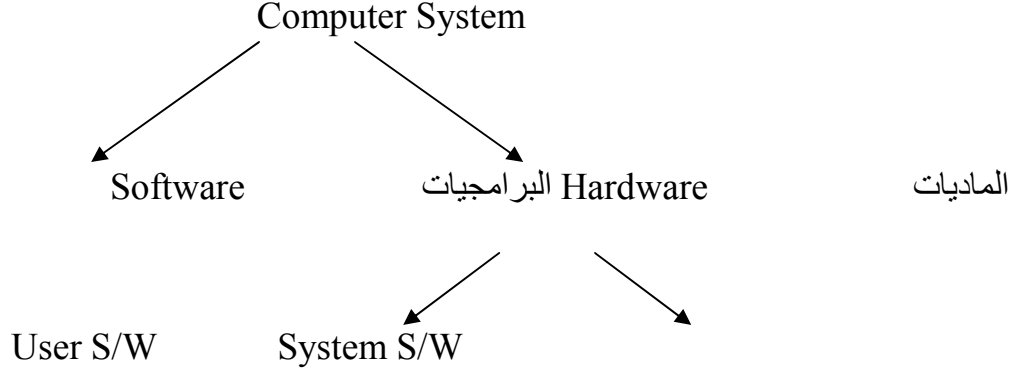


1- Introduction to C++

High level شهرة واسعة لانها اصبحت تنتمي الى لغات المستوى الاعلى C++ او C اخذت لغة مثل لغات بييسك والباسكال من حيث سهولة الاستخدام من ناحية ومن ناحية اخرى تنتمي الى لغات language المكونات المادية للحاسبة H/W (Hardware) المستوى الادنى من حيث قوتها على مخاطبة مكونات الجهاز او الكمبيوتر.



C++- مميزات لغة

عن غيرها من لغات البرمجة , وهي المميزات التي تدعوك لتفضيلها على غيرها C++ ان اهم مميزات لغة من لغات البرمجة المعروفة ومن هذه المميزات :-

1- General Language لغة عامة

اي تصلح لعمل برامج قولعد البيانات والرسومات والحسابات ونظم التشغيل .

2- Structural Language لغة تركيبية

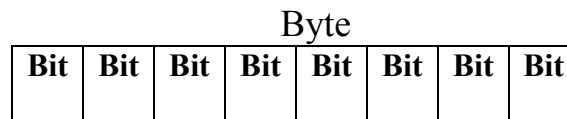
من خلالها يمكن استدعاء دوال () main عبارة عن دالة رئيسية تدعى C++ البرنامج المكتوب بلغة اخرى وكل دالة تتكون مجموعة من الاوامر والخطوات .

3- لغة سريعة

لان ادوات اللغة تعامل مباشرة مع الالة مما يختصر وقت التنفيذ .

4- Bit Manipulation تتعامل على المستوى " البت "

هي Bit , وكما هو معروف فان Bit حيث تستطيع ان تقراء وتكتب وتغير وتقوم بعمليات على مستوى اصغر وحدة ياس المعلومات داخل الكمبيوتر وهي جزء من ثمانية اجزاء تعادل من مجموعات حرف واحد Byte()



وهذه الميزة جعلتها متخصصة في بعض مجالات التحكم الالي والروبوت وبرامج معالجة الصور و ضغط الملفات.

5- لغة متقلة (Portable)

والاجهزة المتوسطة والكبيرة. IBM يعمل مع اكثر من جهاز مثل C++ اي يمكن ابرنامج المكتوب بلغة

تتميز الكمبيوترات عن البشر بكونها

- 1- تستطيع الكمبيوترات ان تخزن كميات كبيرة من المعلومات.
- 2- تستطيع ان تتذكر تلك المعلومات بسرعة ودقة.
- 3- تستطيع ان تنفذ العمليات الحسابية بسرعة عالية وبدقة مثالية.

2- C++ Program Structure

- You need to know what components are necessary to create a C++ program and how to enter and run a program in Turbo C++.

STEP 1: Analyze the problem تحليل المشكلة

- Begin with a clear idea of what you want a program to do.
- Most programs are written to solve a problem.
- Consider what sort of input the program needs to work with and what the program will provide as output.
- As you analyze the problem, it is usually best to get pen and paper together and write down on paper exactly what the program going to do.

STEP 2: Design the program

- Decide how the program will work.
- Work any algorithms required by the program.

STEP 3: Implement the program by writing code

- Design the code to carry out the goals of your program description.
- Your knowledge of the language is most important at this step.
- To implement the program you need to know the C++ programming statement and instruction.
- This is the step in which you start using IDE to type in and edit the source code for the program

STEP 4: Compile and run the program

- Before you can actually execute the program, you must compile it. Because the IDE (Integrated Development Environment) has the compiler integrated into it, the compilation step is fast and easy.
- Compilation is converted from high-level source code to the machine-level (low-level) instruction.
- The compiler also makes sure that the program you typed in is grammatically correct.

- Just as the English language has rules of grammar and structure that are correct, the compiler has rules of grammar and structure it understands.

Machine-level الى High-level هي عملية تحويل Compilation عملية

يجري عملية تصحيح الاخطاء الموجودة في البرنامج له قواعد وخطوات اساسية. Compiler.

STEP 5: Test and Debug the program

3- A Simple C++ Program

Listing 1: shows our first C++ program, which when run, simply output the message (Hello World).

1- # include <iostream.h>	
2- void main ()	
3- {	بداية الدالة
4- cout <<"Hello world\n";	
5- }	نهاية الدالة

Listing 1

- اسم الملف التوجيه الخاص < > بين العلامتين #include اسم الملف < H . يبدأ البرنامج بالعبار > بالدوال المستخدمة في البرنامج .
- { وتنتهي بالقوس } تبدأ بالقوس () main يتكون البرنامج من دالة رئيسية .
- جميع كلمات ودوال اللغة تكتب بالحروف الصغيرة .
- (كما في السطر رقم (4) مع وجود استثناءات: تنتهي كل عبارة بفاصلة منقوطة)
- (خاصة بالمبرمج بوضعها بين العلامتين /* بداية Comments يجوز كتابة اي ملاحظات او تعليقات) /* في نهاية التعليق لاي عدد من السطور . اما في حالة كون التعليقات سطر واحد // توضع هذه العلامة في بداية السطر .
- > بمعنى اشتمل او حمل محتويات هذا الملف الى البرنامج حيث ان هذا #include اسم الملف < H . والسطر معناه (حمل او اضع محتويات include في الدليل C++ الملف موجود مع حزمة برامج لغة هذا الملف الى البرنامج). لان هذا الملف يحتوي على تعريفات الدوال التي سوف تستخدم في البرنامج ومن هنا نفهم ان كل مجموعة دوال من دوال اللغة لها ملف خاص بها يجب كتابته في اول البرنامج.
- هذه العبارة هي الدالة الرئيسية للبرنامج حيث ان البرنامج سوف () main يحتوي البرنامج على عبارة هي الدالة الرئيسية التي تنادي باقي الدوال () main يتركب من مجموعة دوال ولكن الدالة

Example:-

```
/* First program
Name: Hello world in C++ by
Class: A or B */
#include <iosteream.h> // I/O library
void main ( ) // name of function
{ // Body of function
    cout << "Hello, world!" << endl;
}
```

تعلیق

معنى هذه الدالة هي بداية سطر جديد

Example: - H.W

```
# include <iosteream.h>
# include <conio.h>
void main ( )
{
    clrscr ( );
    cout << "matrix";
    cout << "\t matrix";
    cout << "matrix \n";
    cout << "matrix \n\n";
    cout << "matrix \?"
    cout << "matrix \?";
}
```

Escape Sequence تسلسلات الهروب

: "\n "	الانتقال الى سطر جديد	new line
: "\t "	نقل المؤشر بعد 8 مسافات	tap
: "\\? "	طباعة علامة الاستفهام	
: "\\\" "	طباعة علامة التخصيص	double quotation
: "\' "	طباعة	Single quotation
: "\\ "	طباعة	back slash

H.W

1. Write C++ program Run the table

Name	First yeare	end yeare
Ahmade	1990	1994
Ali	1995	2000

2. Write C++ program print the string :

Computer "and Information" Dep

3. Write C++ program print the string :

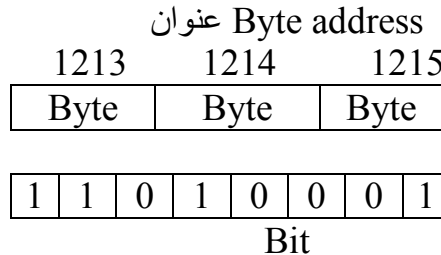
The Tap Character is represented as '\t'

4. Write C++ program print the string :

The new line escape sequence is '\n'

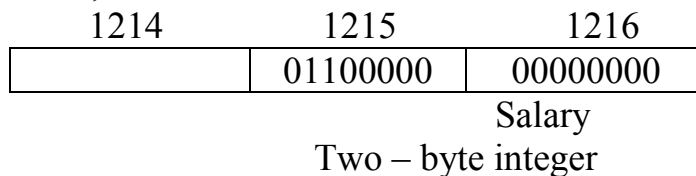
4- Memory and Data type الذاكرة وانواع البيانات

A computer provides a Random Access Memory (RAM) for storing. This memory can be thought of as a contiguous sequence of bite each of which is capable of storing a binary digit (0 or 1). Typically, the memory is also divided into groups of 8 consecutive bits (called bytes). The bytes are sequentially addressed. Therefore each byte can be uniquely identified by its address.



The C++ compiler generates executable code which maps data entities to memory location. For example, the variable definition

int salary = 32768;



Data type انواع البيانات

(.بالاضافة الى ذلك , يستعمل text) او نصية (digit تأتي البيانات في عدة اشكال. وهي عادة اما رقمية) (6.1-) بينما تستعمل البيانات الرقمية الاخرى ارقاما حقيقية (65 , 0 , 7- بعض البيانات الرقمية ارقاما كاملة) (. 7.3 , 0.6 , لن يؤثر نوع البيانات الذي نختاره على شكل تخزين البيانات فحسب , بل على كمية الذاكرة المطلوبة لتخزين البيانات ايضا.

نوع المتغير	Bytes طوله بالبايت	المدى المسموح
char حرف	1	حرف واحد او رمز
int عدد صحيح	2	32768 الى -32768
long عدد صحيح	4	$214704830648 \bar{+}$
float عدد حقيقي	4	$38 \bar{+} \in 3.4$
double عدد حقيقي مضاعف	8	$308 \bar{+} \in 17$

- ان حجم نوع البيانات يهتما في مسألتين مرتبطتين
- (1) النطاق (المدى): القيم المختلفة التي قد يمثلها نوع البيانات
 - (2) كمية الذاكرة المطلوبة لتخزين نوع البيانات

سؤال (1): ايجاد مساحة المستطيل باستخدام قيم ثابتة

Example: -

```
# include <iostream.h>
void main ( )
{
    int length, width, area;
    length = 20;
    width = 15;
    area = length * width;
    cout <<" The area =" << area << endl;
```

سؤال (2): ايجاد مساحة المستطيل باستخدام قيم متغيره للطول والعرض

Example :-

```
# include <iostream.h>
void main ( )
{ int length, width, area;
  cin >> length;
  cin >> width;
  area = length * width;
  cout <<" The area =" << area << endl;
}
```

The algorithm to find the perimeter and area:

Step1: Get the length of rectangle.

Step2: Get the width of rectangle.

Step3: Find perimeter using the following equation perimeter
 $= 2 * (\text{length} + \text{width})$.

Step4: Find the area using the following equation $\text{area} = \text{length} * \text{width}$.

Example:-

```
# include <iostream.h>
void main ( )
{
    int length, width, area;
    cout <<" Enter the value of length =";
    cin >> length;
    cout <<" Enter the value of width =";
    cin >> width;
    area = length * width;
    cout <<" The area =" << area << endl;
}
```

تنفيذ البرنامج

Enter the value of length =

9 ←

Enter the value of width =

10 ←

The area = 90

5- Operator المؤثرات

المؤثرات هي الرموز التي تربط بين المتغيرات والثوابت لإنشاء علاقة ما او معادلة تؤدي وظيفة معينة .

1. Arithmetic Operator المؤثرات الحسابية

Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

% يشترط ان يكون نوع بيانات من نوع int 13%3

2. Relational Operator مؤثرات المقارنة

Operator	Name	Example	gives
==	Equality	5==5	1
!=	Inequality	5!=5	0
<	Less Than	5<5.5	1
<=	Less Than or equal	5<=5	1
>	Great or Than	5>5.5	0
>=	Great or Than or equal	6.3>=5	1

Where 1 = True

0 = False

3. Logical Operators

Operator	Name	Example	gives
!	Not Logical Negation	!(5==5)	0
&&	Logical And	5<6 &&4<6	1
	Logical Or	5<6 6<5	1

4. Increment \ Decrement Operators

Operator	Name	Example	gives
++	auto Increment	K=5++k+10	16
--	auto Decrement	- -k+10	14

5. Assignment Operator

Operator	Name	Example	gives
=	n = 5		n = 5
+=	n += 25	n = n + 25	n = 30
-=	n -= 25	n = n - 25	n = -20
*=	n *= 25	n = n * 25	n = 125
/=	n /= 25	n = n / 25	n = 0.2
%=	n %= 25	n = n % 25	n =

Example:-

$$m + n \neq 10 / m = m + (n = p = 10)$$

An arithmetic expression may contain one or more sets of parentheses. In such expression, you must deal with parentheses first. Start with the innermost set of parentheses and evaluate the sub expressions according to the following rules.

1. First do all multiplication and / or division operators, including modulus (%) operators. All three have the same priority. Then do all additions and subtractions. These operators also have the same precedence as each other.
2. If there are several operators of the same precedence, they have left associativity, so perform the operation from left to right.

Let us now apply the rules to the following arithmetic,

$$12m - (mn \% 13 + m/n) * 10/k$$

Assume $m = 12$, $n = 5$ and $k = 20$

sub expression	Result	Expression After Each step
m * n	60	m*12+(60%13+m/n)*k/10
60 % 13	8	m*12+(8+m/n)*k/10
M / n	2	m*12+(8+2)*k/10
8 + 2	10	m*12+10*k/10
m * 12	144	144+10*k/10
10 * k	200	144+200/10
200 // 0	20	144+20
14 + 20	164	164

Example: - A C++ program for computing the volume of a sphere.

$$V = 4 * \pi * r^3 / 3$$

```
# include <iostream.h>
void main ( )
{ float r, v; // declare variables . r is radius. v is volume
  float pi = 3.14 // set value of pi
  cout <<" Enter the radius r = ";
  cin >> r; // read from keyboard
  v = (4.0 * pi * r * r * r)/3.0;
  cout <<" \n v = " << v << endl;
}
```

Exercises

1. Write a program which inputs a temperature reading expressed in Fahrenheit and output its equivalent in Celsius, using the formula:

$$C^{\circ} = \frac{5}{9} (F - 32)$$

Compile and run the program, its behavior should resemble this:

Temperature in Fahrenheit: 41

41 degrees Fahrenheit = 5 degrees Celsius.

2. Write a program for the following computations

a) $Z = 1 + \frac{x^2}{2!} + \frac{x^4}{4!}$

b) $Z = \frac{x^3 - 4x^2 + X}{x^2 + 2X + 2}$

c) $W = [(x-y)^2 - (x+y)] / 32.0$

3. Give the length and width of a rectangle, write a program to compute and display its area and perimeter as shown below:

O/P	length	width	area	perimeter
	100	50	5000	300

4. we consider the arithmetic expression

$$a * b / (-c * 31 \% 13) * d$$

What is the order in which the operations will be carried out?

5. consider the expression

$$a * (b * b) - (c * b) + d$$

What is the order in which the operations will be executed?

6. What is the evaluation sequence of the following expression if $a=10.0$, $b=20.0$, $c=14.0$, $d=8.0$, and $e=40.0$.

$$(a + b / (c - 5.0)) / ((d + 7.0) / (e - 37.0) / 3.0)$$

7. The expression $(a \% b)$ and $(a - (a/b) * b)$, where a and b are integer variable, are equivalent. (T or F).

8. For each the following algebraic expressions write an equivalent C++ arithmetic expression.

a) $\frac{a^3 - b^3}{d^3 + 17}$

b) $\frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4}$

d) $x + y^2 + \frac{t}{z}$

6. Conditional Operator

condition ? exp1 : exp2

Example: -

$$y = \begin{cases} x + 3 & x > b \\ x^2 - 10 & x \geq b \end{cases}$$

Sol:

$$y = \underset{\text{condition}}{(a > b)} ? \underset{\text{exp1}}{x + 3} : \underset{\text{exp2}}{x * x - 10};$$

Example: - اكتب برنامج ليجاد كلفة صر الطاقة الكهربائية علما بان تكلفة وحدات الطاقة
الكهربائية حسب مامدون في الجدول التالي

counter	Cost/unit
0 - 100	20
101-150	30
151 - ∞	50

counter 70 : cost = 70 * 20

counter 130 : cost = 100 * 20 + 30 * 30

counter 200 : cost = 100 * 20 + 50 * 30 + 50 * 50

```
# include <iostream.h>
void main ( )
{int a, cost;
cout <<" Enter the value of units:";
cin >> a;
cost = (a <= 100) ? a * 20: (a <= 150)? 100 * 20 +
(a - 100) * 30: 100 * 20 + 50 * 30 + (a - 150) * 50;
cout <<" The cost = "<< cost << endl;
}
```

رسم هنا 21

Statements

- Simple and compound statements

A simple statement is a computation terminated by a semicolon

```
int i ;
++ i ;
cout << i ;
```

Multiple statements can be combined into a compound statement by enclosing them within braces

```
{int min , i=10 , J=20
  cout << " the min value =";
  cout << min <<endl;
}
```

Conditional statements

- 1) if statements
- 2) if else statements
- 3) Nested if statements
- 4) Switch statements

1) if statements الجملة if
True (لتنفيذ جملة او اكثر سب شرط معين في حالة كون الشرط صحيح (if تستخدم الجملة
if (exp.)
statement; رسم هنا 23

Example: اكتب برنامج يفحص اذا كان العدد الذي يكتبه المستخدم هو رقم زوجي او فردي.

```
# include <iostream.h>
void main ( )
{ int n;
  cout <<"Enter a number = ";
  cin >> n;
  if (n % 2 == 0)
    cout <<" The number is even " << endl;
}
```

رسم هنا 23

2) if else statement

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

24 رسم هنا

Example 2: - اكتب برنامج لحل المعادلة التالية -

$$y = \begin{cases} x+3 & x > 10 \\ 10x & x \leq 10 \end{cases}$$

1- Using if else statement.

```
# include <iostream.h>
void main ( )
{ float y, a;
  cout <<"Enter value of a=";
  cin >> a;
  if (a>10)
    y=a+3;
  else
    y = 10*a;
  cout <<" The value of y=";
}
```

2- By using conditional operator.

```
y = (a>10)? a+3: a*10;
```

3- The if else if else statements

```
if (condition)
    statement 1;
else if (condition)
    statement 2;
else if (condition)
    statement 3;
else
    statement 4;
```

Example: - البرنامج التالي يوضح استخدام if-else if البرنامج يوضح تقدير كل درجة

```
# include <iostream.h>
void main ( )
{ int s;
  cout <<" Enter your test score =";
  cin >> s;
  if (s>=90)
    cout <<" your grade is excellent" << endl;
  else if (s>=80)
    cout <<" your grade is very good" << endl;
  else if (s>=70)
    cout <<" your grade is good" << endl;
  else if (s>=60)
    cout <<" your grade is med" << endl;
  else if (s>=50)
    cout <<" your grade is a pass";
  else
    cout <<" your grade is a not pass ";
}
```

Example: - (/ , * , - , +) اكتب برنامج ينفذ العمليات الحسابية

```
# include <iostream.h>
3- void main ()
{ float x,y; char op;
  cout <<"Enter value of x =";
  cin >> x;
  cout <<"Enter value of y =";
  cin >> y;
  cout <<"Enter operator =";
  cin >> op;
  if (op == '+' )
  cout << x+y;
  else if (op == '*' )
  cout << x*y;
  else if (op == '/' )
  cout << x/y;
  else
  cout <<"unknown operator" <<op;
}
```

3) Nested if statement
if (condition 1)
if (condition 2)

if (condition n)
statement;

رسم هنا 28

Example: -

```
    y = x2           10 < x ≤ 100
if (x > 10)
if (x ≤ 100)
    {y = x * x;
    cout << y;
    }
```

OR

```
If (x > 10 && x ≤ 100)
    {y = x * x;
    cout << y;
    }
```

H.W

$$1- y = \begin{cases} 2x^2 + 4 & 0 < x \leq 100 \\ 0 & x \leq 0 \\ x^2 + x + 2 & x > 100 \end{cases}$$

- 2- اكتب برنامج لمعرفة اذا كان العدد موجب (positive) او سالب (negative) او صفر (Zero)
- 3- Write program to find the number if positive, negative or zero.

4- If a=1 , b=2 , and c=3 , what are the values of a,b and c at the end of the following program segment?

```
if (a <= b)
    if (c > 2)
        c = 2;
if (c < 3)
    a = 0;
else
    b = 0;
if (ch >= '0' && ch <= '9')
    cout << "kind = digit";
else if (ch >= 'A' && ch <= 'Z')
    cout << "kind = capital letter";
else if (ch >= 'a' && ch <= 'z')
    cout << "kind = small letter";
else
    cout << "kind = special";
```

character	ASCII	انواع البيانات النص
0 - 9	48 - 57	
A - Z	65 - 90	
a - z	97 - 122	

Q6: A digital thermostat displays the following messages according to the temperature in your house, in Celsius,

Cold	Chilly	Good	Warm	Hot
10 - 14	15 -19	20 - 24	25 – 29	30 and higher

Write a nested if – else statement for the above conditions.

```
If (temperature > 14)
  If (temperature > 19)
    If (temperature > 24)
      If (temperature > 29)
        cout << "Hot" << endl;
      else cout << "Warm" << endl;
    else cout << "Good" << endl;
  else cout << "Chilly" << endl;
else cout << "Cold" << endl;
```

4. Switch

انها تقييم قيمة تعبير رقم صحيح ثم تقارن تلك القيمة بقيمتين if /else if / else شابه Switch الجملة .
اخرين او اكثر تحديد اي حالة يجب تنفيذها .

```
# include <iostream.h>
void main ( )
{ char g;
  cout <<"Enter your grade";
  cin >> g;
  switch (g)
  {
    case 'A':
      cout <<"In your average must between 90 – 100" ;
      break;
    case 'B':
      cout <<"In your average must between 90 – 100" ;
      break;
    case 'C':
      cout <<"In your average must between 90 – 100" ;
      break;
    case 'D':
      cout <<"In your average must between 90 – 100" ;
      break;
    default:
      cout <<"your average must between 90 – 100" ;
  }
}
```

Loops

1. For loop

1. الدوارة for

تستخدم لتكرار تنفيذ عملية عدد محدد من المرات وتأخذ الصيغة العامة التالية :

```
for (initial-value ; condition ; increment)
    statements;
```

حيث

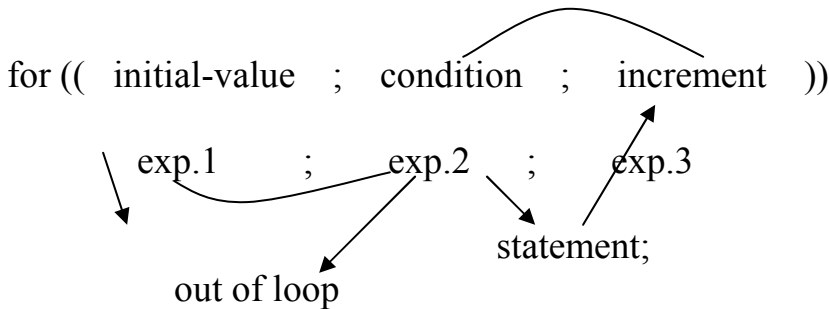
initial – value. القيمة الابتدائية.

condition. شرط انتهاء التكرار.

increment. هي قيمة ازيادة الدورية.

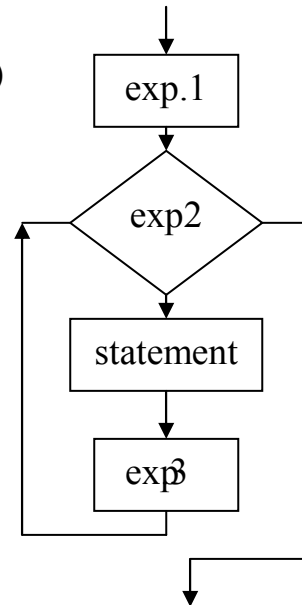
(صحيح ومقدار condition) طالما ان الشرط (initial – value) ومعناه ابداء العد من القيمة الابتدائية)

increment الزيادة كل مرة هو



Note:-

تستخدم for loop عندما يكون عدد التكرار معلوم



1- increasing

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

```
    cout << i;
```

2- decreasing

```
for (int i = 9 ; i > 0 ; i --)
```

```
    cout << i;
```

3- increasing By 2

```
for (int i = 0 ; i <= 100 ; i += 2)
```

```
    cout << i;
```

4- increasing By 3

```
for (int i = 0 ; i <= 100 ; i += 3)
```

اكتب برنامج يطبع الاعداد من 1 الى 10

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

اكتب برنامج لايجاد مجموع الاعداد المحصورة بين 50 الى 100.

```
# include <iostream.h>
void main ( )
{ int sum= 0;
  for (int i = 51 ; i < 100 ; i ++ )
    sum += i;
  cout << "the sum = " << sum;
}
```

اكتب برنامج لايجاد مجموع عشرة اعداد يتم ادخالهم اثناء تنفيذ البرنامج

```
# include <iostream.h>
void main ( )
{ int sum= 0, num;
  for (int i = 0 ; i < 10 ; i ++ )
  {
    cout << "Enter your number: ";
    cin >> num;
    sum += num;
  }
  cout << "The sum=" << sum;
}
```

H.W (OOP C++)

Q1: write C++ program to input a character and check if it is a VOWEL LETTER (a,e,u,i,o) using switch-break statements.

Q2: write C++ program that takes as input given lengths expressed in feet and inches. The program should then convert and output the length in centimeters. Assume that the given length in feet and inches are integers. (using the approximation that 1 inch is equal to 2.54 cm) and (1 foot is equal to 12 inches).

$$\begin{aligned}\text{Ex: - input 5 feet and 7 inches} \\ \text{Total inches} &= (12 * \text{feet}) + \text{inches} \\ &= 12 * 5 + 7 \\ &= 67 \\ \text{Centimeters} &= \text{Total inches} * 2.54 \\ &= 67 * 2.54 \\ &= 170.18\end{aligned}$$

Q3: write C++ program that takes as input any change expressed in cents. It should then compute the number of half-dollars, quarters, dimes, nickels and pennies to be returned, returning, as many half-dollars as possible, then quarters, dimes, nickels and pennies, in that order.

For example, 646 cent yields the following calculation:

1. Change = 646
2. Number of half-dollars = $646 / 50 = 12$
3. Remaining changes = $646 \% 50 = 46$
4. Number of quarters = $46 / 25 = 1$
5. Remaining change = $46 \% 25 = 21$
6. Number of dimes = $21 / 10 = 2$
7. Remaining change = $21 \% 10 = 1$
8. Number of nickels = $1 / 5 = 0$
9. Number of pennies = remaining change = $1 \% 5 = 1$

Q4: write C++ program that prompts the user to input a length expressed in centimeters. The program should then convert the length expressed in yards, feet, and inches, in that order. (1 yard = 36 feet and 1 foot = 12 inches)

For example 123 inches would be outputs as:

3 yards, 1 feet and 3 inch

Nested For Loops

```
for (int i = 1 ; i <= 3 ; i ++)  
    for (int j = 1 ; j <= 3 ; j ++)  
        cout << i << "\t" << j << "\n";
```

عند تنفيذ البرنامج

O/P

	i	j	
Loop	1	1	} Loop داخلي
خارجي {	1	2	
	1	3	
	2	1	
	2	2	
	2	3	
	3	1	
	3	2	
	3	3	

Example: -

- Write a C++ program to find the factorial for the **positive number**.
if (n >= 0)

```
# include <iostream.h>  
void main ()  
{ int n;  
  long f = 1;  
  cout << "Enter value of number =";  
  cin >> n;  
  if (n >= 0)    // for positive number  
  for (int i = n; i > 0; i --)  
      f *= i;    // f = f * i;  
  cout << "The factorial is =" << f;  
}
```

- Write a C++ program which produces a simple multiplication table of the following format for integers in the range 1 to 9

```
# include <iostream.h>  
void main ()  
{ for (int i = 1 ; i <= 9 ; i ++)  
  for (int j = 1 ; j <= 10 ; j ++)  
      cout << i << "*" << j << "=" << i * j << endl;  
}
```

2. W

While (condition)
Statement;

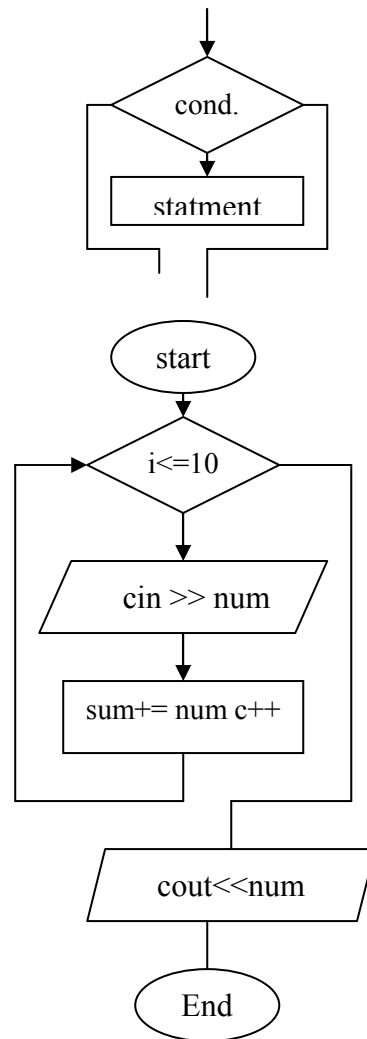
الصيغة العامة

Note:-

تستخدم while loop في حالة عدم معرفة عدد مرات ال loop اي عدم معرفة عدد تكرار. وليس بالضرورة تنفيذ loop .

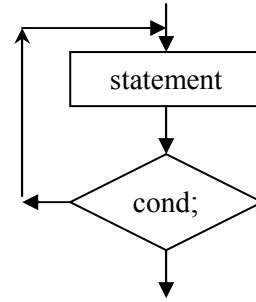
اكتب برنامج لاجاد مجموع عشرة اعداد يتم ادخالها اثناء تنفيذ باستخدام while loop

```
void main ()  
{ int i=1, sum= 0, num;  
  while (i <= 10)  
  {  
    cout <<"Enter no. =";  
    cin >>num;  
    sum += num;  
    i ++  
  }  
  cout <<"the sum =" << sum;  
}
```



3. Do While Statement

```
do الصيغة العامة
{
    statement;
}
while (condition)
```



Note:-

do تشبه while فقط الاختلاف هو انه دائما التنفيذ لل loop يكون على الاقل مرة واحدة والشرط يعتمد على التنفيذ.

نفس المثال السابق

```
# include <iostream.h>
void main ( )
{ int i=1, sum= 0, num;
  do
  { cout <<"Enter value =";
    cin >>num;
    sum += num;
  } while (num);
  cout <<"the sum =" << sum;
}
```

في حالة كون num=0 يتوقف البرنامج ويخرج خارج loop

OR while (num!=0);

Q1: اكتب برنامج لطباعة جدول الضرب من 1 الى 5 ويكون شكل الطباعة

```
void main ( )
{ for (int i = 1 ; i < 6 ; i ++ )
  { for (int j = 1 ; j < 6 ; j ++ )
    cout << i * j;
    cout << endl;
  }
}
```

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Q3: write program that calculates the value of (pi) from the infite series, stop calculation when the value of (pi) exceed 8.7235.

$$Pi = 4 + 4/3 + 4/5 + 4/7 + 4/9 + \dots +$$

Q4: write program that finds the smallest of several integers. Assume that the first value read specifies the number of values.

$$10^1, 5^2, 9^3, 3^4, \dots, 18^n \quad \text{عدد القيم } n$$

Q5: write program that converts the positive integer number into binary and prints it:-

Q6: نفس السؤال السابق يحول العدد الى **octal** نظام ثماني

Q7: An integer is said to be prime if it divisible only by 1 and itself. Write program that determines if a number is prime and use this program that determines and prints all the prime number between (10 To 30).

Q8: what is the O/P of the following program.

```
# include <iosteream.h>
void main ( )
{ for (int i=0 ; i< 7 - i; j <=7 + i; j++)
  { for (int i=0 ; i< 7 - i; j <=7 + i; j++)
    cout << i * j << " ";
    cout << endl;
  }
}
```

Q9: what is the value of the int variable j at the end of the following loops if (j=0) when the loop begins?

- a- for (int i = 8; i >= 0; i = i - 3)
j = j + 1;
- b- for (int i = 0; i <= 8; i = i + 2)
j = j + 1;
i = i + 1;

Q10: if (i=0) and (g=5), what are the values of i and g after the following program segment?

```
while ((I <= 4) && (g > 0))
{ i = I + 1;
  g = g - 1.0;
}
```

Q11: Replace the following for loop with a corresponding while loop.

```
float a = 8;
for (int i = 0; i < 10; i++)
{ if (a == 0) continue;
  cout << 1/a << endl;
  a = a - 1;
}
```

Q12: what will be the output of the following fragments?

- a)

```
int x = 0;
while (x < 10)
  cout << x << endl;
  x = x + 1;
  cout << "Done\n";
```
- b)

```
char ltr;
for (ltr = 'A'; ltr <= 'F'; ltr = ltr + 1)
  cout << ltr;
  cout << endl;
```

Q13: what does this program print?

```
#include <iostream.h>
void main ()
{ for (int i=0 ; i<=10 ; i++)
  { for (int j=0 ; j <=5; j++)
    if (i==0 || i==5 || i==1 || j==0 || j==5)
      cout << ' * ';
    else
      cout << '  ';
    cout << endl;
  }
}
```

Q14: Rewrite the program in (Q13) by using while statements instead of for statements and draw its flow chart.

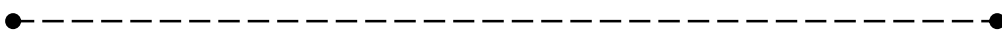
Q15: write a program to computer the number of decimal diits that a positive integer number has.

((حلول الاسئلة))

Q2:

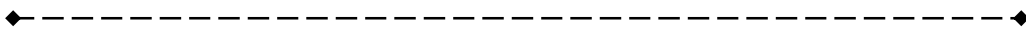
```
for (int i=0 ; i<=4 ; i ++)  
  { for (int j=0 ; j <=4; j ++)  
    { if (i==5 || j==0 || i==j)  
      cout <<" * ";  
      else  
        cout <<"  ";  
    }  
  }  
  cout << endl;  
}
```

```
*  
*  *  
*    *  
*      *  
*        *  
*          *  
*            *  
*              *  
*                *  
*                  *
```



```
{ if (i==0 ; j==0 ; i+j =4)  
  cout <<' * ' ;  
  else  
    cout <<"  " ;  
}
```

```
* * * * *  
*      *  
*        *  
*          *  
*            *  
*              *  
*                *  
*                  *
```



$m = +1$
 $i = m j + c$ المحيط (2 , 2)
 $2 = +2 + c$
 $c = 0$
 $i = j$

$2 = 1 + 0$ داخلية (2 , 1)

$2 = 1 + 0$ \therefore i اكبر من j $i \geq j$

if (j==0 || j==6 || ((i==j) && (i<4)) || ((i+j==6) && (i<4)))

((تكملة حل السؤال Q2))



$i = m j + c$	$(2, 6) \quad m = +1$
$(2, 2) \quad m = -1$	$2 = +6 + c$
$2 = -2 + c$	$c = -4$
$c = 4$	$i = j - 4$
$i = -j + 4$	$j = i + 4$
$j = 4 - i$	$(2, 5)$
$(2, 3)$	$5 = 2 + 4$
$3 = 4 - 2$	$j \leq (i + 4)$
$3 = 2$	
$j \geq (4 - i)$	

```

for (int i = 0 ; i <= 4 ; i ++ )
  {for (int j = 0 ; j <= 8 ; j ++ )
    { if (j >= (4 - i) && j <= (4 + i))
      cout << " * ";
      else
        cout << " ";
    }
    cout << endl;
  }

```

Q5: Binary

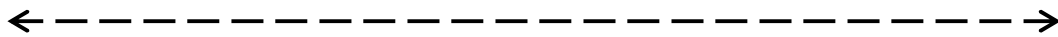
```

{ int x, y = 0 , d , w = 1;
  cout << "Enter dec. number =";
  cin >> x;
  while (x > 0)
    { d = x % 2;
      x = x / 2;
      y = y + d * w;
      w = w * 10;
    }
  cout << y << " The Binary number";
}

```

Q7: Prime

```
{ int f, z;
  for (int x = 10; x <= 20; x++)
    { f = 0
      for (int y = 2; y < x; y++)
        { z = x % y;
          if (z == 0)
            f = 1;
        }
      if (f == 1)
        cout <<x<<" not prime "<< endl;
      else
        cout <<x<<" prime " << endl;
    }
}
```



Q4:

```
{ int min, n;
  int no; \\ no. of variable (values) in array
  cout <<" Enter no. of elements = ";
  cin >> no;
  cout <<" Enter 1st element = ";
  cin >> min;
  for (int i = 1; i <= no; i++)
    { cout <<" Enter next element = ";
      cin >> n;
      if (min > n)
        min = n;
    }
  cout <<" The min. (smallest) value is = " << min;
}
```

Arrays

المصفوفات

(وتسمى هذه float) وبيانات حقيقية (int) وبيانات رقمية (char) وتنقسم البيانات الى بيانات حرفية (بلانواع الرئيسية البيانات , حيث لايمكن تجزئتها اقل من ذلك. ولكن هناك انواع int , float , char الانواع) Arrays. من هذه الانواع المصفوفات (Derived data type) اخرى من البيانات تسمى بالانواع المشتقة

تعرف المصفوفة بانها مجموعة من العناصر تنتمي الى نوع واحد من البيانات ويخصص لها اسم واحد وتنقسم المصفوفات الى مصفوفات ذات بعد واحد ومصفوفات ذات بعدين.

الصيغة العامة للاعلان عن المصفوفات هي

Type name [size];
↑ ↑ ↑
نوع المصفوفة اسم المصفوفة حجم المصفوفة

```
int a[5];  
int a[5] = {7,4,13,8,10};  
int a[ ] = {7,4,13,8,10};
```

في حالة عدم تحديد حجم المصفوفة

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

```
cin >> A [i];              ادخال قيم الى المصفوفة
```

```
for (int j = 0; j < 10; j++)
```

```
cout << A [j] << end L;              طباعة قيم المصفوفة
```

```
OR cout <<"a [" << i << "]" = << a [i] << end L;
```

Example: - write a program to calculate the sum of array?

```
# include <iostream.h>  
void main ()  
{ int a [5] = {8,7,3,9,2};  
int sum = 0;  
for (int i =0; i < 5; i++)  
sum += a [i];  
cout <<" The sum =" << sum << end L;  
}  
for calculate the average  
{  
cout <<" The average =" << float (sum)/5.0 <<endl;  
}
```

Example: - write a program to find the max no. in an array?

```
# include <iostream.h>
void main ()
{ const int size = 5;
int a [size] = {          };
int max = a [0];
for (int i = 1; i < 5; i++)
if (max < a [i])
max = a [i];
cout <<" The max number = " << max <<endl;
}
```

Example: - write a program to calculate the no. of even in tiger no. in array.

```
{ const int size = 5;
int a [size] = {          };
int count= 0
for (int i = 0; i < 5; i++)
if (a [i] % 2 == 0)
count ++ ;
cout <<" The number of even number = " << count <<endl;
}
```

1). Array of string مصفوفة العبارة الحرفية

هي (string) وان عن سلسلة حروف (string) تستعمل كلمة سلسلة حروف مقابل كلمة
char name [10];
ومعناه مصفوفة من الحروف

char name [5] [10];

(صفوف كل صف عبارة عن كلمة لا 5). اي (10) وعدد اعمدتها (5)معناه مصفوفة حرفية عدد صفوفها)
(حروف 10 يتعدى عدد حروفها عن)

```
char name [5] [10] = {{"string 1"},  
                      {"string 2"},  
                      {"string 3"},  
                      {"string 4"},  
                      {"string 5"}};
```

في هذه المصفوفة يمكن التعامل مع حرف معين كما يمكن التعامل مع صف كامل على انه سلسلة حروف
او مصفوفة حرفية. array of string ولهذا تسمى هذه المصفوفة string

Example: -

البرنامج يشتمل على مصفوفة عبارة عن مجموعة كلمات وفيه يطلب البرنامج من المستخدم ادخال
كلمة س فاذا كانت هذه الكلمة احدى الكلمات في مصفوفة الكلمات سم له البرنامج بتكملة العمل
والاستمرار البرنامج في دورة do...while حتى يدخل المستخدم احدى الكلمات الموجودة المصفوفة.

```
{ int i = 0;  
  char name [10];  
  char name [5] [10] = {{"ahmed...."},  
                        {"mohamed"},  
                        {"samy....."},  
                        {"hamdy...."},  
                        {"nabil....."},  
                        };  
  
  do { clrscr ( );  
      cout <<" \n Enter your name ";  
      cin >> name 2;  
      for (i=0; i < 5; i++)  
      {  
          if (strcmp (name?, name [i] == 0)  
              t = 1;  
      }  
  } while (t == 0)  
}
```

String المصفوفة من نوع char

هي string وان الاعلان عن سلسلة حروف string تستعمل كلمة سلسلة حروف مقابل كلمة

```
char name [10];
```

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

```
char a [10] = " A B C - m n p 3 9 r "; اعطاء قيم مباشرتا الى المصفوفة
```

```
cin اعطاء قيم للمصفوفة باستخدام
```

```
for (int i = 0; i < 9; i++)
```

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

OR

```
cout << " Enter your name ";
```

```
cin >> a; ← Ali maged
```

A	l	i		m	a	g	e	d	10
---	---	---	--	---	---	---	---	---	----

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

Example:- اكتب برنامج ليجاد عدد الاحرف الكبيرة ضمن string معينة

```
{ char a [10] = " ";
  int count = 0
  for (int I = 0; I < 10; I++)
  if (a [i] >= 'A' && a [i] < 'Z')
    count ++;
  cout << " the number of capital characters ="
  << count << endl;
```

Example: - write a program to delete spaces from a string

```
{ int p = 0
  cout << " Enter your string ";
  cin >> s;
  for (int I = 0; s [i]; i++)
  if (s [i] != ' ')
  s [p++] = s [i];
  s [p++] = \0;
  cout << s << endl;
```

Example: - write a program to find the length of string

```
{ char s [20]; int n;
  cout <<" Enter your string ";
  cin >> s;
  for (int I = 0; s [i]; i++)
    n ++;
  cout <<" The length of string =" << n << endl;
}
```

Example: - write a program to convert any capital letter to small in a string

```
{ char s [20]; int n;
  cout <<" Enter your string ";
  cin >> s;
  for (int I = 0; s [i]; i++)
    if (s [i] >= 'A' && s [i] <= 'Z')
      s [i] += 32;
  cout << s << endl;
}
```

Example: - write a program to print any prime number in an array?

```
{ int a [10] = {          };
  for (int I =0; I < 10; i++)
    for (int n = 2; n < a [i]; n++)
```

Q: - write a program to delete a number of characters from any position in a string.

```
# include <iosteream.h>
void main ( )
{
  int p, l , sl;
  char s [10] = "          ";
  cout <<" Enter he position and length:";
  cin >> p >> l;
  sl = strlen (s);
  if ((p + l) < sl)
    { for (int i = p + l; i < sl; i++)
      s [i - l] = s [i];
      cout << s;
    }
  else
    cout <<" Input ewor: " << endl;
}
```

2). Arrays المصفوفة ذات البعدين

هي المصفوفة التي ترتب عناصرها على شكل صفوف واعمدة ويتم الاعلان عنها بالشكل التالي

```
int A [5] [10];
```

(اعمدة . ويتم الاشارة او التعامل الى 10 صفوف و (5) مصفوفة ذات بعدين A. وهذا معناه ان المصفوفة
العنصر برقم الصف ورقم العمود.

```
cout << A [1] [2]
```

في حالة المصفوفة for loop ويجب الانتباه الى انه عند التعامل مع المصفوفة لابد من استعمال دوراته
nested for. اما في حالة المصفوفة ذات البعدين فلا بد من استعمال مايسمى بالدورات المتداخلة
for.

$$C = \begin{matrix} & \begin{matrix} j=0 & 1 & 2 \end{matrix} \\ \begin{matrix} i=0 \\ 1 \\ 2 \end{matrix} & \begin{pmatrix} 5 & 4 & 2 \\ 7 & 1 & 7 \\ 2 & 9 & 5 \end{pmatrix} \end{matrix}$$

```
cout << c [0] [0] = 5
```

```
cout << c [2] [1] = 9
```

```
cout << c [2] [2] = 5
```

```
cout << c [1] [2] = 7
```

ادخال قيم الى المصفوفة

```
cout << " input the matrix =";
```

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

```
{ for (int j = 0; j < 3; j++)
```

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

```
}
```

طباعة قيم المصفوفة

```
cout << " input the matrix =\n";
```

```
for (int n = 0; n < 3; n++)
```

```
{ for (int m = 0; m < 3; m++)
```

```
cout << a [n] [m] << "\t";
```

```
cout << endl;
```

```
}
```

اعطاء قيم ابتدائية للمصفوفة

```
int A [3] [4] = {
```

```
{7, 5, 7, 8},
```

```
{2, 3, 4, 5},
```

```
{7, 9, 8, 1},
```

```
};
```

Example: - write a program to add two matrices A and B and the result will be stored in matrix C. Let the matrices is (3×4)

```
# include <iostream.h>
void main ()
{ int A [3] [4], B [3] [4], C [3] [4];
  A [3] [4] = {{1, 2, 3, 4},
              {5, 6, 7, 8},
              {9, 10, 11, 12}};
  B [3] [4] = A
  for (in i = 0; i < 3; i++)
  { for (int j = 0; j < 4; j++)
    {
      C [i] [j] = A [i] [j] + B [i] [j];
      cout << C [i] [j] << "\t";
    }
    cout << endl;
  }
}
```

Example: - write a program to find the average of each **row** of a matrix (3×4)

```
# include <iostream.h>
void main ()
{ int B [3] [4] = {          }
  int sum;
  for (int r = 0; r < 3; r++)
  { sum = 0;
    for (int c = 0; c < 4; c++)
      sum+ = B [r] [c];
    cout << float (sum)/4.0 << endl;
  }
}
```

Example: - write a program to exchange **row2** with **col3** in a matrix (4×4)

```
{ int B [4] [4] = {          };
  int t;
  for (int i = 0; i < 4; i++)
    { t = B [2] [i];
      B [2] [i] = B [i] [3];
      B [i] [3] = t;
    }
  for (int i = 0; i < 4; i++)
  for (int j = 0; j < 4; j++)
    cout << B [i] [j] << "\t";
    cout << endl;
}
```

Example: - Find the max no. a primary diagonal and the min no. in a secondary diagonal.

```
{
  int B [4] [4] = {          };
  int max = B [0] [0];
  int min = B [0] [3];
  for (int i = 1; i < 4; i++)
    if (max < B [i] [i])
      max = B [i] [i];

  for (int i = 1; i < 4; i++)
    if (min > B [i] [i])
      min = B [i] [3-i];

  cout <<" The max =" << max << endl;
  cout <<" The min =" << min << endl;
}
```

Q1:- write a program to exchange the primary and secondary diagonal of matrix A.

Let A is a (4×4) matrix.

Q2:- write a program to convert a two dimensional array into on dimensional array

Q3:- write a program to fill the following matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Q4:- write a program to fill the following matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Q5:- write a program to find the transpose of any matrix.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \longrightarrow A^T = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{pmatrix}$$

Q6:- write a program to computer the sum of secondary diagonal element in a square matrix of type integer.

Q7:- write a program to delete a number of characters from any position in a string

Example3: - write a program to exchange the primary and secondary diagonal of matrix A.

Let A is a (4×4) matrix.

```
# include <iostream.h>
void main ( )
{ int t;
  int A [4] [4] = {{1,  2,  3,  4},
                  {5,  6,  7,  8},
                  {9, 10, 11, 12},
                  {13, 14, 15, 16}};
  for (int i = 0; i < 4; i++)
    { t = A [i] [i];
      A [i] [i] = A [i] [3-i];
      A [i] [3-i] = t;
    }
// O/P
  for (int i = 0; i < 4; i++)
    { for (int j = 0; j < 4; j++)
      cout << A [i] [j] << "\t";
      cout << endl;
    }
}
```

Example4: - write a program to convert a two dimensional array into one dimensional array

```
# include <iostream.h>
void main ( )
{ int A [3] [4] = {{   }};
  int B [12];
  for (int i = 0; i < 3; i++)
    for (int j = 0; j < 4; j++)
      B [i * 4 + j] = A [i] [j];
// O/P
  for (int i = 0; i < 12; i++)
    cout << B [i] << endl;
}
```

Example5: - write a program to fill the following matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
# include <iostream.h>
void main ( )
{
    int A [3] [4];
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 4; j++)
            if (j >= i)
                A [i] [j] = 1;
            else
                A [i] [j] = 0;
    for (int i = 0; i < 4; i++)
        { for (int j = 0; j < 4; j++)
            cout << setw(5) << A [i] [j];
          cout << endl;
        }
}
```

Example6: - write a program to fill the following matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

```
# include <iostream.h>
void main ( )
{ int A [3] [4];
  for (int i = 0; i < 4; i++)
    { for (int j = i; j < 4; j++)
      { A [i] [j] = i + 1;
        A [j] [i] = i + 1;
      }
    }
  for (int i = 0; i < 4; i++)
    { for (int j = 0; j < 4; j++)
      cout << A [i] [j] << "\t";
      cout << endl;
    }
}
```

Example7: - write a program to find the transpose of any matrix.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \longrightarrow A^T = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \end{pmatrix}$$

```
# include <iostream.h>
void main ( )
{
    int t;
    int A [4] [4] = {{    },    };
    for (int i = 0; i < 4; i++)
        for (int j = i+1; j < 4; j++)
            {
                t = A [j] [i];
                A [j] [i] = A [i] [j];
                A [i] [j] = t;
            }
    for (int j = i+1; j < 4; j++)
        { for (int j = 0; j < 4; j++)
            cout << A [i] [j] << "\t";
            cout << endl;
        }
}
```

Example8: - write a program to computer the sum of secondary diagonal element in a square matrix of type integer.

```
# include <iostream.h>
void main ( )
{
    int A [3] [3] = {{1,2,3},{4,5,6},{7,8,9}};
    for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
                }
}
```

Example: - write a program to reverse a Constance of array.

```
#
void main ( )
{ int X [10], T;
  for (int i = 0; i < 10; i++)
    cin >> X [i];
  for (int j = 0; j < (10/2); j++)
    { T = X [j];
      X [j] = X [9-j];
    }
  for (int i = 0; i < 10; i++)
    cout << X [i];
}
```

Example: - write a program to delete an item form array of a specified position:

```
#
void main ( )
{ int a [10] = {          };
  int pos;
  cout <<"Enter position;
  cin >> pos;
  for (int i = pos+1; i < 10 i++)
    a [i-1] = a [i];
    a [9] = 0
  for (int i = 0; i < 10 i++)
    cout <<"a[" << i <<" ] =" << a [i];
}
```

Example: - write a program to insert a item into array at a specified position:

```
{ int a [10] = {1,2,3,4,5,6,7,8,9};
  int item, pos;
  cout <<"Enter your number";
  cin >> item;
  cout <<"Enter your position";
  cin >> pos;
  for (int i = 8; i >= pos; i--)
    a [i+1] = a [i];
    a [pos] = item;
  for (int i = 0; i < 10 i++)
    cout <<"a[" << i <<" ] =" << a [i] << endl;
}
```

Q:- write a program to sort an integer array by using sequential

```
#include <iostream.h>
void main ()
{
    int t;
    int a [10] = {5,9,13,3,2,4,19,1,7,8};
    for (int i = 0; i < 9; i++)
        for (int j = i+1; j < 10; j++)
            {
                if (a [i] > [j])
                    { t = a [i];
                      a [i] = a [j];
                      a [j] = t;
                    }
            }
    for (int i = 0; i < 10 i++)
        cout << a [i] << endl;
}
```

Structure

السجل

من اهم تطبيقات في عالم البرامج تطبيقات قواعد البيانات فمثلا قاعدة بيانات موظفين تمثل بيانات الموظفين في شكل سجلات يتكون من مجموعة حقول ولو ان ذلك خبرة بأخذ برامج قواعد البيانات مثل (و دائما تحتاج fields) والسجل ينقسم الى حقول (records) فستعرف ان الملف ينقسم الى سجلات (dbase) للتعامل مع السجل كوحدة وكذلك مع الحقول كوحدة.

استعمال السجل

خطوات استعمال السجل

1. انشاء السجل .
2. تحديد الحقول المطلوبة .
3. الاعلان عن متغير من نوع هذا السجل .
4. التامل مع حقول هذا السجل .

Example: -

```
1) # include <iostream.h>
2) void main ( )
3) {
4)     struct data                (1) الاعلان عن السجل
5)     {
6)         int num;                (2) محتويات السجل
7)         char stat;
8)     };
9)     stud data stud;            (3) اعلان ن متغيرات من نوع السجل data.
10)    stud . num = 5;
11)    stud . stat = 't';         (4) اعطاء قيم للمتغيرات داخل السجل باستخدام dot operat (.).
12)    cout <<" stud . num ="<< stud . num;
13)    cout <<"\n stud . stat =" << stud . stat;
14) }
```

- ✓ data واعطاء هذا السجل اسم وهو struct في السطر رقم (4) يبدأ لانشاء السجل وذلك باستعمال كلمة ممكن ان تكون اي كلمة اخرى data وكلمة
- ✓ وفي السطرين (6 و 7) اعلان عن حقول السجل وهو { في السطر رقم (5) تبدأ مكونات السجل بالاقوس } عبرة عن متغير من نوع صحيح ومتغير من نوع حرف وينتهي السجل في السطر رقم (8) بالاقوس stud وبالتالي اخذ المتغير stud في السطر رقم (9) يتم اعلان عن متغير من نوع السجل وهو المتغير stat. من نوع صحيح وكذلك عنصر من نوع حرف وهو num بنفس التركيب فاصبح له عنصر اسم
- ✓ في السطرين (10 و 11) تم اعطاء قيم لحقول السجل ولكن الملاحظ انه للتعامل مع حقل في سجل يتم stud . num كتابة اسم الحقل مسبقا باسم السجل التابع له وببيها نطة بالصورة
- ✓ وفي السطرين (12 و 13) يتم طباعة قيم الحقول ف السجل وب نفس الاسلوب اي كتابة اسم الحقل مسبقا اسم السجل وبينهما النقطة للاشارة ان ها الحقل تابع لهذا السجل .

ملاحظة

يمكن الاعلان عن اكثر من متغير من نوع السجل كما يحدث مع انواع البيانات الاخرى.

```
struct data
```

```
{ int num;
```

```
char ch;
```

```
} stud1 , stud2; اعلان عن المتغيرات بطريقة اخرى
```

Example1: - complex number

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

```
struct complex
```

```
{ float R, I;
```

```
};
```

```
void main ()
```

```
{
```

```
complex C1, C2, C3;
```

```
cout <<" Enter Real part of C1.R ";
```

```
cin >> C1.R;
```

```
cout <<" Enter Imaginary part of C1.I ";
```

```
cin >> C1.I;
```

```
cout <<" Enter Real part of C2.R ";
```

```
cin >> C2.R;
```

```
cout <<" Enter Imaginary part of C2.I ";
```

```
cin >> C2.I;
```

```
C3.R = C1.R + C2.R;
```

```
C3.I = C1.I + C2.I;
```

```
cout <<" The sum of Real part C3.R =" << C3.R << endl;
```

```
cout <<" The sum of Imaginary part C3.I =" << C3.I;
```

```
}
```

(1 اعلان عن السجل

(2 محتويات السجل (الحقول)

(3 نهاية السجل

Example2: - point

```
# include <iostream.h>
struct point
{
    int x, y;
} P1, P2, P3;
void main ()
{
    cout <<" Enter point one P1="
    cin >> P1.x >> P1.y;
    cout <<" Enter point two P2="
    cin >> P2.x >> P2.y;
    P3.x = P1.x + P2.x;
    P3.y = P1.y + P2.y;
    cout <<" the point one P3=";
    cout << P3.x << "\ t" << P3.y;
}
```

Example3: - distance

```
# include <iostream.h>
struct distance
{
    float inches;
    int feet;
};
void main ()
{
    distance d1,d2,d3;
    cout <<" Enter The distance d1=";
    cin >> d1.feet >> d1.inches;
    cout <<" Enter The distance d2=";
    cin >> d2.feet >> d2.inches;
    d3.inches = d1.inches + d2.inches;
    d3.feet = 0;
    if (d3.inches >= 12)
        { d3.inches -= 12;
          d3.feet ++;
        }
    d3.feet = d3.feet + d1.feet + d2.feet;
    cout << d3.feet << ".";
    cout << d3.inches;
}
```

Example4: - stud

```
# include <iostream.h>
struct stud
{ int m;
  int c;
  int l;
  float av;
};
void main ( )
{ stud ali, ahmad;
  ali.m = 70;
  ali.c = 60;
  ali.l = 90;
  ali.av = (ali.m + ali.c + ali.l)/3;

  ahmad.m = 77;
  ahmad.c = 80;
  ahmad.l = 93;
  ahmad.av = (ahmad.m + ahmad.c + ahmad.l )/3;
  cout <<"ali"<< ali.m << "\t" << ali.c << "\t" << ali.l << "\t" << ali.av <<endl;
  cout <<"Ahmad"<<ahmad.m<<"\t"<<ahmad.c<<"\t"<<ahmad.l<<"\t"<<ahmad.av;
}
```

Nested Structure

Example5: -

```
# include <iostream.h>
    struct distance
    {
        int feet;
        float inches;
    };
    struct Room
    {
        distance length;
        distance width;
    };
void main ()
{ Room dining; // define a room.
  dining.length.feet = 13;
  dining.length.inches = 6.5;
  dining.width.feet = 10;
  dining.width.inches = 0.0;
  float l = dining.length.feet + dining.length.inches /12;
  float w = dining.width.feet + dining.width.inches /12;
  cout << "\n dining room area is" << l * w << "square feet";
}
```

Functions

الدوال

وهي دوال عام يستطيع اي C++ هي دوال مبنية في لغة cout) و (cin الدوال التي استخدمت مسبقا المرونة في الاستخدام ولذلك يمكنك انشاء دوال مثل الدوال القياسية C++ مبرمج استخدامها . من مزايا لغة الموجودة في صلب اللغة لتؤدي وظائف مختلفة او متشابهة. والدوال هي عبارة عن برنامج صغير (او مجموعة تعليمات تؤدي غرض او وظيفة معينة) ويخصص لهذا () main البرنامج اسم ويتم استدعائه داخل الدالة الرئيسية.

ويحقق استخدام الدالة مزايا عديدة منها :-
عدم تكرار الخطوات داخل البرنامج حيث يتم انشاء الدالة مرة واحدة ثم يتم استدعائها اكثر من مرة عند الحاجة اليها.
باستخدام ادالة يصبح البرنامج اكثر وضوحا حيث ياخذ البرنامج الشكل التركيبي فيصبح بالشكل التالي :-

```
# include <filenam.h>
1- function declarations;          1- اعلان عن الدالة
main ( )
{
2- function calling;              2- استدعاء الدالة
}
3- function definition ( )        3- تعريف الدالة
{ ;
} ;
```

الصيغة العامة للاعلان عن الدالة

Type	Name	(list of parameters)
↑	↑	↑
(نوع الدالة)	(اسم الدالة)	(المتغيرات او المعاملات)
O/P		I/P

Example: -

```
# include <iosteream.h>
# include <conio.h>
void line2 (void); ← (1) اعلان عن الدالة
void main ()
{ clrscr ( );
  line2 ( ); ← (2) استدعاء الدالة function calling
  cout << "Computer Dep." << endl;
  line2 ( ); ←
}
void line2 (void) ← (3) تعريف الدالة function definition
{ for (int j=0; j<=40; j++)
  cout << " * ";
  cout << endl;
}
```

- 1- function declaration `main` () يسمى بالاعلان عن الدالة يكون ذلك قبل الدالة الرئيسية `void line2 ()` في نهاية هذا الجزء لانه اعلان , ويكون بالصورة (;) ونلاحظ الفاصلة المنقوطة هي نوع الدالة التي سوف نتكلم عنها فيما بعد بالتفصيل. `void` وكلمة `line2 ()` وفيه يتم كتابة () `main` يظهر في اي مكان داخل الدالة الرئيسية `function calling` . وكانها برنامج صغير وبعد القوس نبدأ {محتويات الدالة ونبدأ في البرنامج من القوس بطباعة علاقة `cout` تقو الدالة `for` بكتابة تعليمات الدالة والدالة هنا عبارة عن دواراة { (*) (40مرة) عند استدعائها (الدالة) وتنتهي الدالة بالقوس

تنفيذ البرنامج

```
* * * * * 1 0
* * * * * 40
Computer Dep.
* * * * * 1 0
* * * * * 40
```

انواع الدوال

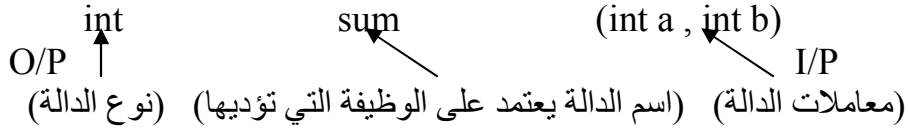
نوع الدالة تعتمد على القيمة التي تعيدها الدالة

void main ()

هي احد انواع الدوال وهناك انواع اخرى من الدوال نوضحها فيما يلي void وان كلمة

- int function دوال تعيد قيمة صحيحة
- float function دوال تعيد قيمة حقيقية
- string function دوال تعيد قيمة حرفية
- char function دوال تعيد قيمة حرفية
- void function دوال لاتعيد قيمة
- struct function تسمى structure دوال تعيد قيمة من نوع

EX:-



استدعاء الدالة

يتم استدعاء الدالة اما بمعاملات او بدون معاملات . وتكون بدون معاملات . وتكون بدون معاملات كما في وبدون معاملات معناه عدم كتابة قيم بين اقواس الدالة. (line الدالة line2 (). البرنامج الموجود يوضح استدعاء الدالة بمعاملات وذلك بعد تعديل الدالة

```
# include <iosteream.h>
void line2 (int no);
void main ( )
{
    line2 (30);
    cout <<"computer Dep ";
    line2 (50);
}
void line2 (int no)
{ for (int i=0 ; i<=no ; i++)
  cout << "*"
  cout << endl;
}
```

Example: - لحساب مجموع عددين

```
# include <iostream.h>
int sum (int a , int b)
void main ( )
{ int Z, x=10, y=40;
  Z = sum (x,y);
  cout << Z;
}
int sum (int a , int b)
{ int s
  s = a + b;
  return s;
}
```

ملاحظة: القيمة التي تعيدها الدالة الى البرنامج فاذا كانت القيمة الي تعيدها الدالة int كان نوع الدالة int واذا كانت float. اما في حالة void فلا تستخدم كلمة return لانها لا تعيد اي قيمة.

Example: - الدالة من نوع int

```
#
int sqr (int);
int qup (int);
void main ( )
{ int s, qu, no=5;
  s = sqr (no);
  qu = qup (no);
  cout << "The square no. =" << s;
  cout << "\n the quq. no. =" << qu;
}
int sqr (int a)
{
  return (a*a);
}
int qup (int qa)
{
  return (q*q*q);
}
```

Example: - الدالة من نوع float

```
# include <iostream.h>
float ava (float x, float y);
void main ( )
{
    float no1, no2, av;
    cout <<"Enter no1 = ";
    cin >> no1;
    cout <<"Enter no2 = ";
    cin >> no2;
    av = ava (no1 , no2);
    cout <<"The average =" << av;
}
float ava (float x, float y)
{
    float z
    z = (x + y)/2.0;
    return z;
}
```

Example: - write program that computer the max of two number by using the max (). The prototype of max () is int max (int a, int b);
The function max () return the maximum of a and b;

```
#
int max (int , int);
void main ( )
{ int x , y;
  cout << " Enter the first no. = second no. =";
  cin >> x >> y;
  cout <<"the max no. is =" << max (x , y);
}
int max (int a, int b)
{ if (a > b)
  return a;
  else
  return b;
}
```

Example2: - write program that calculates the area of circle by using the function `carea ()`. The return the area of circle of radius `r`. the prototype of `carea ()` is `float carea`

```
# include <iostream.h>
float carea (float r);
void main ( )
{
    float rad;
    cout <<" Enter the radus of circule =";
    cin >> rad;
    cout <<" The area of circule is "<< carea(rad);
}
float carea (float r)
{ float a;
  a = (r * r * 3.145);
  return a;
}
```

Example3: - write program to find the dashed area by using function `carea ()`.

```
# include <iostream.h>
float carea (float);
void main ( )
{
    float r1 , r2 , dsa;
    cout <<" Enter the radus of circules r1,r2 =";
    cin >> r1 >> r2;
    dsa = carea (r2) - carea (r1);
    cout <<" The dashed area =" << dsa;
}
float carea (float r)
{ float a;
  a = (r * r * 3.145);
  return a;
}
```

Example4: - write program that calculates the distance between two points. Each point represents by calculated by using function distance (). The prototype of the function is int distance (int x1, int y1, int x2, int y2);

```
# include <iostream.h>
# include <math.h>
int distance (int x1, int y1, int x2, int y2);
void main ( )
{ int x1, y1, x2, y2;
  cout <<" Enter the value of x1, y1, x2 and y2 =";
  cin >> x1 >> y1 >> x2 >> y2;
  cout <<" The distance =" << distance (x1, y1, x2, y2);
}
int distance (int x1, int y1, int x2, int y2);
{ int dx, dy;
  dx = x2 - x1;
  dy = y2 - y1;
  return sqrt (dx * dx + dy * dy);
}
```

Example5: - write program to computer the max of n number by using function n max (). The prototype of n max () is int n max (int n); the function return the maximum of n number where the number are entered with n function.

```
# include <iostream.h>
int nmax (int n);
void main ( )
{ int num;
  cout <<" Enter the number of numbers =";
  cin >> num;
  cout <<" The max.no. =" << nmax (num);
}
int nmax (int n)
{ int x, max = -1000;
  for (int i = 0; i < n; i++)
  { cout <<" Enter new no. =";
    cin >> x;
    if (x > max)
      max = x;
  }
  return max;
}
```

Example6: - write program to find the binary of any positive integer number by using function binary (), the prototype of binary () is.

```
int binary (int x);
```

```
# include <iostream.h>
int binary (int x);
void main ( )
{ int num;
  cout <<" Enter your number =";
  cin >> num;
  cout <<" The binary =" << binary (num);
}
int binary (int a);
{ if (a < 0)
  return 0;
  int b = 0, w = 1, d;
  while (a)
  { d = a % 2;
    b = b + d * w;
    w = w * 10;
    a = a / 2;
  }
  return b;
}
```

Example7: - write program that calculates the average of n number by using function average (). The prototype of average () is.

```
float average (int n);
```

```
# include <iostream.h>
float average (int n);
void main ( )
{ int num;
  cout <<" Enter the number of numbers =";
  cin >> num;
  cout <<" The average =" << average (num);
}
float average (int n);
{ float x, sum = 0;
  for (int i=0; i<n; i++)
  { cout <<" Enter new number = ";
    cin >> x;
    sum = sum + x;
  }
  return (sum / n);
}
```

Example8: - By using overloaded function, write a program to find area of square and rectangular. The prototype of function are :

```
float area (int);  
float area (int, int);
```

```
float area (int);  
float area (int, int);  
void main ()  
{ int x, x1, x2;  
  cout <<" Enter the side of square =";  
  cin >> x;  
  cout <<" Enter the length and width of rectangle =";  
  cin >> x1 >> x2;  
  cout <<" The area of square " << area (x) << endl;  
  cout <<" The area of rectangular = " << area (x1, x2);  
}  
float area (int l)  
{  
  return l * l;  
}  
float area (int l, int w)  
{  
  return l * w;  
}
```

Example9: - write program to test the number

```
# include <iostream.h>  
int test (int);  
void main ()  
{ int num;  
  cout <<" Enter your integer numbers =";  
  cin >> num;  
  if (test (num) == 1)  
    cout <<" the no. is positive ";  
  else if (test (num) == 0)  
    cout <<" the number is zero ";  
  else  
    cout <<" the number is negative ";  
}  
int test (int a)  
{ if (a > 0)  
  return 1;  
  else if (a < 0)  
    return -1;  
  else  
    return 0;  
}
```

Example10:- write program that transform a small letter into capital letter by using to capital (). The prototype is char to capital (char ch);

```
# include <iostream.h>
char to capital (char);
void main ( )
{ char c;
  cout <<" Enter your character =";
  cin >> c;
  cout <<" The capital =" << tocapital (c);
}
char tocapital (char ch)
{
  if (ch >= 'a' && ch <= 'z')
    return (ch - 32);
  else
    return ch;
}
```

character	AscII
0 - 9	48 - 57
A - Z	65 - 90
a - z	97 - 122

Note : فرق بين a , A هو 32
 $A = a - 32$
 $Z = z - 32$

Example11:- write program to test the character if it is small letter or not by using issmall (). The prototype is int issmall (char c);

```
# include <iostream.h>
int issmall (char c);
void main ( )
{
  char ch;
  cout <<" Enter the character =";
  cin >> ch;
  if (issmall (ch))
    cout <<" The character is small " << endl;
  else
    cout <<" The character is not small ";
}
int issmall (char c)
{
  if (c >= 'a' && c <= 'z')
    return 1;
  else
    return 0;
}
```

Example12:- write program to test the character if it is numeric digit or not by using is n digit ();

The prototype is int isndigit (char c);

```
# include <iostream.h>
int isndigit (char c);
void main ( )
{
    char ch;
    cout <<" Enter the character =";
    cin >> ch;
    if (isndigit (ch))
        cout <<" The character is numeric digit. " << endl;
    else
        cout <<" The character is non numeric digit ";
}
int isndigit (char c);
{
    if (c >= '0' && c <= '9')
        return 1;
    else
        return 0;
}
```

Example13:- delete a space character from a string by using Del-cher function

```
# include <iostream.h>
void Delchar (char s[ ], char);
void main ( )
{ char a [30];
  cout <<" Enter your string ";
  cin >> a;
  Delchar (a [ ], ' ');
  cout << a << endl;
}
void Delchar (char s [ ], char c).
{
    for (int i=0, s [i], i++)
        if (s [i] == c)
            for (int j=i; s[j]; j++)
                s[j] = s [j+1];
}
```

Example14:- To replace any char. With other by function

```
void replace (char s[ ], char, char).
```

```
# include <iosteream.h>
void Delchar (char s[ ], char a, char b)
{ for (int i=0; s[i]; i++)
  if (a == s[i])
    s[i] = b;
}
void main ( )
{ char n [20];
  cout <<" Enter your string =";
  cin >> n;
  replace (n[ ], ' ', '*' )
  cout << n << endl;
}
```

Q: - An integer is said to be prime if it is divisible only by 1 and itself. Write a function that determines if a number is prime; and use this function in a program that determines and prints all the prime number between 10 and 50.

```
int prime (inta);
```

if a is prime, the the function returns 1, else it returns 0.

```
# include <iosteream.h>
int prime (int a);
void main ( )
{
  for (int i=1; i<50; i++)
  {
    if (prime (i))
      cout << i <<" prime " << endl;
  }
}
int prime (int a)
{
  for (int j=2; j<a; j++)
    if ((a % j) ==0)
      return 0,
    else
      return 1;
}
```

Call by Reference

المتغير المرجعي

من اجل يمرير بالمرجع يجب وضع & بعد نوع البيانات في الوسيطة . ان الغاية من الدلة هو تغير قيمة المتغير المرسل اليها.

```
# include <iostream.h>
void doubleIt (num);
void main ( )
{ int num;
  cout <<" Enter number =";
  cin >> num;
  doubleIt (num);
  cout <<" The number doubled =" << num;
}
void doubleIt (int x)
{
  x* = 2;
}
```

←----->

```
void doubleIt (int &);
void main ( )
{ int num;
  cin >> num;
  doubleIt (num);
  cout << num;
}
void doubleIt (int & x)
{
  x* = 2;
}
```

↑ استخدام المرجع