

Algorithms and Flow charts

لحل أي مسألة حسابية أو منطقية عبر استخدام الحاسبة لابد ان يتم تعريف المسألة ثم تحديد معالمها وتحليل عناصرها ،
بالتالي يجب مراعاة ما يلي:

- 1- تحديد البيانات المعطاة.
 - 2- توضيح الهدف او الغاية من حل المسألة. 3- تحديد النتائج المطلوبة.
 - 4- وضع طريقة للحل على شكل مجموعة من الخطوات المتسلسلة والمتعاقبة.
- ان مجموعة الخطوات المتسلسلة التي يتم وضعها لحل المسألة تدعى الخوارزمية Algorithm ويمكن تمثيل هذه الخطوات باستخدام اشكال رمزية لها معاني محددة ومتراطة فيما بينها بشكل مخطط وصفي تسلسلي يدعى المخطط الانسيابي (Flow Chart).

الخوارزميات (Algorithms)

ما هي الخوارزمية :

هي عبارة عن مجموعة من الخطوات التي تؤدي عند تنفيذها الى الحل المطلوب. لماذا سميت الخوارزمية بهذا الاسم؟

سميت بهذا الاسم نسبة للعالم محمد بن موسى الخوارزمي الذي عاش في القرن التاسع ونبغ في علم الجبر. ملاحظات هامة عن الخوارزميات : الخطوات العامة لبناء الخوارزمية:

- 1- لكل خوارزمية مجموعة من الخطوات تمثل العمليات التي نجريها للوصول الى الناتج.
- 2- يجب أن يكون لكل خوارزمية بداية ونهاية.
- 3- غالبا نستخدم في الخوارزمية الأمر (اقرا) لإدخال البيانات أو المعطيات.
- 4- غالبا نستخدم في الخوارزمية الأمر(احسب) لحساب معادلة أو صيغة رياضية .
- 5- ند نستخدم في الخوارزمية الأمر (اجعل) في حالة إسناد قيمة إلى متغير أو في الصيغ لمنطقية.
- 6- ند نستخدم في الخوارزمية الأمر (انتقل إلى) للانتقال إلى خطوة سابقة أو لاحقة .
- 7- ند نستخدم في الخوارزمية الأمر (إذا كان) للدلالة على عملية مقارنة أو الشرط ويجب أن يتلو عملية المقارنة عمليتين للدلالة على تحقق الشرط التي تبدأ ب(فان ، فاحسب ، فاجعل، فانقل إلى) و عند عدم تحققه التي تبدأ ب(وإلا فاحسب،وإلا فانقل إلى ، وإلا فاجعل).
- 8- ند نستخدم في الخوارزمية الرمز(*) كعلامة للضرب، والرمز (/) كعلامة للقسمة ، والرمز (+) للجمع ، والرمز(-) للطرح.
- 9- غالبا نستخدم في الخوارزمية الأمر (اطبع) لإخراج المطلوب حسابه.
- 11- تؤدي العمليات بمجملها إلى حل المسألة الحل الصحيح.

11- ليس من الضروري أن نستخدم كل الأوامر الموجودة في أعلاه لحل خوارزمية معينة (حيث أن الاستخدام يتحدد حسب طبيعة المسألة).

مثال رقم (1) : اكتب برنامج يقوم باستقبال درجة الحرارة بالنظام المئوي (Celsius) ويحولها الى النظام الفهرنهايتي (Fahrenheit) ، علما بان معادلة التحويل هي :

$$\text{Fahrenheit} = 9/5 * \text{Celsius} - 32$$

الحل :

المعطيات : درجة الحرارة بالنظام المئوي ، ومعادلة التحويل. المطلوب : درجة الحرارة بالنظام الفهرنهايتي.

1- البداية

2- اقرا درجة الحرارة بالنظام المئوي (Celsius) .

3- احسب درجة الحرارة بالنظام الفهرنهايتي (Fahrenheit= 9/5 * Celsius -32)

4- اطبع قيم Celsius و Fahrenheit .

5- النهاية

مثال رقم (2): اكتب خوارزمية لقراءة عددين وايجاد حاصل جمعهما؟ الحل :

المعطيات : العدد الاول (س) ، العدد الثاني (ص) المطلوب :

حساب المجموع (م) ، حيث م = س+ص

1- البداية

2- اقرا العدد الاول (س)

3- اقرا العدد الثاني (ص)

4- احسب م = س+ص

5- اطبع (م)

6- النهاية

المخطط الانسيابي (Flow Chart)

ما هو المخطط الانسيابي ؟

هو عبارة عن مجموعة من الأشكال الهندسية بينها أسهم لتوضيح عمل الخوارزمية، ويمتاز هذا المخطط بوضوح أكثر من الخوارزمية وخاصة في توضيح عمل الحلقات والشروط المنطقية.

الأشكال الهندسية المستخدمة في المخطط الانسيابي :

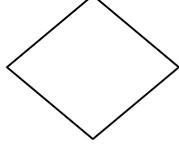
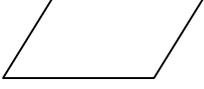


1- الشكل البيضوي : يستخدم للبداية (START) والنهاية (END).

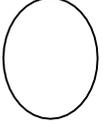


2- المستطيل : يستخدم للعمليات الحسابية وتعيين قيم المتغيرات (MAKE, CALC).

3- متوازي الأضلاع : للقراءة والطباعة (READ,PRINT).



4- شكل المعين : يستخدم للمقارنة والعمليات المنطقية (IF).



5- الدائرة: تستخدم لتوضيح نقاط توصيل المخطط الانسيابي ، وغالبا في حالة المخططات ذات الاحجام الكبيرة.

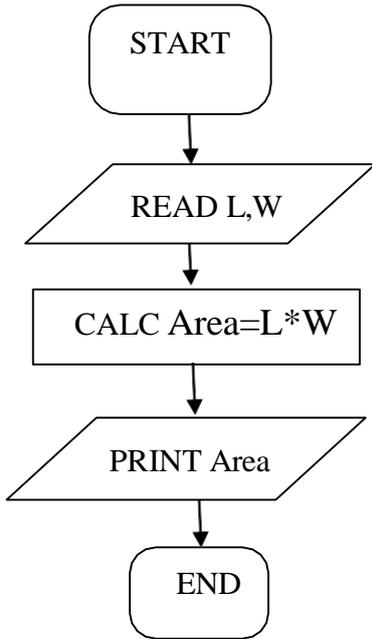


6- السهم يستخدم للتوصيل بين الأشكال الهندسية السابقة وللانتقال من خطوة إلى أخرى وتوضيح مسار المخطط.

مثال رقم (3) : اكتب خوارزمية الحل وارسم المخطط الانسيابي لقراءة طول وعرض مستطيل وحساب مساحته ؟
الحل :

المعطيات : طول المستطيل (L) ، عرض المستطيل (W).
المطلوب : مساحة المستطيل $Area=L*W$

المخطط الانسيابي



الخوارزمية :

1- البداية

2- اقرأ (L) و (W)

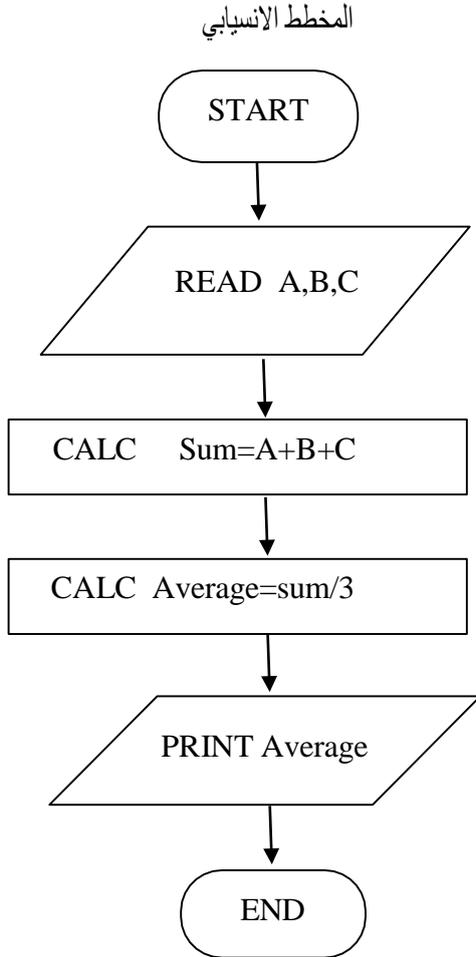
3- احسب $Area=L*W$

4- اطبع قيمة (Area)

5- النهاية

مثال رقم (4): اكتب خوارزمية الحل وارسم المخطط الانسيابي لبرنامج يقوم بطباعة معدل ثلاث درجات ؟
المعطيات : قراءة ثلاثة درجات (A,B,C)
المطلوب : جمع الأعداد الثلاثة ثم تقسيم المجموع على 3

الحل :
الخوارزمية



- 1- البداية
- 2- اقرأ الدرجات الثلاث (A,B,C)
- 3- احسب قيمة المجموع (sum=A+B+C) -4
- 5- اطبع المعدل (Average=sum/3) -5
- 6- النهاية

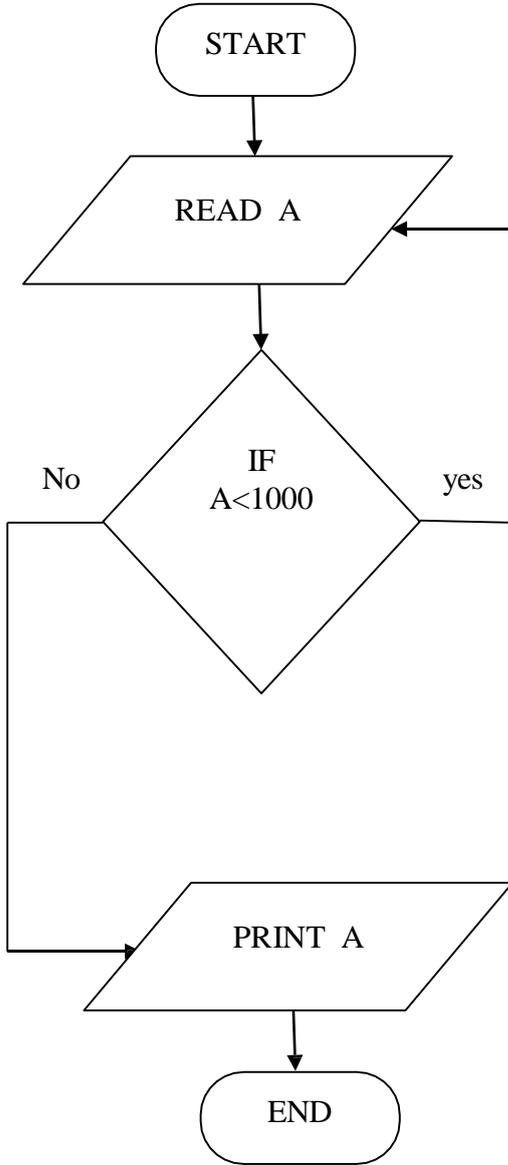
مثال رقم 5 : اكتب خوارزمية الحل وارسم المخطط الانسيابي لبرنامج يقوم بقراءة عدد واحد فقط في كل مرة ويقوم بمقارنة العدد مع الرقم 1111 فإذا كان العدد اقل من 1111 يقوم بإعادة القراءة مرة أخرى لعدد آخر وإلا فإنه يقوم بطباعة العدد ؟

المعطيات : قراءة عدد واحد في كل مرة (A)
المطلوب: مقارنة العدد مع 1111 فإذا كان اقل من 1111 نقوم بإعادة القراءة وإلا يقوم البرنامج بطباعة العدد.

المخطط الانسيابي

الخوارزمية :

- 1- البداية
- 2- اقرأ العدد A
- 3- إذا كان $(A < 1000)$
- 4- فانتقل للخطوة رقم 2
- 5- وإلا فاطبع A
- 6- النهاية

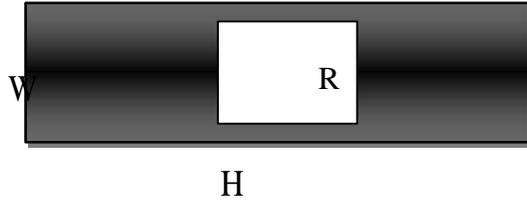


مثال رقم 6 : اكتب خوارزمية الحل وارسم المخطط الانسيابي لبرنامج يقوم بإيجاد وطباعة مساحة الشكل الغامق في الشكل التالي؟

علما أن:

مساحة المربع = طول الضلع R^2

مساحة المستطيل = الطول (H) * العرض (W)



الخوارزمية :

1- البداية

2- اقرأ R,H,W

3- احسب

Square Area= R^2

4- احسب

Rectangle Area= $H*W$

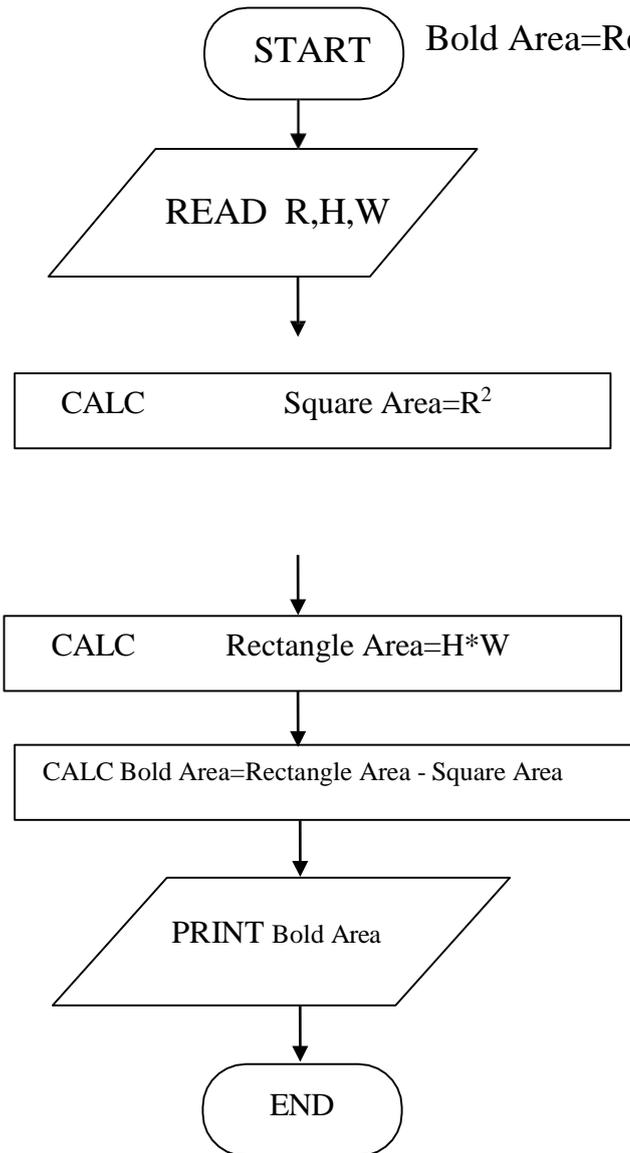
5- احسب

Bold Area=Rectangle Area - Square Area

6- اطبع 7- قيمة Bold Area

النهاية

المخطط الانسيابي

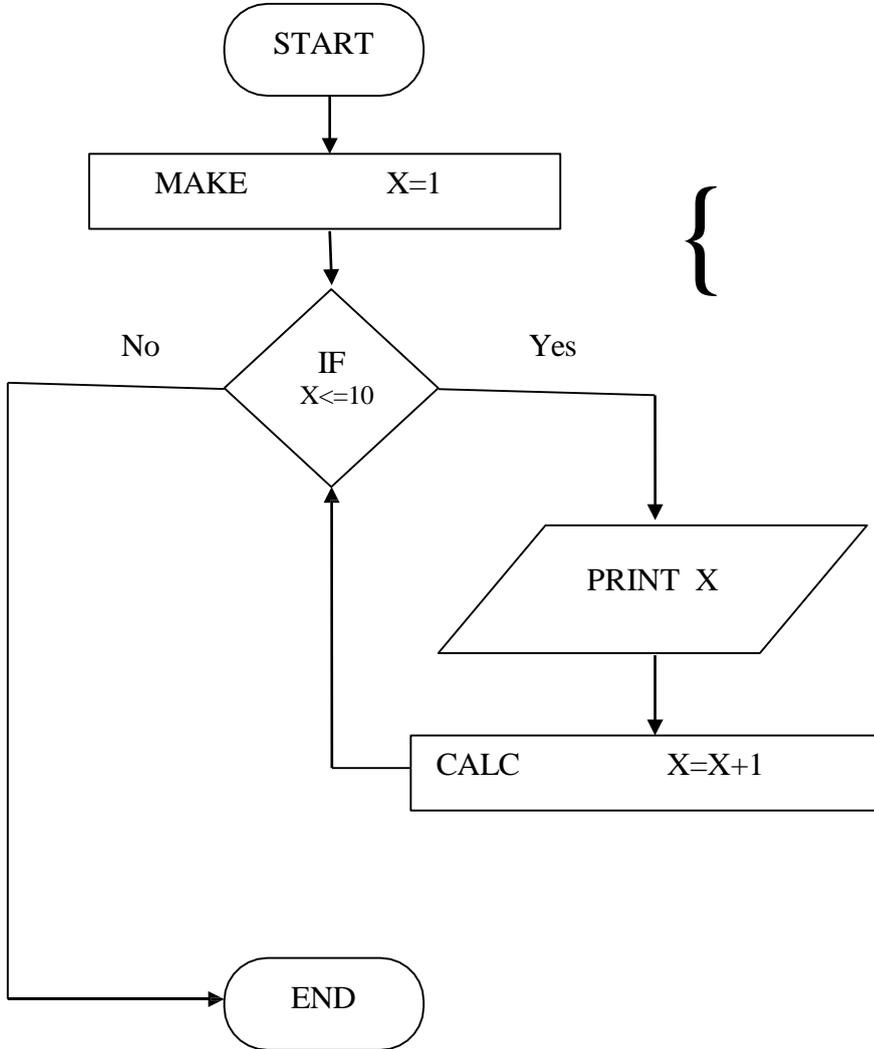


مثال رقم 7 : اكتب خوارزمية الحل وارسم المخطط الانسيابي لبرنامج يقوم بطباعة الأعداد المحصورة بين (11-1) ؟

المعطيات : اجعل $X=1$

المطلوب : طباعة الأعداد من 1 إلى 11

المخطط الانسيابي الخوارزمية :



- 1- البداية
- 2- اجعل $X=1$
- 3- إذا كان $X<10$
- 4- فاطبع X
- 5- فاحسب $X=X+1$
- 6- فانتقل للخطوة 3
- 7- وإلا فانتقل للخطوة 8
- 8- النهاية

تمارين إضافية : التمارين التالية مطلوب حلها كواجب بيتي وبناقش الحل داخل المحاضرة .

تمرين رقم 1: اكتب الخوارزمية وارسم المخطط الانسيابي لبرنامج يقوم بقراءة ثلاثة أعداد (A,B,C) ومعرفة العدد الأكبر بينها؟

تمرين رقم 2 : اكتب الخوارزمية وارسم المخطط الانسيابي لبرنامج يقوم بقراءة عدد وطباعة كلمة Positive إذا كانت قيمة العدد اكبر من أو تساوي صفر ، وكلمة Negative إذا كان العدد اقل من الصفر ؟

تمرين رقم 3: اكتب الخوارزمية وارسم المخطط الانسيابي لبرنامج يقوم بطباعة الأعداد الزوجية المحصورة بين (1 2- 111) ؟

تمرين رقم 4 : اكتب الخوارزمية وارسم المخطط الانسيابي لبرنامج يقوم بطباعة الأعداد الفردية بين (11-29) ؟

تمرين رقم 5 : اكتب الخوارزمية وارسم المخطط الانسيابي لبرنامج يقوم باستقبال عددين ، ويستبدل أماكنهما في الذاكرة ؟

Input output statements

Input output statements

- The most important input output statements are:
cin, cout instructions

For output we use cout and the insertion operator << as in the following syntax

Syntax:

```
cout<<output list;
```

Ex:

```
cout<<"my name is adel";
```

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

```
cout<<"x="<<x;
```

Input output statements

For input we use cin and the extraction operator >> as in the following syntax

Syntax:

```
cin>>input list;
```

Ex:

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

```
cin>>a;
```

example

Ex1) Write a program to find the value of y where

$$Y=x+3b-c$$

solution

```
#include <iostream>
void main ()
{
    int x,b,c,y;
    cout << "Give the value of x " ;
    cin>>x;
    cout << "Give the value of b " ;
    cin>>b;
    cout << "Give the value of c " ;
    cin>>c;
    y=x+3*b-c;

    cout << "the value of y= "<<y ;

}
```

example

Ex2) Write a program to find the value of y where

$$Y=a-3b+c$$

$$C=3x+7$$

solution

```
#include <iostream>
void main ()
{
    int x,b,c,y,a;
    cout << "Give the value of x " ;
    cin>>x;
    c=3*x+7;
    cout << "Give the value of b " ;
    cin>>b;
    cout << "Give the value of a " ;
    cin>>a;
    y=a-3*b+c;

    cout << "the value of y= "<<y ;

}
```

C - IF...ELSE STATEMENT

http://www.tutorialspoint.com/cprogramming/if_else_statement_in_c.htm

Copyright © tutorialspoint.com

An **if** statement can be followed by an optional **else** statement, which executes when the boolean expression is false.

Syntax:

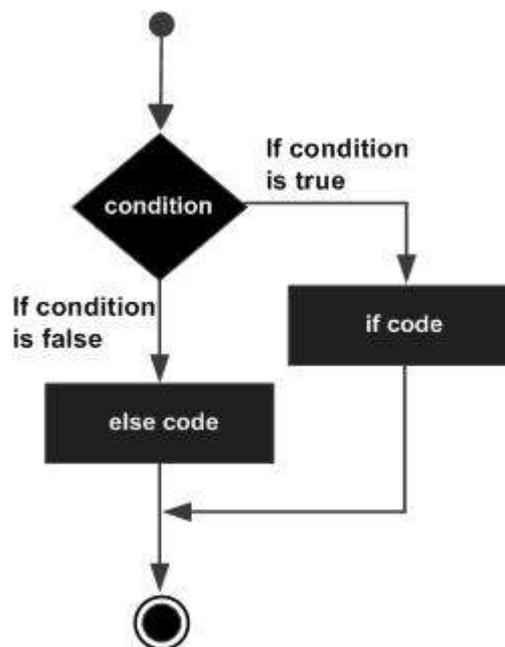
The syntax of an **if...else** statement in C programming language is:

```
if(boolean_expression)
{
    /* statement(s) will execute if the boolean expression is true */
}
else
{
    /* statement(s) will execute if the boolean expression is false */
}
```

If the boolean expression evaluates to **true**, then the **if block** of code will be executed, otherwise **else block** of code will be executed.

C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

Flow Diagram:



Example:

```
#include <stdio.h>

int main ()
{
    /* local variable definition */
    int a = 100;

    /* check the boolean condition */
    if( a < 20 )
    {
        /* if condition is true then print the following */
        printf("a is less than 20\n" );
    }
}
```

```

else
{
    /* if condition is false then print the following */
    printf("a is not less than 20\n" );
}
printf("value of a is : %d\n", a);

return 0;
}

```

When the above code is compiled and executed, it produces the following result:

```

a is not less than 20;
value of a is : 100

```

The if...else if...else Statement

An **if** statement can be followed by an optional **else if...else** statement, which is very useful to test various conditions using single if...else if statement.

When using if , else if , else statements there are few points to keep in mind:

- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of the remaining else if's or else's will be tested.

Syntax:

The syntax of an **if...else if...else** statement in C programming language is:

```

if(boolean_expression 1)
{
    /* Executes when the boolean expression 1 is true */
}
else if( boolean_expression 2)
{
    /* Executes when the boolean expression 2 is true */
}
else if( boolean_expression 3)
{
    /* Executes when the boolean expression 3 is true */
}
else
{
    /* executes when the none of the above condition is true */
}

```

Example:

```

#include <stdio.h>

int main ()
{
    /* local variable definition */
    int a = 100;

    /* check the boolean condition */
    if( a == 10 )
    {
        /* if condition is true then print the following */
        printf("Value of a is 10\n" );
    }
    else if( a == 20 )
    {
        /* if else if condition is true */

```

```
    printf("Value of a is 20\n" );
}
else if( a == 30 )
{
    /* if else if condition is true */
    printf("Value of a is 30\n" );
}
else
{
    /* if none of the conditions is true */
    printf("None of the values is matching\n" );
}
printf("Exact value of a is: %d\n", a );

return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
None of the values is matching
Exact value of a is: 100
```

Loops

```
# include<iostream.h>
void main()
{int x,m=1,i;
for (i=0;i<=10;i++)
{ cin>>x;
  If (x<0)
      if(m==1) m=x else if (x>m) m=x;
}
  cout<<m;
}
```

WHILE LOOP

- The syntax of **while** statement :
while (loop repetition condition)
statement
- **Loop repetition condition** is the condition which controls the loop.
- The *statement* is repeated as long as the loop repetition condition is **true**.
- A loop is called an **infinite loop** if the loop repetition condition is always true.

Condition



```
while (i < 5)
{
    cout << "Please input a number: ";
    cin >> Num1;

    Total = Total + Num1;
    cout << endl;

```



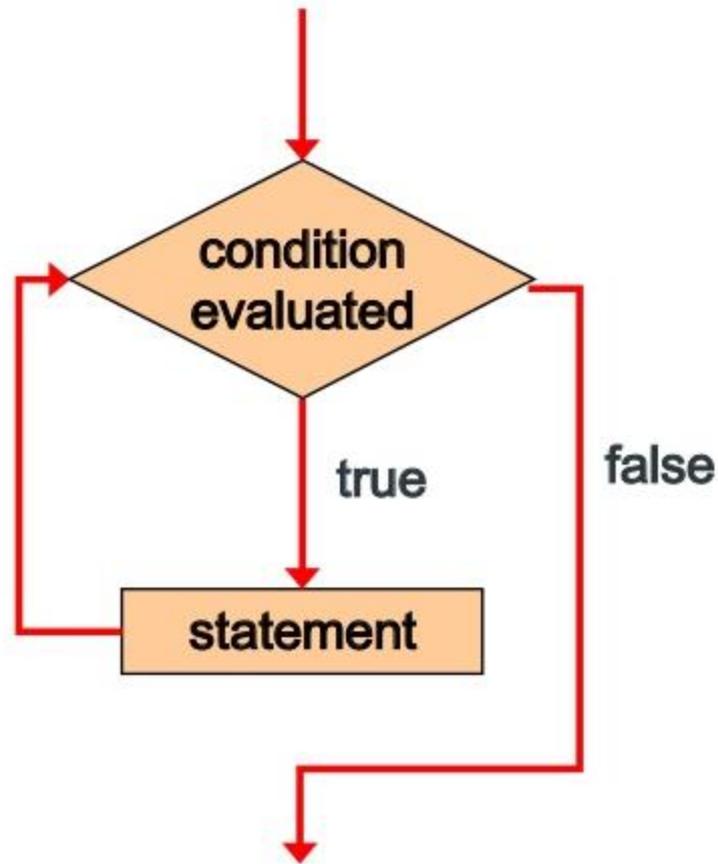
Code

Counter



```
    i++;
}
```

Logic of a while Loop



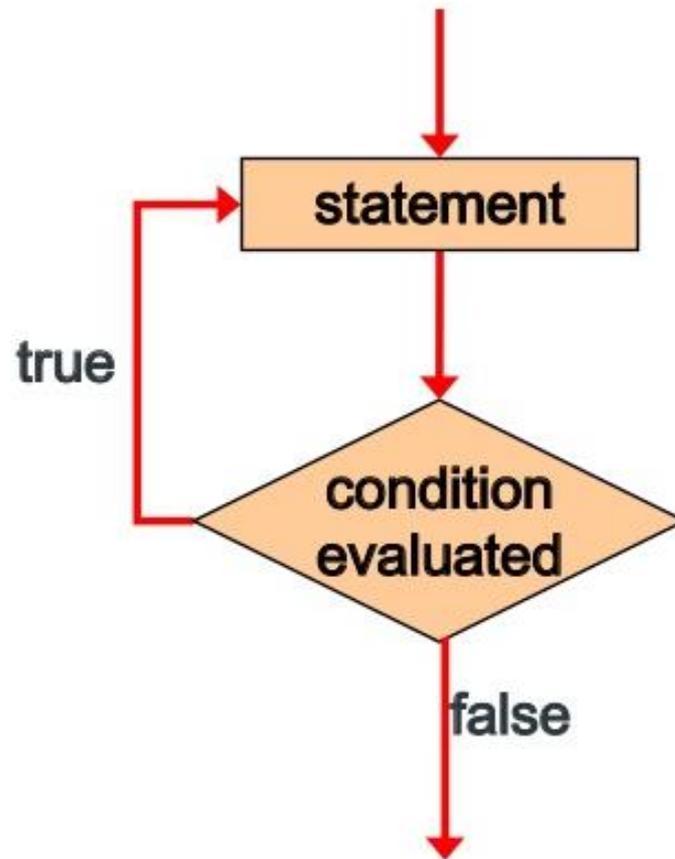
Ex) Write a program to find the summation of group of numbers where the last number is 7

```
# include<iostream.h>
void main()
{int x,m;
 cin>>x; m=x;
 while(x!=7)
 { cin>>x;
  m+=x;
 }
 cout<<m;
 }
```

DO...WHILE LOOP

- The syntax of **do-while** statement in C:
do
statement
while (**loop repetition condition**);
- The *statement* is first executed.
- If the **loop repetition condition** is true, the *statement* is repeated.
- Otherwise, the loop is exited.

Logic of a do...while loop



Ex) Write a program to find how many number can be divided by 3 between 15 numbers

```
# include<iostream.h>
void main()
{int x,i=0,j=0;
do
i++;
  cin>>x;
  if (x%3==0)j++;
while(i<15)
  cout<<j;
}
```

Another solution to find the largest negative number between ten numbers

```
# include<iostream.h>
void main()
{int x,i=0,m;
do
    i++; cin>>x;
    if (x<0)m=x;
while((x>0)&&(i<10))

while(i<10)
{i++;
cin>>x;
if(x<0) if (x>m)m=x;
}
cout<<m;
}
```



Thank you

Arrays

Bubble sort

How to sort one dimension array ?

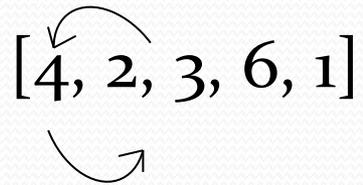
One of the most important algorithms to sort one dimension arrays is (bubble sort) which can be illustrated as in bellow :

Suppose that we have the following element:

[4, 2, 3, 6, 1]

And we want to sort it in ascending order to be

[1, 2, 3, 4, 6]


[4, 2, 3, 6, 1]

[2, 4, 3, 6, 1] [2, 3, 4, 1] [2, 3, 1] [1, 2]

[2, 3, 4, 6, 1] [2, 3, 4, 1] [2, 1, 3]

[2, 3, 4, 6, 1] [2, 3, 1, 4]

[2, 3, 4, 1, 6]

[1, 2, 3, 4, 6]

Bubble sort

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int a[6],i,j,x,n;
Cout<<"how many element:";
Cin>>n;
Cout<<"now give your elements: ";
for(i=0;i<n;i++)
Cin>>a[i];
for(i=0;i<n-1;i++)
for(j=0;j<n-i-1;j++)
if(a[j]>a[j+1])
{x=a[j]; a[j]=a[j+1];a[j+1]=x;
}
for(i=0;i<n;i++)
Cout<<a[i]<<"\n";
getch();
}
```

Two dimension Array examples (2)

Ex1) Write a program to create the following two dimension array

1	1	1
2	2	2
3	3	3

```
# include <iostream.h>
Void main()
{ int a[3][3] ,i,j;

  for (i=0;i<3;i++)
    for (j=0;i<3;i++)
      a[i][j]=i+1;
for (i=0;i<3;i++)
{
  for (j=0;j<3;j++)
    cout<<a[i][j]<<" ";
  Cout<<"\n";
}
}
```

Ex2) Write a program to create the following two dimension array

1	2	2
3	1	2
3	3	1

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

```
Void main()
```

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

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

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

```
if (i==j) a[i][j]=1; else if (i<j) a[i][j]=2;else a[i][j]=3;
```

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

```
{
```

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

```
cout<<a[i][j]<<"  ";
```

```
cout<<"\n";
```

```
}
```

```
}
```

Ex3) Write a program to create the following two dimension array

1	2	3
4	5	6
7	8	9

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

```
Void main()
```

```
{ int a[3][3] ,i,j,k=0;
```

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

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

```
        {k++;
```

```
        a[i][j]=k;
```

```
        }
```

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

```
{
```

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

```
        cout<<a[i][j]<<"  ";
```

```
    Cout<<"\n";
```

```
}
```

```
}
```

Ex4) Write a program to create the following two dimension array

1	4	6
8	1	12
14	16	1

```
# include <iostream.h>
Void main()
{ int a[3][3] ,i,j,k=0;

for (i=0;i<3;i++)
    for (j=0;i<3;i++)
        {k+=2;
         if (i==j) a[i][j]=1; else a[i][j]=k;
        }
for (i=0;i<3;i++)
{
    for (j=0;j<3;j++)
        cout<<a[i][j]<<" ";
    Cout<<"\n";
}
}
```

Two Dimensional Arrays

Contents

- Two-Dimensional Array
- Declaring Two-Dimensional Array
- Initializing Two-Dimensional Array
- Accessing elements of Two-Dimensional Array
- Inputting elements of Two-Dimensional Array
- Outputting elements of Two-Dimensional Array
- Operations on two Dimensional arrays

What is a Two Dimensional Array ?

- A two dimensional array is the arrangement of elements in rows and columns.
- It has two indices, one for row and other for column. First index represents the number of rows, second index represents the number of columns.

Example : `int list[4][4];`

ROWS

78	90	56	45
89	88	76	99
22	69	97	24
34	60	44	56

COLUMNS

The diagram shows a 4x4 grid of numbers. A horizontal bracket above the grid is labeled 'ROWS', indicating the 4 rows. A vertical bracket to the right of the grid is labeled 'COLUMNS', indicating the 4 columns. The numbers in the grid are: Row 1: 78, 90, 56, 45; Row 2: 89, 88, 76, 99; Row 3: 22, 69, 97, 24; Row 4: 34, 60, 44, 56.

Declaring a Two Dimensional Array

data-type name [no of rows] [no of columns];

integer constant / literal

Example :

```
int List[4][4];
```

- Data type= integer
- Name= list
- No. of rows=4
- No. of columns=4



	0	1	2	3
0	List[0][0]	List[0][1]	List[0][2]	List[0][3]
1	List[1][0]	List[1][1]	List[1][2]	List[1][3]
2	List[2][0]	List[2][1]	List[2][2]	List[2][3]
3	List[3][0]	List[3][1]	List[3][2]	List[3][3]

Initializing a Two Dimensional Array

- A two dimensional array can be initialized using curly braces:

```
int List[4][2]={{20, 25},{22, 24}, {18, 22}, {17, 24}};
```

row 1 row 2 row 3 row 4

We can also initialize in this way:

```
int List[4][2]={20, 25, 22, 24, 18, 22, 17, 24};
```

List[0][0]	20	25	List[0][1]
List[1][0]	22	24	List[1][1]
List[2][0]	18	22	List[2][1]
List[3][0]	17	24	List[3][1]

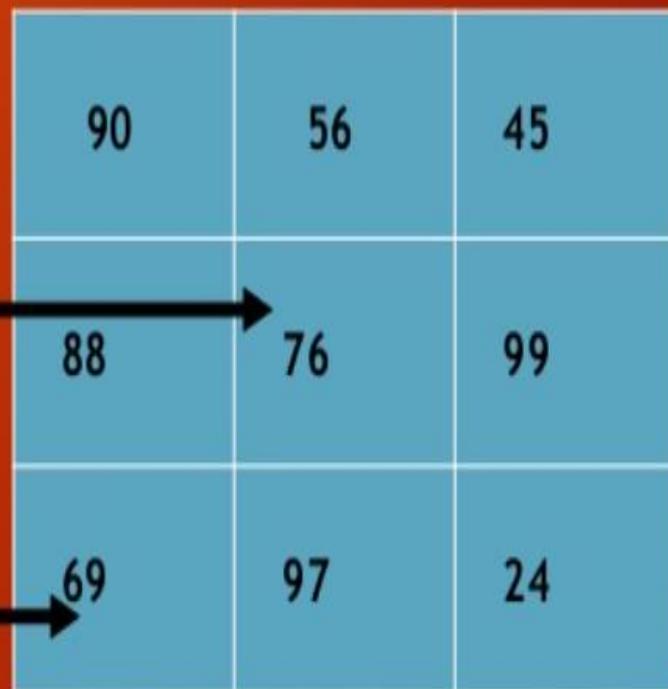
Memory Representation
Activate Windows

Accessing two dimensional array elements

```
int a[3][3]={90,56,88,76,99,69,97,24};
```

```
cout<<[1][1]; // displays 76
```

```
cout<<[2][0]; // displays 69
```



90	56	45
88	76	99
69	97	24

Inputting values into a two Dimensional Array

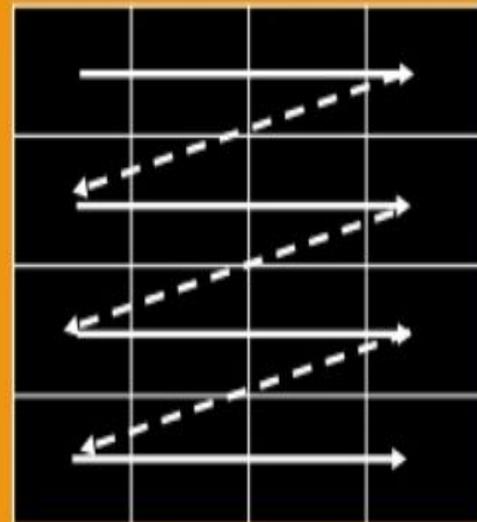
To input values into a two dimensional array we use a double loop.

Example : To input elements into the array List the form would be:

```
for(int i=0; i<4; i++)    outer loop for row index
  for(int j=0; j<2; j++)  inner loop for column index
    cin>>list[i][j];
```

The input order will be row wise.

Row wise input



Ex) Write a program to read and print two dimension array

Traversal means visiting each element of the array once. An example program would be to input and display the elements.

```
# include<iostream.h>
void main()
{ int A[20][20]; // array declaration
int M, N; // for no of rows and columns to be input
cout<<"Enter no of rows and columns";
cin>>M>>N;
for (int i = 0; i < M; i++)
for (int j = 0; j < N; j++)
{
cout<<"\nEnter element at row & column " <<i<<" ,
"<<j<<":";
cin >> A[i][j];
}
```

```
cout<<"\n\nThe elements of the array are :";
cout<<"\n *****\n";
for (int i = 0; i < M; i++)
{ for (int j = 0; j < N; j++)
cout<< A[i][j]<<"\t";
cout<<"\n";
}
}
```

Ex) Write a program to find the summation of each row in two dimension array

2. Sum of row elements : Program

```
#include<iostream.h>
void main()
{ int A[20][20], M,N, sum;
int M, N;
cout<<"Enter no of rows and columns";
cin>>M>>N;
for (int i = 0; i < M; i++)
for (int j = 0; j < N; j++)
{
cout<<"\nEnter element at row & column
"<<i<<" , "<<j<<": ";
cin >> A[i][j];
}
```

```
cout<<"\n";
for(i=0;i<M;i++)
{
sum=0;
for(int j=0;j<N;j++)
sum+=A[i][j];
cout <<"\nSum of Row " << i + 1 << "
=" << sum;
}
```

Strings in c++



String as a value:

“abcd”

“a12bc”

“1234760”

“#%^&ERT!@#1234”

How to define a string:

```
Char s1[10];
```

```
Char s2[]="abcd";
```

```
Char * S;
```

All of these definitions are valid

Dealing a string as an array of characters:

```
char s[10];  
  
for(i=0;i<10;i++)  
    cin>>s[i];  
  
for(i=0;i<10;i++)  
    cout<<s[i];
```

Here we will read the string and print it as
(character by character)

Dealing a string as one piece:

```
Char s [8];
```

```
cin>>s;
```

```
cout<<s;
```

Here we can read a string directly by one statement and so about printing it (without using index)

Using the above two types of dealing together for one string:

```
char s[10];
```

```
cin>>s;
```

```
for(i=0;i<10;i++)  
    cout<<s[i];
```

Here we will read a string in one statement then print it as character by character

More specialized standard functions for string processing

`gets(string name)` // to read a string

`puts(string name)` // to print a string

`strlen(string name)` // to find a string length

Here we need to include libraries like

```
#include<stdio.h>
```

```
#include<string.h>
```

Ex) Write a program to read a string then print it's characters in a separate lines

```
# include<stdio.h>
#include<string.h>
#include<iostream.h>
#include<conio.h>
void main()
{ clrscr();
char s[10];
  int i;
  gets(s);
  for(i=0;i<strlen(s);i++)
    cout<<s[i]<<'\n';
  getch();
}
```

Ex) Write a program to read a string then print it's words in a separate lines

```
# include<stdio.h>
#include<string.h>
#include<iostream.h>
#include<conio.h>
void main()
{ clrscr();
char s[10];
  int i;
  gets(s);
  for(i=0;i<strlen(s);i++)
    if (s[i]==‘ ‘) cout<<‘\n’;
    else cout<<s[i];
  getch();
}
```

More specialized standard functions for string processing

`strcpy(s1,s2)` // to copy s2 into s1

`strcat(s1,s2)` // return the concatenation of s1 and s2 strings

`strcmp(s1,s2)` // to compare string and return a value as follows :

‘value=0’ means strings are equal

‘value<0’ means $s1 < s2$

‘value>0’ means $s1 > s2$

More specialized standard functions for string processing

`strchr(s,ch)` // Returns a pointer to the first occurrence of character `ch` in string `s`

`strstr(s1,s2)` // Returns a pointer to the first occurrence of string `s2` in string `s1`.

Structure in C++

What is structure:

C/C++ arrays allow you to define variables that combine several data items of the same kind but **structure** is another user defined data type which allows you to combine data items of different kinds.

Structures are used to represent a record, suppose you want to keep track of your books in a library. You might want to track the following attributes about each book:

- Title
- Author
- Subject
- Book ID

Defining a Structure:

To define a structure, you must use the struct statement. The struct statement defines a new data type, with more than one member, for your program. The format of the struct statement is this:

```
struct [structure tag]
{
    member definition;
    member definition;
    ...
    member definition;
} [one or more structure variables];
```

The **structure tag** is optional and each member definition is a normal variable definition, such as `int i;` or `float f;` or any other valid variable definition. At the end of the structure's definition, before the final semicolon, you can specify one or more structure variables but it is optional.

How to declare the Book structure:

```
struct Books
{
    char    title[50];
    char    author[50];
    char    subject[100];
    int     book_id;
}book;
```

Another way to declare structures:

```
struct rec
{ char name[20];
  int dgree1,degree2;
};
void main()
{ rec student1, student2;
  ....
  ....
```

How to access the structure field:

We have to write the structure name then dot then the field name as follows

```
struct rec
{ char name[20];
  int dgree1,degree2;
};
void main()
{ rec student1, student2;
student1.degree1=80;
cin>>student1.name;
.....
.....
```

To make structure more useful:

We can define an array where each element of it is a structure therefore we can save an information like a table of data base , suppose that we have the following table and we want to save it's information:

St_name	Lesson 1	Lesson 2	Average
Ahmed	79	81	80
Ali	60	80	70

Another way to declare structures:

```
struct rec
{ char name[20];
  int dgree1,degree2;
};
void main()
{ rec student1, student2;
  ....
  ....
```

Ex)Write a program to save an information about student like in the table below and find the average for each student

Name	Lesson1	Lesson2	Average

```

≡ File Edit Search Run Compile Debug Project Options Window Help
[ ] REC.CPP 1=[ ]
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
struct student
{
char name[20];
int lesson1,lesson2;
float average;
};
void main()
{clrscr();
student a[10];
int n,i;
cout<<"How many student:";
cin>>n;
for (i=0;i<n;i++)
{
cout<<"give student name:";
gets (a[i].name);
cout<<"give lesson1:";

```

2:54

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```

≡ File Edit Search Run Compile Debug Project Options Window Help
[ ] REC.CPP 1=[ ]
cin>>a[i].lesson1;
cout<<"give lesson2:";
cin>>a[i].lesson2;
a[i].average=(a[i].lesson1+a[i].lesson2)/2;
}

cout<<' \n' ;
int j,k;
cout<<"Name      les1   les2   av " <<' \n' ;
for (i=0;i<n;i++)
{ cout<<a[i].name;
  j=strlen(a[i].name);
  for(k=1;k<15-j;k++)cout<<' ' ;
  cout<<a[i].lesson1<<" " <<a[i].lesson2<<" " <<a[i].average<<' \n' ;
}

getch();
}

2:54
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```

Ex) Write a program to save an information about inventory system like in the table below and find the number of finished goods

Gono	Goname	Qty	price

```

File Edit Search Run Compile Debug Project Options Window Help
REC_INJVE.CPP
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
struct rec
{
int gono;
char gname[40];
int qty;
float price;
};
void main()
{clrscr();
rec a[100];
int n,i,q=0;
cout<<"How many goods you have:";
cin>>n;
for (i=0;i<n;i++)
{
cout<<"give goods no:";
cin>>a[i].gono;
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```

```

File Edit Search Run Compile Debug Project Options Window Help
REC_INJVE.CPP
cout<<"give Gname :";
gets (a[i].gname);
cout<<"give Qty:";
cin>>a[i].qty;
cout<<"give price:";
cin>>a[i].price;
}
cout<<"\n";
cout<<"Gono Gname Qty price " <<"\n";
for (i=0;i<n;i++)
{
cout<<a[i].gono<<" " <<a[i].gname<<" " <<a[i].qty<<" " <<a[i].price<<"\n";
if (a[i].qty==0) q++;
}
cout<<"\n";
cout<<"number of finished goods is : " <<q;
getch();
}
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

```

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

void print(char c, int& t)
{
    for(int i=1;i<=t;i++)
        cout<<i<<"-   "<<c<<'\\n';
    t=t+i;
    cout<<"the value of t= "<<t<<'\\n';
}

void main()
{
    clrscr();
    int x;
    char ch;

    cout<<"Give the character: " ;
    cin>>ch;
    cout<<"Give the value of x: " ;
    cin>>x;
    cout<<"the value of x= "<<x<<'\\n';
    print(ch,x);
    cout<<"the value of x= "<<x<<'\\n';
    getch();
}
```

```

#include<iostream.h>
#include<conio.h>

void swap(int& x, int& y)
{
    int z;
    z=x;
    x=y;
    y=z;
}

void main()
{
    clrscr();
    int x,n,h,m;

    cout<<"Give the value of x: " ;
    cin>>x;
    cout<<"Give the value of n: " ;
    cin>>n;
    swap(x,n);
    cout<<"After swaping x= "<<x<<" and n= "<<n<<'\n'
;
    cout<<"Give the value of h: " ;
    cin>>h;
    cout<<"Give the value of m: " ;
    cin>>m;
    swap(h,m);
    cout<<"After swaping h= "<<h<<" and m= "<<m<<'\n'
;

    getch();
}

```

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

int fact(int x)
{
    int f;
    if (x==0) return 1;
    else f=x* fact(x-1);
    return (f);
}

void main()
{ clrscr();
  int x,n,m;
  float z;

  cout<<"Give the value of x: " ;
  cin>>x;
  cout<<"Give the value of n: " ;
  cin>>n;
  cout<<"Give the value of m: " ;
  cin>>m;

  z=(fact(x)+fact(n))/(fact(m)*1.0);
  cout<<"The value of Z= "<<z ;

  getch();
}
```

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

int power(int x, int n)
{
    int p;
    if (n==0) return 1;
    else
        p=x*power(x,n-1);
}
```

```
    return (p);
}

void main()
{ clrscr();
  int x,n,h,m,z,r;
  float y;

  cout<<"Give the value of x: " ;
  cin>>x;
  cout<<"Give the value of n: " ;
  cin>>n;
  cout<<"Give the value of h: " ;
  cin>>h;
  cout<<"Give the value of m: " ;
  cin>>m;
  cout<<"Give the value of z: " ;
  cin>>z;
  cout<<"Give the value of r: " ;
  cin>>r;

  y=(power(x,n)+power(h,m))/(power(z,r)*1.0);
  cout<<"The value of Y= " <<y ;

  getch();
}
```