' }
```

```
10
3
f
```


facebook.com/Smurfs.On.The.Way

1

C++

م. سامر حامر

٩-٥-٢٠١١

Functions

library ⇒ # include <math.h>

Built-in Functions :-

1 ceil يقرّب لأكبر عدد صحيح

Example :-

ceil (10.2) ⇒ 11

ceil (10.7) ⇒ 11

ceil (10) ⇒ 10

ceil (-10.2) = -10

2 Floor يقرّب لأصغر عدد صحيح

example :

floor (10.2) ⇒ 10

floor (10.7) ⇒ 10

floor (10) ⇒ 10

floor (-10.7) ⇒ -11

3 Sqrt الجذر التربيعي

example

sqrt(4) → 2

www.muslimengineer.info
facebook.com/muslim.engineer.page

[3] pow القوّة

example *

$pow(2,3) = 2^3 = 8$

$pow(4,2) = 4^2 = 16$

[4] $\log(4) \longrightarrow Ln = \boxed{1.38}$

$\log_{10}(4) \rightarrow \boxed{0.6}$

$\log_{10}(10000) \rightarrow \boxed{4}$ $\quad 10^{?} = 10000 \Longrightarrow \boxed{4}$

[5] $\cos(x)$, $\sin(x)$, $\tan(x)$ ...

x: Radian

$(\pi/2) \Leftarrow (90^\circ)$

$\pi \Leftarrow 180^\circ$

⋮

etc

[6] abs $\Longrightarrow$ The absolute value.

$abs(11) = 11$

$abs(-11) = 11$

[7] exp $\Longrightarrow$ $exp(5) = e^5 = 148$

$e = 2.71 \leftarrow$

facebook.com/Smurfs.On.The.Way

8 Rand ( ) % 5 ;

اعطاء رقم عشوائي

ex

```
float a = -4.9 ;
cout << pow(floor(a), 2);
```

$\downarrow$

$-5 \Rightarrow pow(-5, 2) = \underline{\underline{25}}$

www.muslimengineer.info
facebook.com/muslim.engineer.page

```
int w = 38;
Void fet( ) { cout << endl << w ;
} void main( )
{ int w = 23;
  cout << w;
  fet( ); }
```

W = 23

23
38

facebook.com/Smurfs.On.The.Way

1-What is the output of the following code?

```
bool x=2;
float z=5.2;
cout<<z+x;
```

- * 3.2
- -2
- 6.2
- 5.2

2-What is the output of the following code?

```
int x=4;
if(x==3);
cout<<x;
cout<<++x;
```

- * 5
- 4
- 45
- 54

3-What is the output of the following code?

```
int w=11;
w=2+3*1/6-5>=3;
cout<<w;
```

- 1
- 3
- 0
- * 11

4-What is the output of the following code?

```
int k=30;
if(k=40)
cout<<k;
else
cout<<++k;
```

- * 31
- 40
- 30
- None

5-What is the output of the following code?

```
float y=3.3;
if(y>3.2)
y/=2;
cout<<y;
else
cout<<y;
```

- 3.3
- None
- Error
- * 1.65

6-What is the output of the following code?

```
int w=5;
switch(w){
case 4: cout<<"four= "<<w;break;
case 5: cout<<"five= "<<w;
```


www.muslimengineer.info
facebook.com/muslim.engineer.page

```
case 6: cout<<"six= "<<w;break;
}
cout<<w/2;
```

* five= 5six= 52
- five= 5six= 62.5
- five= 52
- five= 5six=62

7-What is the output of the following code?

```
int x=2;
while(x<3);
{cout<<x;
x++;}
```

* 2
- Error
- 23
- Infinite loop

8-What is the function of the following code?

```
int i,j,t,a[ ]={3,0,-1,5,4};
cin>>j;
for(i=0; i<5;i++)
if(a[i]==j)
{t=1;
break;}
if(t==1)
cout<<"element"<<i;
```

- Descending sorting
- Ascending Sorting
- Binary search
- * Linear search

9-What is the output of the following code?

```
int a[3][3]={{1,2},{3},{5,6,7}};
cout<<a[0][2];
cout<<a[1][0];
```

- 33
- 03
- 35
- 00

10-What is the output of the following code?

```
char a='c';  //c in ASCII=99
int x=9;
cout<<x;
x=a;
cout<<x;
a++;
cout<<a;
x++;
cout<<x;
```

- 9989999
- 999d100
- * 999100100
- 9cdd

11-What is the output of the following code?

```
float z=-0.3;
cout<<pow(floor(z),3);
```

* -1
- -8
- 8
- 1

facebook.com/Smurfs.On.The.Way

1

12-What is the equivalent of the following in C++?

x=|sin(45)|

* x=|sin(0.78)

x=abs(sin(45))

x=abs(sin(1.57))

x=abs(sin(0.78))

13-What is the output of the following code?

```
int a=10,c=10;
cout<<a<<c<<endl;
{int a=5;
c++;
a+=3;
cout<<a<<c<<endl;
}
a*=2;
cout<<a<<c;
```

a
1010
811
2010

b
1010
811
2011

c
1010
810
2011

d* none

14-What is the output of the following code?

```
int f(int a, int b)
{cout<<a*b;
return a+b;}
void main()
{int n1=5,n2=7;
cout<<n1<<n2;
cout<<f(n1,n2);}
```

57120

* 573512

5700

571235

15-What is the output of the following code?

```
void f(int &s, int n, int &k)
{for(int i=n;i<=k;i++)
s=s+i;
n=n*3;
k=k+2;
cout<<n+k;}
void main()
{int x=0,y=3,z=5;
f(x,y,z);
cout<<x<<y<<z;}
```

161237

1612310

* 16035

www.muslimengineer.info
facebook.com/muslim.engineer.page

- 161235

16-What is the output of the following code?

```
int f(int x)
{static int y=3;
x++;
y++;
return y+x;}
void main()
{int y=5,x=2;
cout<<y<<x;
cout<<f(y);
cout<<f(y);
cout<<y<<x;
}
```

- * Error
- 52101152
- 52101182
- 52141552

17-What is the output of the following code?

```
void f(int b[])
{b[4]=b[0];
cout<<b[4];
++ b[2];
cout<<b[3];}
void main()
{int a[]={30,1,2,4,-3};
f(a);
cout<<a[2];
}
```

- 3043
- 3045
- 3034
- * Error

18-What is the output of the following code?

```
int x=5;
void f(int x)
{cout<<x;}
void main()
{x++;
f(x);
int x=2;
f(x);
}
```

- * 62
- Error
- 52
- 12

19-What is the output of the following code?

```
char a[50]="final exam";
char b[20]="C++ ";
strcat(b,a);
cout<<b;
```

- * C++
- C++ final exam

facebook.com/Smurfs.On.The.Way

- final exam
- Error

20-What is the output of the following code?

```
char a[50]="first exam";
char b[20]="final exam";
cout<<strcmp(a,b);
```

- None
- -1
- 0
- *1

21-What is the output of the following code?

```
int x=4;
while(1)
{cout<<x%2;
x--;
if(x<0)
break;
x--;}
```

- *Infinite loop
- 000
- 1010
- 11-1

22-What is the output of the following code?

```
int x=3;
while(x>-2)
{if(x<1)
continue;
cout<<x;
x--;}
```

- *321
- None
- 3210
- Infinite loop

23-What is the output of the following code?

```
float f(int z)
{z++;
return z/2.0;}
void main()
{int h=50;
cout<<h+f(4.5);
}
```

- 52
- *52.75
- Error
- 52.5

24-What is the output of the following code?

```
const int x=8;
cout<<x;
```

- 7
- Error
- *8
- None

25-What is the output of the following code?

```
int f(int z=8)
{z++;
return z;
z++;
cout<<z;}
void main()
{ cout<<f();
cout<<f(3);
}
```

- Error
- 94
- *99
- 91045

www.muslimengineer.info
facebook.com/muslim.engineer.page