

# UNIT 3

## Control Structures

Control structure enables us to specify the order in which the various instructions in a program are to be executed by the computer. In other words, control statement determine the “flow of control” in a program. There are 3 types of control statements in C.

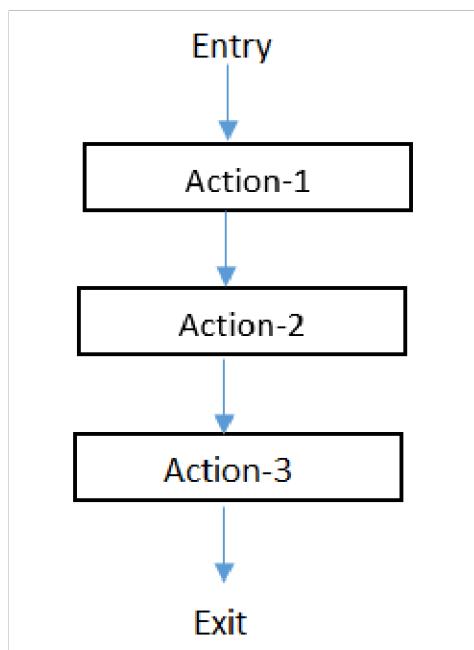
They are:

- 1 Sequential Control statement
- 2 Selection or decision or branching control statement
- 3 Repetition or looping control statement

### 1. Sequential Control Statement

- Statements are executed in the same order in which they appear.
- Each statement is executed exactly once.
- No condition is evaluated.

*Flowchart*



### **Example**

#### **Program to find the sum of two numbers**

```
#include<stdio.h>
int main()
{
int num1,num2,sum;
printf("Enter any two numbers\n");
scanf("%d%d",&num1,&num2);
sum=num1+num2;
printf("Sum of two numbers is %d",sum);
return 0
}
```

## **2. Selection/Branching/Decision Control statements**

They are used when we have a number of situations where we may need to change the order of execution of statements based on certain conditions.

- i. simple if statement
- ii. if else statement
- iii. nested if else statement
- iv. else if ladder

### **i) Simple if statement**

Evaluates the expression first then:

- If the value of expression is true , it executes the statement within the block
- Otherwise it skips the statements within its block and continues from the first statement outside the block.

Syntax	Flowchart
<pre>if(test expression) { statement-block; } statement-x;</pre>	<pre> graph TD     Start(( )) --&gt; Decision{test expression ?}     Decision -- True --&gt; StatementBlock[statement-block]     StatementBlock --&gt; NextStatement[Next statement]     Decision -- False --&gt; StatementX[statement-x]     StatementX --&gt; NextStatement     </pre>

**Program to input the average marks of a student and add 10% bonus marks if his/her average marks is greater than or equal to 65.**

```
#include<stdio.h>
int main()
{
float marks;
printf("Enter the average marks of student\n");
scanf("%f",&marks);
if(marks>=65)
{
    marks=marks+marks*0.1;
}
printf("Final marks=%f",marks);
return 0;
}
```

## ii) If-else statement

- Extension of simple if statement.
- If test expression is true,
  - then true block statement(s) (*immediately following the if statements are executed*)
  - otherwise false block statements are executed.

Syntax	Flowchart
<pre>if(test_expression) {     true-block statements } else {     false-block statements } statement-x;</pre>	<pre> graph TD     Entry((Entry)) --&gt; Decision{test expression ?}     Decision -- True --&gt; TrueBlock[True-block statement]     Decision -- False --&gt; FalseBlock[False-block statement]     TrueBlock --&gt; Merge(( ))     FalseBlock --&gt; Merge     Merge --&gt; StatementX[statement-x]     StatementX --&gt; Exit((Exit))     </pre>

**Write a program to find the maximum number between two numbers**

```
#include<stdio.h>
int main()
{
int num1,num2;
printf("Enter the first number\n");
scanf("%d",&num1);
printf("Enter the second number\n");
scanf("%d",&num2);
if(num1>num2)
{
printf("Maximum number=%d",num1);
}
else
{
printf("Maximum number=%d",num2);
}
return 0;
}
```

### iii) Nested if-else statement

When a series of decisions are involved, we may have to use more than one if. . . else statement in nested form as shown below.

Syntax	Flowchart
<pre>If(test condition-1) {     If(test condition-2)     {         Statement -1;     }     else     {         Statement-2;     } } else {     Statement-3; } Statement x;</pre>	<pre>graph TD     Entry((Entry)) --&gt; D1{test condition-1 ?}     D1 -- False --&gt; S3[Statement-3]     D1 -- True --&gt; D2{test condition-2 ?}     D2 -- False --&gt; S2[Statement-2]     D2 -- True --&gt; S1[Statement-1]     S2 --&gt; Join(( ))     S1 --&gt; Join     Join --&gt; Sx[Statement-x]     Sx --&gt; NS[Next Statement]     NS --&gt; Exit(( ))</pre>

**Program to find the largest number among three numbers.**

```
#include<stdio.h>

int main()
{
    int a,b,c;
    printf("Enter three numbers\n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        if(a>c)
        {
            printf("The largest number is %d",a);
        }
        else
        {
            printf("The largest number is %d",c);
        }
    }
}
```

```

else
{
    if(b>c)
    {
        printf("largest number is %d",b);
    }
    else
    {
        printf("largest number is %d",c);
    }
}
return 0;
}

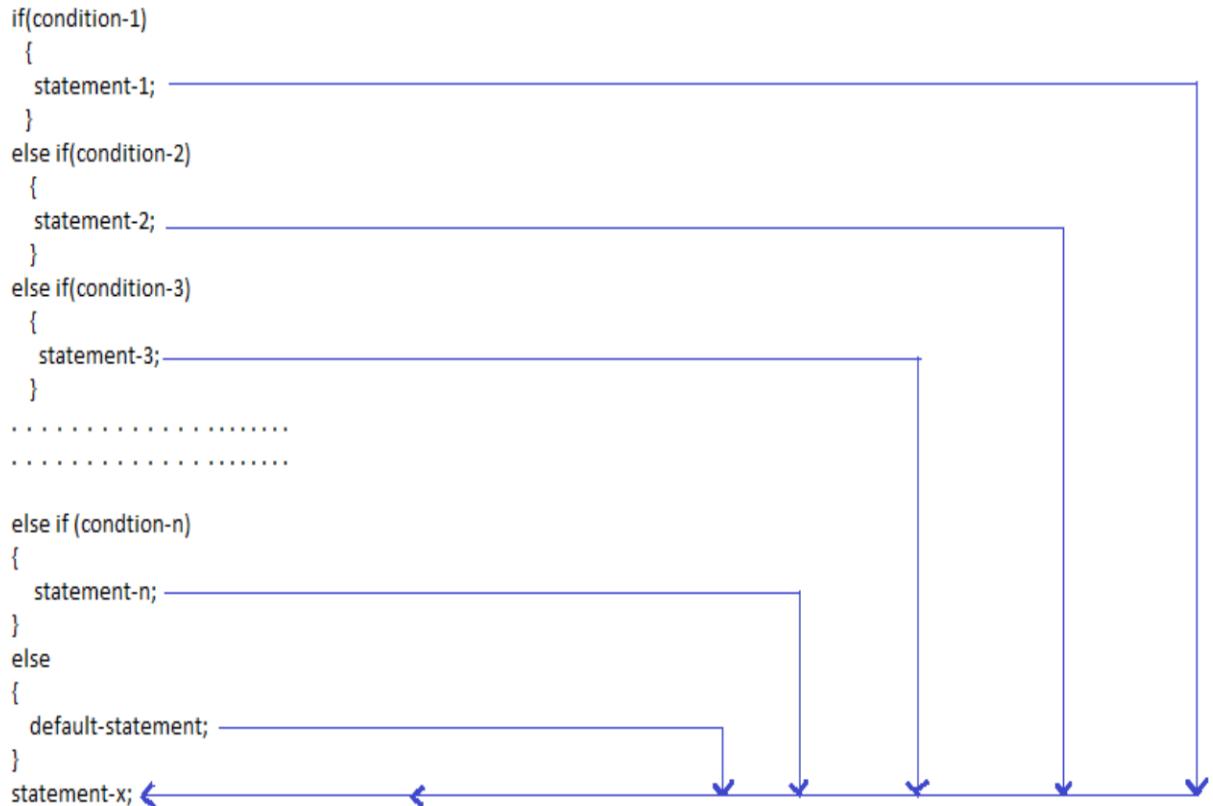
```

#### iv) Else-if ladder

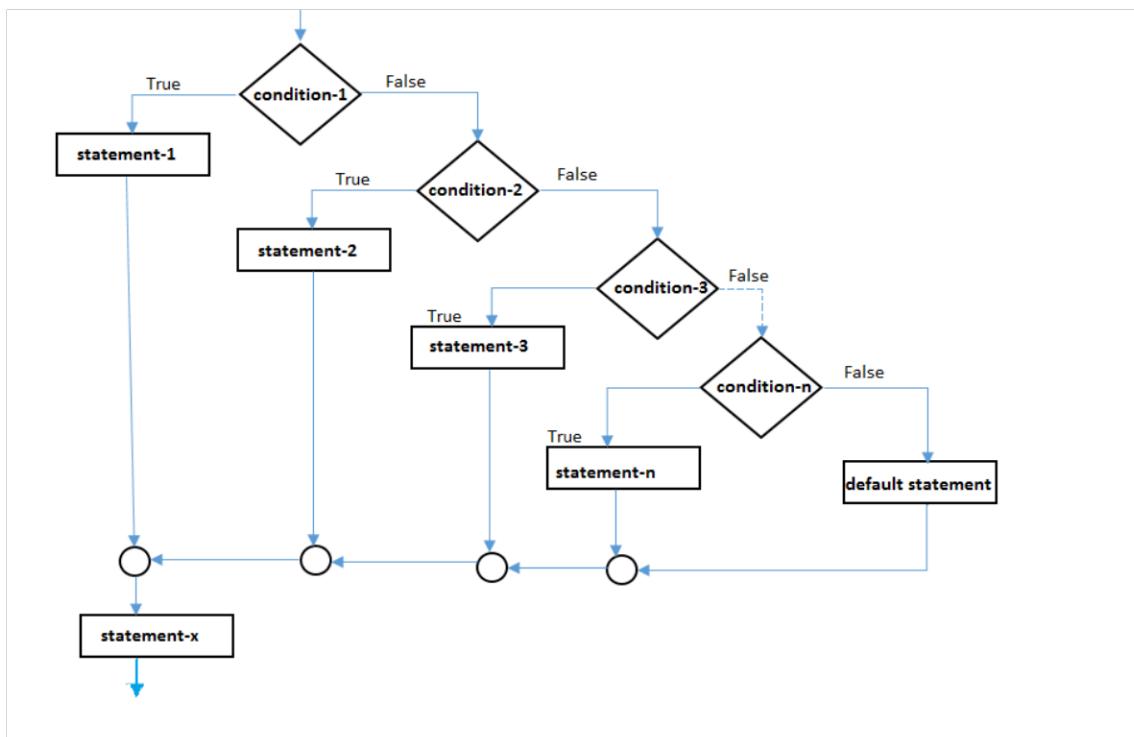
It is used when multiple decisions are involved. Here, the condition expression is evaluated in order.

- If any of these expression is true, the statement associated with it is executed and this terminates the whole chain.
- If none of the expression is true then statement associated with final else is executed.

#### Syntax:



## Flowchart



An electricity board charges according to the following rates.

For the first 100 units ..... Rs 40 Per Unit

For the next 200 units.....Rs. 50 Per Unit

For the beyond 300 Units.....Rs.60 Per unit

All users are also charge meter charge. Which is equal to Rs.50. Write a program to read number of units consumed and print out total charges.

```
#include<stdio.h>
#define METER_CHARGE 50
int main()
{
    int units,charge,totalcharge;
    printf("Enter the number of units\n");
    scanf("%d",&units);
    if(units<=100)
    {
        charge=units*40;
    }
    else if(units<=300)
    {
        charge=100*40+(units-100)*50;
    }
    else
    {
        charge=100*40+200*50+(units-300)*60;
    }
}
```

```

totalcharge=charge+METER_CHARGE;
printf("Total charge=%d",totalcharge);
return 0;
}

```

**Write a program to read the marks of 5 subjects and based on percentage printout the following.(Assume full marks of each subject is 100)**

Percentage	Result
<b>More than or equals to 80</b>	<b>Distinction</b>
<b>Less than 80 or more than or equals to 60</b>	<b>First division</b>
<b>Less than 60 or more than or equals to 45</b>	<b>Second division</b>
<b>Less than 45 or more than or equals to 32</b>	<b>Third division</b>
<b>Less than 32</b>	<b>Failed</b>

```

#include<stdio.h>
int main()
{
float sub1,sub2,sub3,sub4,sub5,per;
printf("Enter the marks of 5 subjects\n");
scanf("%f%f%f%f%f",&sub1,&sub2,&sub3,&sub4,&sub5);
per=(sub1+sub2+sub3+sub4+sub5)/5;
if(per>=80)
{
printf("Distinction");
}

else if(per>=60)
{
printf("First division");
}

else if(per>=45)
{
printf("Second division");
}

else if(per>=32)
{
printf("Third division");
}

else
{
printf("Failed");
}

return 0;
}

```

### 3. Repetition/Iteration/Loop Control statements

Loop control statement causes a program to execute the certain block of code until some conditions for termination of loop are satisfied.

Basically there are three types of loop control statements.

- i) while Loop (*Entry Controlled loop*)
- ii) do while loop (*Exit Controlled loop*)
- iii) for loop

#### i) While loop

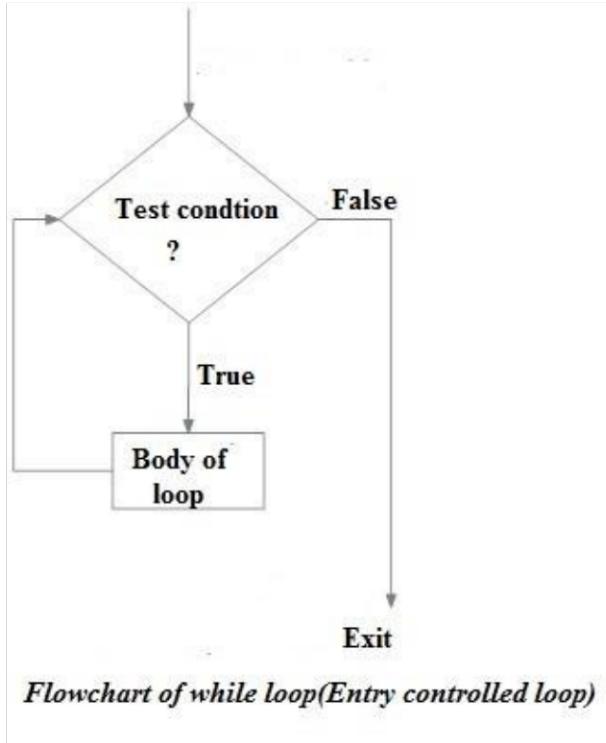
Test condition is evaluated first

- If the condition is true the body of the loop is executed.  
After the execution of the body, test condition is again evaluated and if it is true body is executed once again. This process goes on until test condition is false.
- If test condition is false, the body of the loop will not be executed and control is transferred out of the loop.

#### Syntax:

```
while (test condition)
{
    body of the loop
}
```

#### Flowchart



### Program to print 10 natural numbers using while loop.

```
#include<stdio.h>
int main()
{
int i;
i=1;
while(i<=10)
{
    printf("%d\n",i);
    i++;
}
return 0;
}
```

#### ii) do while loop /exit controlled loop

The test is performed at end of the body of the loop and therefore the body of the loop is executed unconditionally for the first time.

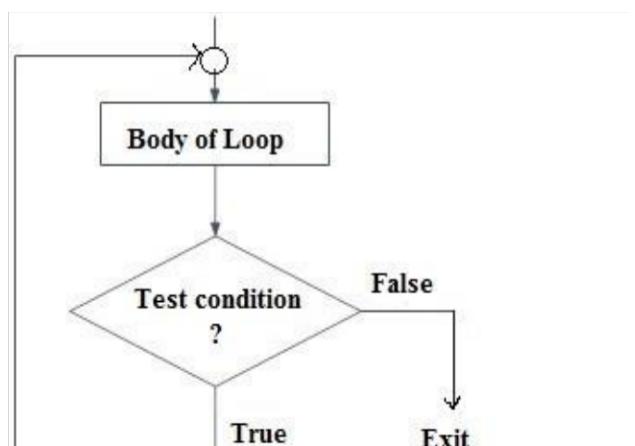
Then, test condition is evaluated.

- If test condition is true, the body of the loop is executed again. This process goes on until the test condition is false.
- When test condition is false, loop is terminated and control goes to the statement that appears immediately after the while statement.

#### Syntax

```
do
{
    Body of the loop
} while(test condition);
```

#### Flowchart



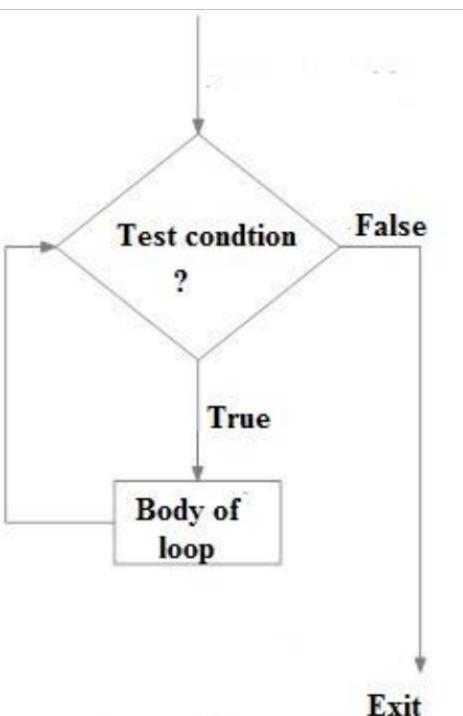
Flowchart of do-while(Exit controlled) loop

### **Program to print 10 natural numbers using do-while loop.**

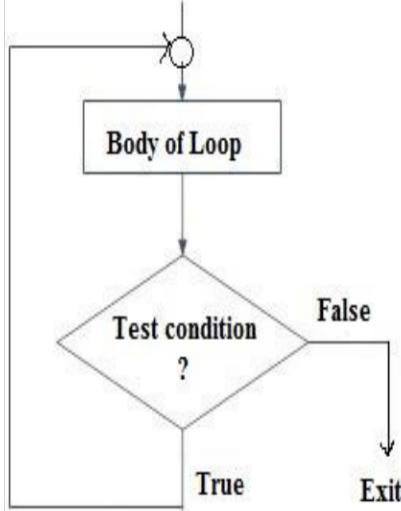
```
#include<stdio.h>
int main()
{
int i;
i=1;
do
{
    printf("%d\n",i);
    i++;
}while(i<=10);
return 0;
}
```

**Differentiate Entry controlled loop (While loop) and Exit controlled loop (Do- While loop) with examples.**

<b>While loop(Entry controlled loop)</b>	<b>Do-while loop(Exit controlled loop)</b>
Test condition is evaluated at beginning of the loop execution.	Test condition is evaluated at the end of the body of the loop.
The body of the loop will execute only if the test condition is true.	The body of the loop will execute at least once without depending on the test condition.
<b>Syntax:</b> <pre>while(test condition) { Body of the loop }</pre>	<b>Syntax:</b> <pre>do { Body of the loop }while(test condition);</pre>



*Flowchart of while loop(Entry controlled loop)*



*Flowchart of do-while(Exit controlled) loop*

#### **Program to illustrate concept of while loop**

```
#include<stdio.h>
int main()
{
int i=0;
while(i>=4)
{
printf("Hello\n");
i++;
}
return 0;
}
```

**Output:** Nothing will display.

#### **Program to illustrate concept of do- while loop**

```
#include<stdio.h>
int main()
{
int i=0;
do
{
printf("Hello\n");
i++;
}while(i>=4);
return 0;
}
```

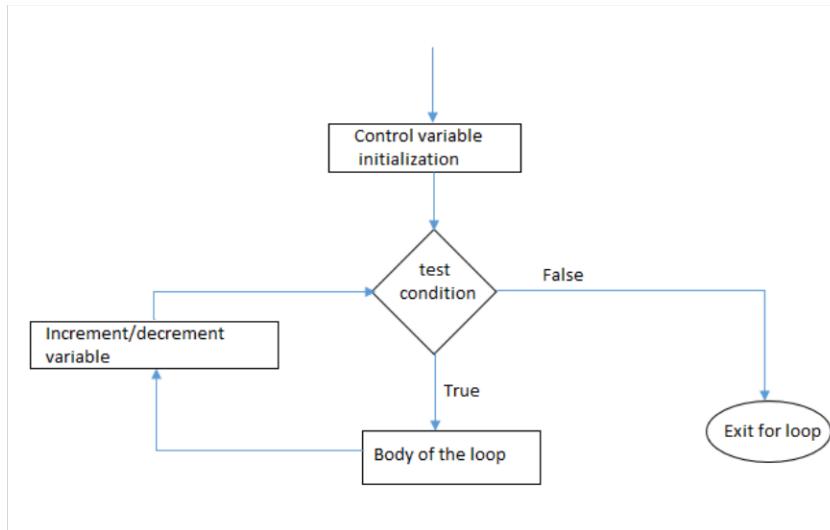
**Output:** Hello

### iii) For loop

Syntax:

```
for(initialization; test condition; increment/decrement)
{
    body of the loop
}
```

Flowchart:



Example:

Program to print 10 natural numbers using for loop.

```
#include<stdio.h>
int main()
{
    int i;
    for(i=1;i<=10;i++)
    {
        printf("%d\n",i);
    }
    return 0;
}
```

## For Loop vs while Loop

For Loop	While Loop
It have definite number of iterations.	It may or may not have definite number of iterations.
Keyword <b>for</b> is used.	Keyword <b>while</b> is used.
In 'for' loop the initialization once done is never repeated.	In while loop if initialization is done during condition checking, then initialization is done each time the loop iterate.
In 'for' loop iteration statement is written at top, hence, executes only after all statements in loop are executed.	In 'while' loop, the iteration statement can be written anywhere in the loop
<b>Syntax:</b>  for(initialization; test-condition;increment/decrement) { body of the loop }	<b>Syntax:</b>  while(test condition) { Body of loop }
<b>For example:</b>  #include<stdio.h>  int main() { int n; for(n=0;n<10;n++) { printf("Nepal\n"); } return 0; }	<b>For example:</b>  #include<stdio.h>  int main() { int n=10; while(n!=0) { printf("Nepal\n"); n--; } return 0; }

## Nested Loop

A loop inside another loop is called nesting of loops. There can be any number of loops inside one another with any combinations depending on the complexity of program.eg. A for loop inside a while loop or while loop inside a for loop.

Nested while loop	Nested do-while loop
<b>while(test-condition1)</b>	<b>do</b>
{	{
Statement(s);	Statement(s);
<b>while(test-condition2)</b>	<b>do</b>
{	{
Statement(s);	Statement(s);
.....	.....
}	} <b>while(test-condition2);</b>
.....	.....
}	} <b>while(test-condition1);</b>

Nested for loop
<b>for(initialization;testcondition;increment/decrement)</b>
{
Statement(s);
<b>for(initialization;testcondition;increment/decrement)</b>
{
Statement(s);
}
}

### Example: Nested for loop

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

## Jump statements

Jump Statement makes the control jump to another section of the program unconditionally when encountered. It is usually used to terminate the loop or switch-case instantly. It is also used to escape the execution of a section of the program. There are basically three types of jump statements.

- Break
- Continue
- Goto

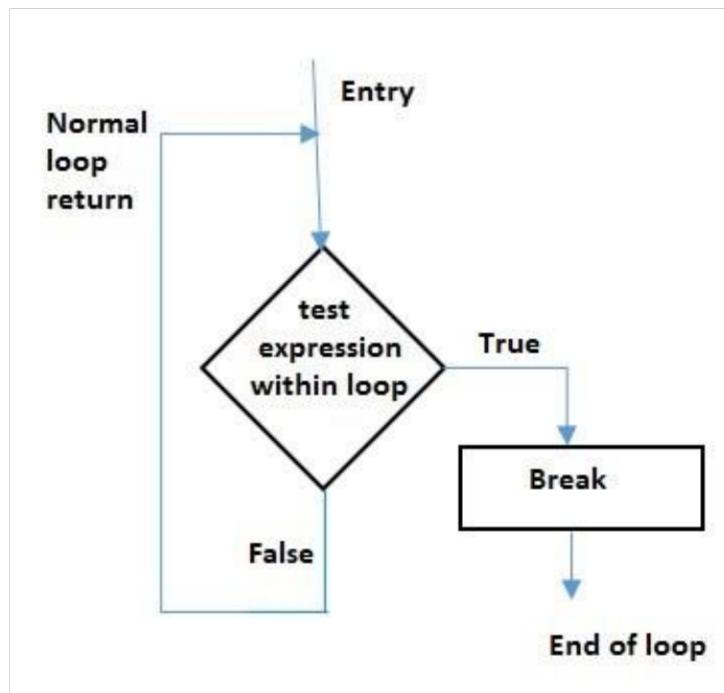
### i) Break statement

The break statement is used inside the loop or switch statement.

The execution of break statement will abort the loop and continue to execute statements followed by loop.

Similarly, execution of break statement causes bypass the rest of statement in switch and takes control out of the switch statement.

**Flowchart:**



### **Example**

```
#include<stdio.h>
int main()
{
int i;
for(i=1;i<=5;i++)
{
printf("%d\t",i);
if(i==3)
{
    break;
}
}
printf("\nLoop terminates here.");
```

return 0;

}

**Output:**

```
1    2    3
```

```
Loop terminates here.
```

**How break statement works?**

<pre>while(test condition) {     statement/s;     if(test condition)         break;     statement/s; }</pre>	<pre>do {     statement/s;     if(test condition)         break;     statement/s; } while(test condition);</pre>
<pre>for(initialization; test condition; increment/decrement) {     statement/s;     if(test condition)         break;     statement/s; }</pre>	

## ii) Continue statement

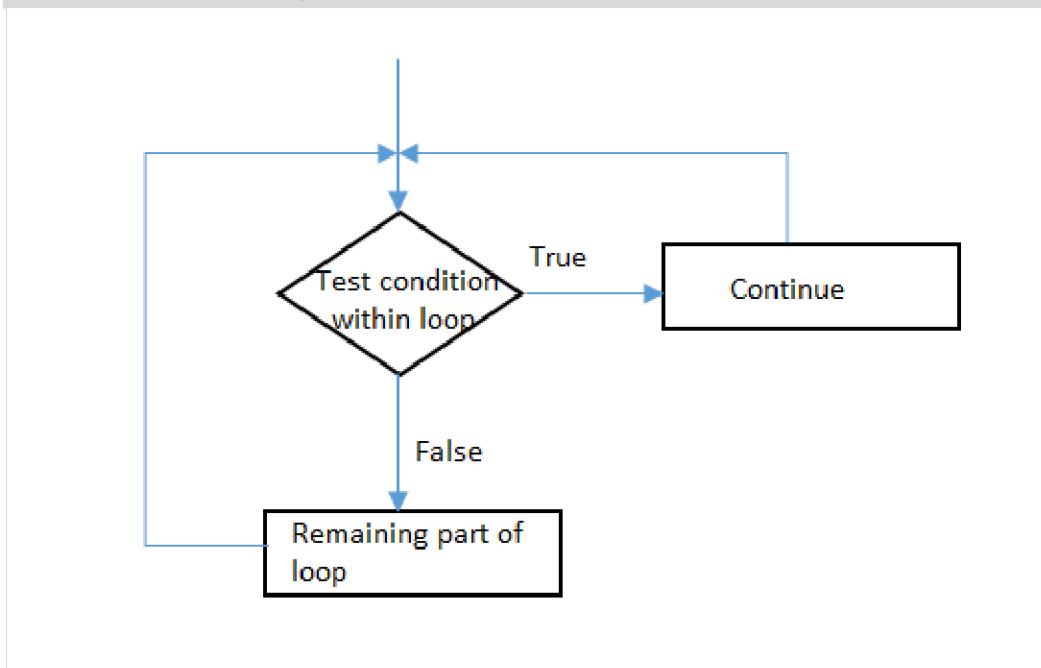
When continue statement is encountered inside the loop, it skips the execution of statements specified within the body of the loop and control automatically passes to the next iteration of the loop.

### Example

```
#include<stdio.h>
int main()
{
    int i;
    for(i=1;i<=5;i++)
    {
        if(i==3)
        {
            continue;
        }
        printf("%d\t",i);
    }
    return 0;
}
```

### Output:

```
1    2    4    5
```



## How continue statement works?

```
► while(test condition)
{
    statement/s;
    if(test condition)
        continue;
    statement/s;
}
► do
{
    statement/s;
    if(test condition)
        continue;
    statement/s;
} while(test condition);
```

```
► for(initialization; test condition; increment/decrement)
{
    statement/s;
    if(test condition)
        continue;
    statement/s;
}
```

### iii) goto statement

The goto statement is used to alter the program execution sequence by transferring the control to some other parts of the program.

Syntax:

```
goto label;
.....
.....
label:
Statement;
label:
Statement;
```

**Forward jump**                           **Backward jump**

```
label:
.....
.....
goto label;
Statement;
```

where, label is an identifier that is used to label the target statement to which control will be transferred.

**Note:** Use of goto statement is highly discouraged in any programming language because it makes difficult to trace the control flow of program, making the program hard to modify and understand.

### **Program to illustrate the working of goto statement**

```
#include<stdio.h>
int main()
{
    int a,b;
    printf("Enter two numbers\n");
    scanf("%d%d",&a,&b);
    if(a>b)
    {
        goto label1;
    }
    else
    {
        goto label2;
    }
label1:
    printf("Greatest number=%d",a);
    return;
label2:
    printf("Greatest number=%d",b);
    return 0;
}
```

### **switch-case statement**

The control statement that allows us to make a decision from number of choices is called switch case statement.

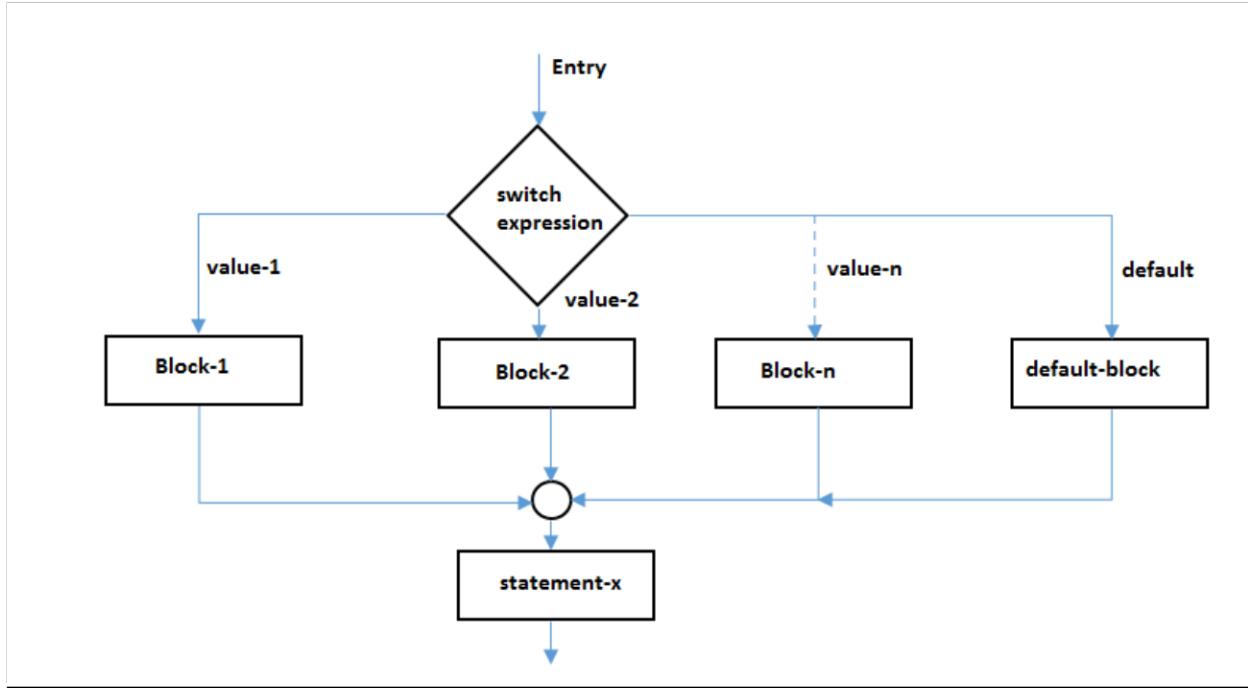
The switch case statement successively test the value of a given variables (an expression) with a list of case value (integer or character constants)

- When match is found, the statement associated with that case is executed.
- If none of the case value matches the expression then default statement is executed.

#### **General form:**

```
switch(expression)
{
    case value-1:
        block-1;
        break;
    case value-2:
        block-2;
        break;
    .....
    default:
        default-block;
        break;
}
statement-x;
```

### Flowchart:



for example, consider the program to display the corresponding days of a week according to the numbers entered.

```
#include<stdio.h>
int main()
{
int day;
printf("Enter the numeric day of week\n");
scanf("%d",&day);
switch (day)
{
case 1:
printf("Day is Sunday")      ;
break;
case 2:
printf("Day is Monday");
break;
case 3:
printf("Day is Tuesday");
break;
case 4:
printf("Day is Wednesday");
break;
case 5:
printf("Day is Thursday");
break;
case 6:
printf("Day is Friday");
break;
}
```

```

case 7:
printf("Day is Saturday");
break;
default:
printf("Invalid choice!");
}
return 0;
}

```

**WAP to display the following menu and perform the following operations.**

- 1. Find the simple interest**
- 2. Convert degree Celsius to Fahrenheit**
- 3. Convert character into ASCII code**
- 4. Exit from the program**

**and perform above operation until user want to exit.**

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
int choice;
float p,t,r,si,cel,fah;
char ch;
while(1)
{
printf("Menu\n");
printf("1.Find simple interest\n");
printf("2.convert celsius of fahrenheit\n");
printf("3.convert character to ASCII code\n");
printf("4.Exit from program\n");
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Enter Principal,Time and Rate\n");
scanf("%f%f%f",&p,&t,&r);
si=(p*t*r)/100;
printf("simple interest=%f\n",si);
break;
case 2:
printf("Enter the temperature in celsius\n");
scanf("%f",&cel);
fah=1.8*cel+32;
printf("Temperature in fahrenheit= %f\n",fah);
break;
}
}

```

```

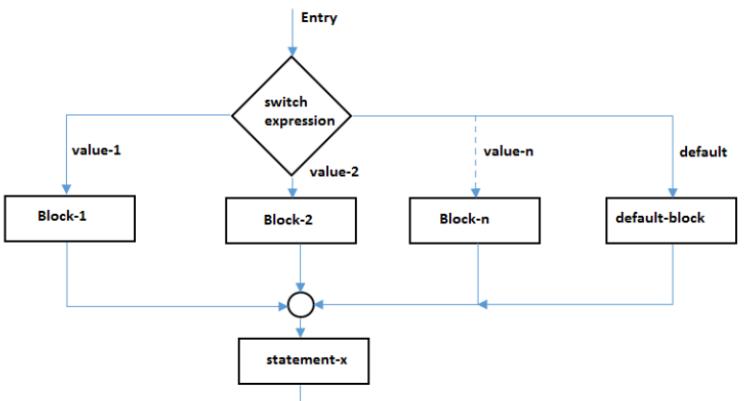
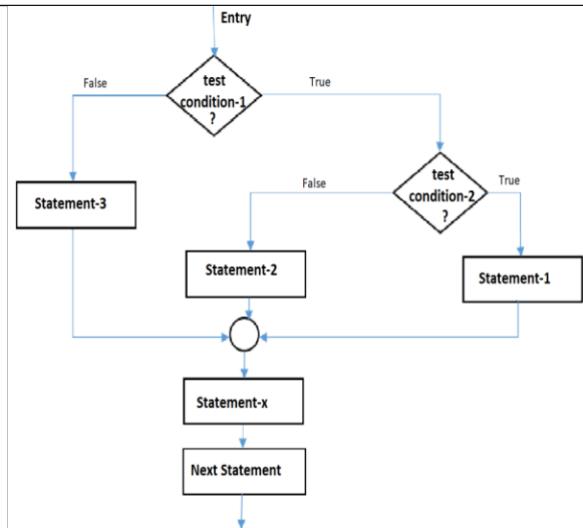
case 3:
printf("Enter a character\n");
scanf("%c",&ch);
printf("The corresponding ASCII code = %d\n",ch);
break;
case 4:
exit(0);
break;
default:
printf("Wrong choice!");
}
}
return 0;
}

```

## Difference between if-else and switch

If-else	Switch
1. Execution of statement will be depend upon the output of the expression inside if statement.	1. Which statement will be executed is decided by user.
2. if-else statement uses multiple statement for multiple choices.	2. switch statement uses single expression for multiple choices.
3. if statement evaluates integer, floating-point type as well boolean type.	3. switch statement evaluates only character or integer value.
4. Either if statement will be executed or else statement is executed.	4. switch statement execute one case after another till a break statement is appeared or the end of switch statement is reached.
5. It does not require break statement because only one of the blocks of code is executed.	5. It needs the involvement of break statement to avoid execution of block just below the current executing block.
6. If the condition inside if statements is false, then by default the else statement is executed if created.	6. If the condition inside switch statements does not match with any of cases, for that instance the default statements is executed if created.

**Differentiate between switch and nested if else statements with a suitable example.**

Switch	nested if-else
It is easy to understand for multiple selections.	It becomes complicated for multiple selections.
It uses a single expression for all cases, but each case must have a constant value of integer type or character type.	It uses an independent expression for each case.
Only a single expression is given in the switch statement which returns a single value. The test condition cannot be given in a specified range.	The test condition can be given in a special range of value. If the given condition matches then the statements under it will be executed.
<b>General form:</b>  <pre>switch(expression) {     case value-1:         block-1;         break;     Case value-2:         block-2;         break;         .....     default:         default-block;         break; } statement-x;</pre>	<b>General form:</b>  <pre>If(test condition-1) {     If(test condition-2)     {         Statement -1;     }     else     {         Statement-2;     } } else {     Statement-3; } Statement x;</pre>
	

It needs involvement of the break statement to avoid the execution of the block just below the current executing block.	It does not require break statement because only one of the blocks of code is execute at a time.
If the condition inside switch statements does not match with any of cases, for that instance the default statements is executed if created.	If the condition inside if statements is false, then by default the else statement is executed if created.

## **PROGRAMS RELATED TO DECISION CONTROL STATEMENTS**

- 1. WAP to find second largest number among three numbers (middle number)**

```
#include<stdio.h>
int main()
{
int a,b,c;
printf("Enter three numbers\n");
scanf("%d%d%d",&a,&b,&c);
if(a>b&&a<c | a>c&&a<b)
{
printf("second largest(middle) number=%d",a);
}
else if(b>a&&b<c | b>c&&b<a)
{
printf("second largest(middle) number=%d",b);
}
else
{
printf("second largest(middle) number=%d",c);
}
return 0;
}
```

- 2. WAP to find the largest number among four numbers.**

```
#include<stdio.h>
int main()
{
int a,b,c,d;
printf("Enter the four numbers\n");
scanf("%d%d%d%d",&a,&b,&c,&d);
if(a>b&&a>c&&a>d)
{
printf("max=%d",a);
}
else if(b>c&&b>d)
{
printf("max=%d",b);
}
else if(c>d)
{
printf("max=%d",c);
}
else
{
printf("max=%d",d);
}
return 0;
}
```

- 3. An electricity board charges according to following rates.**

**For the first 20 units..... Rs 80**

**For the next 80 units.....Rs.7.5 per unit**

**For the next 100 units ...Rs 8.5 per unit**

**For the beyond 200 units ....Rs 9.5 per unit**

**And Tax 15% in total amount is charged to all users.**

**Write a program to read number of units consumed and print out the total charges.**

```
#include<stdio.h>
int main()
{
int units;
float amount,total;
printf("Enter the number of units\n");
scanf("%d",&units);
```

```

if(units<=20)
{
amount=80;
}
else if(units<=100)
{
amount=80+(units-20)*7.5;
}
else if(units<=200)
{
amount=80+80*7.5+(units-100)*8.5;
}
else
{
amount=80+80*7.5+100*8.5+(units-200)*9.5;
}
total=amount+amount*0.15;
printf("Total charges=%f",total);
return 0;
}

```

4. WAP to read the three sides of triangle and print area for the valid data and to print "invalid data" if either one side of the triangle is greater or equals to the sum of other two sides.

(Area= $\sqrt{s(s - a)(s - b)(s - c)}$ ) where a, b, c are the three sides and s=(a+b+c)/2

```

#include<stdio.h>
#include<math.h>
int main()
{
    float a,b,c,s,area;
    printf("Enter the three sides of triangle\n");
    scanf("%f%f%f",&a,&b,&c);
    if(a>=(b+c) || b>=(a+c) || c>=(a+b))
    {
        printf("Invalid data");
    }
    else
    {
        s=(a+b+c)/2;
        area=sqrt(s*(s-a)*(s-b)*(s-c));
        printf("Area of triangle=%f",area);
    }
    return 0;
}

```

5. Consider the following tax table

Income	Tax
< RS.10,000	Nil
Rs. 10,000 to Rs.19,999	10%
RS 20,000 to Rs.29,999	15%
RS 30,000 to Rs.49,999	20%
> =Rs.50,000	25%

Write a program to compute tax amount when income is given.

```
#include<stdio.h>
int main()
{
    float income,tax;
    printf("Enter the income\n");
    scanf("%f",&income);
    if(income<10000)
    {
        tax=0;
    }
    else if(income<20000)
    {
        tax=income*0.1;
    }
    else if(income<30000)
    {
        tax=income*0.15;
    }
    else if(income<50000)
    {
        tax=income*0.2;
    }
    else
    {
        tax=income*0.25;
    }
    printf("Tax amount=%f",tax);
    return 0;
}
```

6. WAP to find whether the entered year is leap year or not.

```
#include<stdio.h>
int main()
{
    int year;
    printf("Enter the year\n");
    scanf("%d",&year);
    if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))
    {
        printf("%d is a leap year", year);
    }
    else
    {
        printf("%d is not a leap year", year);
    }
    return 0;
}
```

7. WAP to read a character from the keyboard and convert it into uppercase if it is in lowercase and vice versa.

```
#include<stdio.h>
int main()
{
    char ch;
    printf("Enter the character\n");
    scanf("%c",&ch);
    if(ch>=65&& ch<=90)
    {
        printf("The character is uppercase\n");
        ch=ch+32;
        printf("The equivalent lowercase character is %c",ch);
    }
    else if(ch>=97&&ch<=122)
    {
        printf("The character is lowercase\n");
        ch=ch-32;
        printf("The equivalent uppercase character is %c",ch);
    }
    else
    {
        printf("Invalid Input\n");
    }
    return 0;
}
```

**8. WAP to read a temperature in of a day in Fahrenheit and print**

"Nice day" if temperature is greater than 60 but less than 80

"Cold day" if temperature is 60 or lower

"Hot day" if temperature is 80 or higher

**9. WAP to input a character and determine whether it is uppercase, lowercase , digits or special symbols.**

```
#include<stdio.h>
int main()
{
int temp;
printf("Enter the temperature in Fahrenheit\n");
scanf("%d",&temp);
if(temp<=60)
{
printf("Cold Day");
}
else if(temp<80)
{
printf("Nice day");
}
else
{
printf("Hot day");
}
return 0;
}
```

```
#include<stdio.h>
int main()
{
char ch;
printf("Enter the character\n");
scanf("%c",&ch);
if(ch>=97&&ch<=122)
{
printf("You Entered lowercase character");
}
else if(ch>=65&&ch<=90)
{
printf("You Entered Uppercase character");
}
else if(ch>=48&&ch<=57)
{
printf("You Entered Digit");
}
else
{
printf("You Entered special symbol");
}
return 0;
}
```

**10. An organization is dealing with two items say A and B and provides the commission on sale of these items according to the following policies**

**i) Commission rate for item A is 5% upto a sale of Rs.2000.If the sale of item is above 2000 then the commission rate is 6% on extra sale**

**ii)For B 10% upto the sale of Rs.4000.If the sale is above 4000 commission rate is 12% on extra sale.**

**Given the sales of both items, Write a Program to compute the net commission.**

```

#include<stdio.h>
int main()
{
    float sa,sb,ca,cb,nc;
    printf("Enter the sales of a and b\n");
    scanf("%f%f",&sa,&sb);
    if(sa<=2000)
    {
        ca=sa*0.05;
    }
    else
    {
        ca=2000*0.05+(sa-2000)*0.06;
    }
    if(sb<=4000)
    {
        cb=sb*0.1;
    }
    else
    {
        cb=4000*0.1+(sb-4000)*0.12;
    }
    nc=ca+cb;
    printf("The net commission of these items is %f",nc);
    return 0;
}

```

- 11. Write a program to calculate the income tax and net salary having following condition**
- If salary $\geq$ 9000 income tax is 40% of salary**
- If salary  $\geq$ 7500 and salary $\leq$ 8999 then income tax is 30% of salary**
- If salary $\leq$ 7499 then income tax is 20 % of salary**

```

#include<stdio.h>
int main()
{
    float salary,netsalary,tax;
    printf("Enter the salary\n");
    scanf("%f",&salary);
    if(salary<=9000)
    {
        tax=salary*0.4;
    }
    else if(salary<=7500)
    {
        tax=salary*0.3;
    }
}

```

```

else
{
    tax=salary*0.2;
}
netsalary=salary-tax;
printf("Income tax=%f\n",tax);
printf("Net salary=%f",netsalary);
return 0;

```

- 12. A bank has introduced an incentive policy. A bonus of 2% of the balance is given to everyone, irrespective of their balance and 5% is given to female account holder if their balance is more than Rs.5000. Write a program to represent this policy and calculate balance after bonus.**

```

#include<stdio.h>
int main()
{
float balance;
char gender;
printf("Enter the balance\n");
scanf("%f",&balance);
printf("Enter the gender\n");
fflush(stdin);
scanf("%c",&gender);
if((gender=='f' || gender=='F') && balance>5000)
{
    balance=balance+0.05*balance;
}
else
{
    balance=balance+0.02*balance;
}
printf("The balance after bonus is %f",balance);
return 0;
}

```

- 13. Write a program and draw a flowchart to read a positive integer value and compute the following sequence .If the number is even, half it ,if it is odd ,multiply 3 and add 1 print the result. If the input value is less than 1, print a message containing word “ERROR”.**

```
#include<stdio.h>
int main()
{
int num;
printf("Enter the integer number\n");
scanf("%d",&num);
if(num<1)
{
    printf("ERROR");
}
else
{
    if(num%2==0)
    {
        num=num/2;
    }
    else
    {
        num=num*3+1;
    }
    printf("Number after result=%d",num);
}
return 0;
}
```

- 14. Write a program which asks time in seconds and convert it into hour minutes and seconds.**

```
#include<stdio.h>
int main()
{
    int hour, minute,second,rem;
    printf("Input seconds:");
    scanf("%d", &second);
    hour = second/3600;
    rem=second%3600;
    minute = rem/60;
    second = rem%60;
    printf("H:M:S - %d:%d:%d\n",hour,minute,second);
    return 0;
}
```

15. Compute the total labour cost for building wall of 50 meter long, 0.2 meter wide and 7 meter tall. The length of brick is 10 cm, height is 8 cm and width is 6 cm. Cost of 1 brick to join is 1.50 paisa. (Assume thickness of mortar is negligible)

```
#include <stdio.h>
int main()
{
    float wall_length,wall_width,wall_height,wall_volume;
    float brick_length,brick_height,brick_width,brick_volume;
    float cost_per_brick,total_cost_paisa,total_cost_rupees;
    int total_bricks;

    //Given dimension of wall is in meter now converting to cm
    wall_length=50 *100;
    wall_width = 0.2*100;
    wall_height = 7*100;
    //Given dimesion of brick in cm
    brick_length = 10;
    brick_height = 8;
    brick_width = 6;

    wall_volume = wall_length * wall_width * wall_height;
    brick_volume = brick_length * brick_height * brick_width;
    printf("Wall volume =%f\n",wall_volume);
    printf("Brick volume =%f\n",brick_volume);

    total_bricks = (int) (wall_volume/brick_volume);

    cost_per_brick = 1.50;

    total_cost_paisa = total_bricks*cost_per_brick;

    // Converting the total cost to rupees (100 paisa = 1 rupee)
    total_cost_rupees = total_cost_paisa / 100.0;
    printf("Cost per brick=%f\n",cost_per_brick);
    printf("Total number of bricks required=%d\n",total_bricks);
    printf("Total labor cost= %f rupees\n",total_cost_rupees);
    return 0;
}
```

## PROGRAMS RELATED TO LOOP CONTROL STATEMENT

- 1. Write a program to print even numbers between 20 and 200.**

```
#include<stdio.h>
int main()
{
    int i;
    for(i=21;i<200;i++)
    {
        if (i%2==0)
        {
            printf("%d\n",i);
        }
    }
    return 0;
}
```

- 2. Write a program find sum of n natural numbers.**

```
#include<stdio.h>
int main()
{
    int n,i,sum=0;
    printf("Enter the value of n\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        sum=sum+i;
    }
    printf("The sum of %d natural number is %d",n,sum);
    return 0;
}
```

- 3. Write a program to find sum of even numbers from 1 to 100.**

```
#include<stdio.h>
int main()
{
    int i,esum=0;
    for(i=1;i<=100;i++)
    {
        if(i%2==0)
        {
            esum=esum+i;
        }
    }
    printf(" sum =%d",esum);
    return 0;
}
```

- 4. Write a program to find sum of even and odd numbers from n1 to n2 as well count them.**

```
#include<stdio.h>
int main()
{
    int n1,n2,esum=0,osum=0,i,ecount=0,ocean=0;
    printf("Enter the values of n1 and n2\n");
    scanf("%d%d",&n1,&n2);
    for(i=n1;i<=n2;i++)
    {
        if(i%2==0)
        {
            esum=esum+i;
            ecount++;
        }
        else
        {
            osum=osum+i;
            ocean++;
        }
    }
    printf("sum of even number =%d\n",esum);
    printf("Total number of even number=%d\n",ecount);
    printf("sum of odd number =%d\n",osum);
    printf("Total number of odd number=%d",ocean);
    return 0;
}
```

**Assignment:**

***Write a program to find sum of even and odd numbers from 50 to 500 as well count them.***

***Write a program to count all the integers greater than 100 and less than 200 that are divisible by 5 .***

***Also find the sum of these numbers.***

**5. Write a program to find reverse of a given number.**

```
#include<stdio.h>
int main()
{
int num,rem,rev=0;
printf("Enter any number\n");
scanf("%d",&num);
while(num!=0)
{
rem=num%10;
rev=rev*10+rem;
num=num/10;
}
printf("Reverse of number=%d",rev);
return 0;
}
```

**6. Write a program to check the given number is palindrome or not.**

```
#include<stdio.h>
int main()
{
int num,rem,rev=0,a;
printf("Enter the number\n");
scanf("%d",&num);
a=num;
while(num!=0)
{
rem=num%10;
rev=rev*10+rem;
num=num/10;
}
if(a==rev)
{
printf("Entered number is palindrome");
}
else
{
printf("Entered number is not palindrome");
}
return 0;
}
```

**7. Write a program to find sum of digit of a given number.**

```
#include<stdio.h>
int main()
{
int num,rem,sum=0;
printf("Enter the number\n");
scanf("%d",&num);
while(num!=0)
{
rem=num%10;
sum=sum+rem;
num=num/10;
}
printf("sum of digits of number= %d",sum);
return 0;
}
```

**8. Write a program to check the given number is armstrong number or not.**

```
#include <stdio.h>
#include <math.h>
int main()
{
int num,orgnum,rem,sum=0,n=0;
printf("Enter the number\n ");
scanf("%d", &orgnum);
num=orgnum;
while (num != 0)
{
num =num/10;
++n;
}
num=orgnum;
while (num != 0)
{
rem = num%10;
sum = sum+pow(rem, n);
num=num/10;
}
if(sum == orgnum)
{
    printf("Given number is armstrong number");
}
else
{
    printf("Given number is not an Armstrong number
}
return 0;
}
```

**9. Write a program to check the given number is prime or not.**

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int i,num;
printf("Enter the number\n");
scanf("%d",&num);
if (num <= 1)
{
printf("Entered number is not a prime number");
exit(1);
}
for(i=2;i<num;i++)
{
if(num%i==0)
{
printf("Entered number is not prime number");
break;
}
}
if(num==i)
{
printf("Entered number is a prime number");
}
return 0;
}
```

**10. Write a program to print prime numbers from 1 to 100.**

```
#include<stdio.h>
int main()
{
int i,j;
for(i=1;i<=200;i++)
{
    for(j=2;j<i;j++)
    {
        if(i% j==0)
        {
            break;
        }
    }
    if(i==j)
    {
        printf("%d\n",i);
    }
}
return 0;
}
```

**11. Write a stated program.**

Until the user presses “Y”, read marks of student of 3 subjects and display their total percentage.

**12. Write a program to find factorial of a given number.**

```
#include<stdio.h>
int main()
{
char choice;
float m1,m2,m3,total,per;
do
{
printf("Enter the marks of 3 subjects\n");
scanf("%f%f%f",&m1,&m2,&m3);
total=m1+m2+m3;
per=(total/3);
printf("percentage=%f\n",per);
printf("Do you want to continue\n");
fflush(stdin);
scanf("%c",&choice);
system("cls");
}while(choice=='y'||choice=='Y');
return 0;
}
```

```
#include<stdio.h>
int main()
{
int num,i,fact = 1;
printf("Enter any number \n");
scanf("%d", &num);
for (i= 1; i <= num; i++)
{
fact = fact * i;
}
printf("Factorial of given number= %d \n", fact);
return 0;
}
```

**13. Write a program to generate Fibonacci series upto n<sup>th</sup> term.  
(0,1,1,2,3,5,8,13 ,....)**

```
#include<stdio.h>
int main()
{
int i,n, a,b,c;
a=0;
b=1;
printf("Enter the value of n\n");
scanf("%d",&n);
printf("%d\t%d\t",a,b);
for (i= 1; i <= n-2; i++)
{
c=a+b;
printf("%d\t",c);
a=b;
b=c;
}
return 0;
}
```

**14. WAP to generate the Fibonacci series upto the nth term when initial values are given by user.**

```
#include<stdio.h>
int main()
{
int i,n, a,b,c;
printf("Enter the values of a and b\n");
scanf("%d%d",&a,&b);
printf("Enter the value of n\n");
scanf("%d",&n);
printf("%d\t%d\t",a,b);
for (i= 1; i <= n-2; i++)
{
c=a+b;
printf("%d\t",c);
a=b;
b=c;
}
return 0;
}
```

**15 . Write a program to enter numbers until user press zero(0) and find the sum of supplied numbers**

```
#include<stdio.h>
int main()
{
int sum=0,num;
do
{
printf("Enter number\n");
scanf("%d",&num);
sum=sum+num;
}while(num!=0);
printf("Total sum of Entered
number=%d",sum);
return 0;
}
```

**16. Write a program to convert decimal number to binary number**

```
#include<stdio.h>
int main()
{
int num,i,base=1,sum=0,rem;
printf("Enter the number\n");
scanf("%d",&num);
i=num;
while(num!=0)
{
rem=num%2;
sum=sum+rem*base;
base=base*10;
num=num/2;
}
printf("Binary equivalent of %d is %d",i,sum);
return 0;
}
```

**17. WAP to print all 3 digits Armstrong numbers (Armstrong numbers from 100 to 999)**

```
#include<stdio.h>
int main()
{
int i,rem,sum,num;
for(i=100;i<=999;i++)
{
    num=i;
    sum=0;
    while(num!=0)
    {
        rem=num%10;
        sum=sum+rem*rem*rem;
        num=num/10;
    }
    if(sum==i)
    {
        printf("%d\t",sum);
    }
}
return 0;
}
```

**18. WAP to display multiplication table of a given number.**

```
#include<stdio.h>
int main()
{
int num,i,pro;
printf("Enter the number\n");
scanf("%d",&num);
for(i=1;i<=10;i++)
{
    pro=num*i;
    printf("%d*%d=%d",num,i,pro);
    printf("\n");
}
return 0;
}
```

## **PROGRAMS RELATED TO SWITCH CASE**

- 1) Write a Program to perform the Arithmetic Operation Using Switch.

```
#include<stdio.h>
int main()
{
float a,b,result;
int choice;
printf("Menu\n");
printf("1.Addition\n");
printf("2.Substraction\n");
printf("3.Multiplication\n");
printf("4.Division\n");
printf("Enter the two numbers\n");
scanf("%f%f",&a,&b);
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
result=a+b;
printf("Addition=%f",result) ;
break;
case 2:
result=a-b;
printf("Subtraction=%f",result);
break;
case 3:
result=a*b;
printf("Multiplication=%f",result);
break;
case 4:
result=a/b;
printf("Division=%f",result);
break;
default:
printf("Invalid choice!");
break;
}
return 0;
}
```

- 2. Write a program to perform various Arithmetic operation on the basics of operators given by the user.**

```
#include<stdio.h>
int main()
{
float a,b,result;
char opr;
printf("Menu\n");
printf("1.Addition\n");
printf("2.Subtraction\n");
printf("3.Multiplication\n");
printf("4.Division\n");
printf("Enter the two numbers\n");
scanf("%f%f",&a,&b);
printf("Enter the operator among +,-,/and* \n");
scanf("%c",&opr);
switch(opr)
{
case '+':
result=a+b;
printf("Addition=%f",result) ;
break;
case '-':
result=a-b;
printf("Subtraction=%f",result);
break;
case '*':
result=a*b;
printf("Multiplication=%f",result);
break;
case '/':
result=a/b;
printf("Division=%f",result);
break;
default:
printf("Invalid choice!");
}
return 0;
}
```

- 3) Write a program to display the following menu and perform the following operations.**

- 1. Find the simple interest**
- 2. Convert degree Celsius to Fahrenheit**
- 3. Convert character into ASCII code**
- 4. Exit from the program**

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
float p,t,r,si,cel,fah,area;
char ch;
int choice;
printf("Menu\n");
printf("1.Find simple interest\n");
printf("2.Convert celsius of Fahrenheit\n");
printf("3.Convert character to ASCII code\n");
printf("4.Exit from program\n");
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Enter Principal,Time and Rate\n");
scanf("%f%f%f",&p,&t,&r);
si=(p*t*r)/100;
printf("simple interest is %f\n",si);
break;
case 2:
printf("Enter the temperature in celsius\n");
scanf("%f",&cel);
fah=1.8*cel+32;
printf("Temperature in fahrenheit is %f\n",fah);
break;
case 3:
printf("Enter a character\n");
scanf("%c",&ch);
printf("The corresponding ASCII code is %d\n",ch);
break;
case 4:
exit(0);
break;
default:
printf("Wrong choice!");
}
return 0;
}

```

**4)Write a menu program to satisfy the following function**

- 1. Check whether the given string is palindrome or not**
- 2. Find all the prime numbers from 100 to 200**
- 3. Display all ASCII characters from 0 to 255.**
- 4. Exit from the program**

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
char str1[20],str2[20];
int choice,i,j;
printf("Menu\n");
printf("1.Check whether the given string is palindrome or not\n");
printf("2.Find all prime numbers from 100 to 200\n");
printf("3.Display all ascii characters from 0 to 255\n");
printf("4.Exit from program\n");
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Enter the string\n");
gets(str1);
strcpy(str2,str1);
strrev(str2);
if(strcmp(str1,str2)==0)
{
printf("String is palindrome\n");
}
else
{
printf("String is not palindrome\n");
}
break;
```

```

case 2:
for(i=100;i<=200;i++)
{
    for(j=2;j<i;j++)
    {
        if(i% j==0)
        {
            break;
        }
    }
    if(i==j)
    printf("%d\t",i);
}
break;
case 3:
for(i=0; i<=255; i++)
{
printf("ASCII value of character %c = %d\n", i, i);
}
break;
case 4:
exit(0);
break;
default:
printf("Wrong choice!");
}
return 0;
}

```

**5) Write a menu driven program to work following cases, take appropriate input whenever required.**

- 1. Reverse a number**
- 2. Find sum of individual unit**
- 3. Check for prime**
- 4. Exit**

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
int choice,i,rem,sum=0,rev=0,num;
printf("Menu\n");
printf("1.Reverse a number\n");
printf("2.Find sum of individual unit\n");
printf("3.Check for prime\n");
printf("4.Exit from program\n");

```

```

printf("Enter the number\n");
scanf("%d",&num);
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
while(num!=0)
{
rem=num%10;
rev=rev*10+rem;
num=num/10;
}
printf("Reverse of number=%d\n",rev);
break;
case 2:
while(num!=0)
{
rem=num%10;
sum=sum+rem;
num=num/10;
}
printf("The sum of digits of number=%d\n",sum);
break;
case 3:
if (num <= 1)
{
printf("Entered number is not a prime number\n");
}
for(i=2;i<num;i++)
{
if(num%i==0)
{
printf("Entered number is not prime number\n");
break;
}
}
if(num==i)
{
printf("Entered number is a prime number");
}
break;
case 4:
exit(0);
break;
default:
printf("Wrong choice!");
}
return 0; }

```

**6) Write a program to display the following menu**

- 1. Conversion of ASCII code to char**
- 2. To find the sum of n natural numbers**
- 3. Exit from the program**

**and perform task as per users choice repeatedly until his/her choice is to exit.**

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int choice,n,sum=0,i,code;
while(1)
{
printf("\nMenu\n");
printf("1.Conversion of ASCII code to char\n");
printf("2.Find sum of n natural numbers\n");
printf("3.Exit from program\n");
printf("Enter your choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("Enter the ASCII code\n");
scanf("%d",&code);
printf("character corresponding to %d is %c\n",code,code);
break;
case 2:
printf("Enter the value of n\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
sum=sum+i;
}
printf("sum=%d\n",sum);
break;
case 3:
exit(0);
break;
default:
printf("Wrong choice!");
}
}
return 0;
}
```

## Write Program to generate the following pattern.

1)        1  
        1   2  
    1   2   3

2)    1  
    1   2  
  1   2   3  
1   2   3   4  
1   2   3   4   5

```
#include<stdio.h>
int main()
{
int i,j;
for(i=1;i<=3;i++)
{
    for(j=1;j<=i;j++)
    {
        printf("%d\t",j);
    }
    printf("\n");
}
return 0;
}
```

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

3)        1  
        2   2  
    3   3   3  
  4   4   4   4  
5   5   5   5   5

4)        1  
    1   1  
  1   1   1  
1   1   1   1  
1   1   1   1   1

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

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

5) \*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

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

7) 1

11 21

31 41 51

61 71 81 91

101 111 121 131 141

```
#include<stdio.h>
int main()
{
int i,j,k=1;
for(i=1;i<=5;i++)
{
    for(j=1;j<=i;j++)
    {
        printf("%d\t",k);
        k=k+10;
    }
    printf("\n");
}
return 0;
}
```

6) 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

```
#include<stdio.h>
int main()
{
int i,j,a=1;
for(i=1;i<=5;i++)
{
    for(j=1;j<=i;j++)
    {
        printf("%d\t",a);
        a++;
    }
    printf("\n");
}
return 0;
}
```

8) 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

```
#include<stdio.h>
int main()
{
int i,j;
for(i=1;i<=5;i++)
{
    for(j=1;j<=5;j++)
    {
        printf("%d\t",i*j);
    }
    printf("\n");
}
return 0;
}
```

9)

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

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

11)

```
1 1  
1 2 2  
1 2 3 3  
1 2 3 4 4  
1 2 3 4 5
```

```
#include<stdio.h>  
int main()  
{  
    int i,j,k;  
    for(i=1;i<=5;i++)  
    {  
        for(k=4;k>=i;k--)  
        {  
            printf("\t");  
        }  
        for(j=1;j<=i;j++)  
        {  
            printf("%d\t",j);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

10)

```
1  
2 2  
3 3 3  
4 4 4 4  
5 5 5 5 5
```

```
#include<stdio.h>  
int main()  
{  
    int i,j,k;  
    for(i=1;i<=5;i++)  
    {  
        for(k=4;k>=i;k--)  
        {  
            printf("\t");  
        }  
        for(j=1;j<=i;j++)  
        {  
            printf("%d\t",i);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

12)

```
5 4 3 2 1  
5 4 3 2  
5 4 3  
5 4  
5
```

```
#include<stdio.h>  
int main()  
{  
    int i,j;  
    for(i=1;i<=5;i++)  
    {  
        for(j=5;j>=i;j--)  
        {  
            printf("%d\t",j);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

**13)** \* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**14)** 5 5 5 5 5

4 4 4 4

3 3 3

2 2

1

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

```
#include<stdio.h>
int main()
{
int i,j;
for(i=5;i>=1;i--)
{
    for(j=1;j<=i;j++)
    {
        printf("%d\t",i);
    }
    printf("\n");
}
return 0;
}
```

**15)** 1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

**16)** 1 1 1 1 1

1 1 1 1

1 1 1

1 1

1

```
#include<stdio.h>
int main()
{
int i,j;
for(i=5;i>=1;i--)
{
    for(j=1;j<=i;j++)
    {
        printf("%d\t",j);
    }
    printf("\n");
}
return 0;
}
```

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

**17)**

```

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

```

```

#include <stdio.h>
int main()
{
    int i,space,j,n=5;
    for (i=1;i<=5;i++)
    {
        for(space=1;space<=n-i;space++)
        {
            printf(" ");
        }
        for(j=1;j<=2*i-1;j++)
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}

```

**19) 1 2 3 4 5 6 7 8 9**

```

  1 2 3 4 5 6 7
  1 2 3 4 5
  1 2 3
  1

```

```

#include <stdio.h>
int main() {
    int i, j, space, n=5;
    for(i=n;i>=1;i--)
    {
        for(space=1;space<=n-i;space++)
        {
            printf(" ");
        }
        for(j=1;j<=2*i-1;j++)
        {
            printf("%d ", j);
        }
        printf("\n");
    }
    return 0;
}

```

**18)**

```

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

```

```

#include <stdio.h>
int main()
{
    int i,space,j,n=5;
    for (i=1;i<=5;i++)
    {
        for(space=1;space<=n-i;space++)
        {
            printf(" ");
        }
        for(j=1;j<=2*i-1;j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}

```

**20)**

```

      1
    1 1
  1 2 1
1 3 3 1
1 4 6 4 1

```

```

#include <stdio.h>
int main()
{
    int i, j, space, val, n=5;
    for (i=1;i<=n;i++)
    {
        for (space=1;space<=n-i; space++)
        {
            printf(" ");
        }
        val = 1;
        for (j = 1; j <= i; j++)
        {
            printf("%d ", val);
            val = val * (i - j) / j;
        }
        printf("\n");
    }
    return 0;
}

```

**Find the output of the following program.(Assume necessary header files)**

```
void main()
{
int i,j,k,c;
i=0,j=1;
for(k=0;k<5;k++)
{
i--;
j++;
c=i+j;
}
printf("value of i=%d,j=%d and c=%d",i,j,c);
return 0;
}
```

**Solution:**

Tracing the above program,

k	Condition (k<5)	i=0	j=1	C
		i --	j++	c=i+j
0	0<5 (T)	-1	2	c= -1+2=1
1	1<5 (T)	-2	3	c=-2+3=1
2	2<5 (T)	-3	4	c=-3+4=1
3	3<5 (T)	-4	5	c=-4+5=1
4	4<5 (T)	-5	6	c=-5+6=1
5	5<5 (F) Now Loop Terminates			

**Output:** i=-5, j=6 and c=1

**Assignment:**

**Find the output of the following program.(Assume necessary header files)**

```
int main()
{
int i,j;
for(i=0;i<=20;i++)
{
printf("%d\n",i++);
}
return 0;
}
```

**(Trace the program yourself)**

**Output:**

```
0  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20
```

## PROGRAMS RELATED TO SERIES

**Write a program to find the sum of series.**

$$1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + \dots + n^2$$

```
#include<stdio.h>  
int main()  
{  
    int i,sum=0,n;  
    printf("Enter the value of n\n");  
    scanf("%d",&n);  
    for(i=1;i<=n;i++)  
    {  
        sum=sum+(i*i);  
    }  
    printf("sum=%d",sum);  
    return 0;  
}
```

**What will be the sum of the given series.**

$$1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \frac{1}{x^4} + \dots + \frac{1}{x^n}$$

```
#include<stdio.h>  
#include<math.h>  
int main()  
{  
    int n,i;  
    float x,sum=0;  
    printf("Enter the value of x\n");  
    scanf("%f",&x);  
    printf("Enter the value of n\n");  
    scanf("%d",&n);
```

```

        for(i=0;i<=n;i++)
    {
        sum=sum+(1/pow(x,i));
    }
    printf("sum=%f",sum);
    return 0;
}

```

**What will be the sum of the given series.**

$$1+x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!}$$

```

#include<stdio.h>
#include<math.h>
int main()
{
int n,i,j,fact;
float x,sum=1;
printf("Enter the value of x\n");
scanf("%f",&x);
printf("Enter the value of n\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
    fact=1;
    for(j=1;j<=i;j++)
    {
        fact=fact*j;
    }
    sum=sum+pow(x,i)/(float)fact;
}
printf("sum=%f",sum);
return 0;
}

```

**Write a program to generate the following series and print the sum.**

**1x4 2x7 3x10 4x13 . . . . . n terms**

```

#include<stdio.h>
int main()
{
int i,j,n,sum=0;
printf("Enter the number of term\n");
scanf("%d",&n);
for(i=1,j=4;i<=n;i++,j+=3)
{
printf("%dx%d\t",i,j);
sum=sum+(i*j);
}
printf("sum =%d",sum);
return 0; }

```

**Write a program to read x, n and generate the following series and print sum.**

$x + 2/x^2 - 3/x^3 + 4/x^4 - 5/x^5 \dots \dots \dots n$  terms.

```
#include<stdio.h>
#include<math.h>
int main()
{
int i,x,n;
float a,sum=0;
printf("Enter the value of x\n");
scanf("%d",&x);
printf("Enter the value of n\n");
scanf("%d",&n);
if(n==0)
{
    printf("There is no term\n");
}
if(n==1)
{
    printf("The term is %d\n",x);
}
if(n>=2)
{
    printf("The term is \t%d",x);
    for(i=2;i<=n;i++)
    {
        a=(pow(-1,i)*i)/pow(x,i);
        printf("\t%f",a);
        sum=sum+a;
    }
    printf("The sum of terms=%f",sum+x);
}
return 0;
}
```

**Write a program to compute the given series.**

$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! \dots \dots$

```
#include<stdio.h>
int main()
{
int i,n,sign=1;
float x,nume,deno,sum=0,val,term;
printf(" Enter the value for x in degree\n");
scanf("%f",&x);
printf(" Enter the number of terms \n");
scanf("%d",&n);
```

```

val=x;
x=x*3.14159/180;
nume=x;
deno=1;
sum=x;
for(i=2;i<=n;i++)
{
    nume=nume*x*x;
    deno=deno*(i*2-2)*(i*2-1);
    sign=sign*(-1);
    term=nume/deno*sign;
    sum=sum+term;
}
printf("sin(%f)=%f",val,sum);
return 0;
}

```

**WAP to compute the given series.**

**Cos( x)= 1- x<sup>2</sup>/2! + x<sup>4</sup>/4! – x<sup>6</sup>/6! .....**

```

#include<stdio.h>
int main()
{
    int i,n,sign=1;
    float x,nume,deno,sum=0,val,term;
    printf(" Enter the value for x in degree\n");
    scanf("%f",&x);
    printf(" Enter the number of terms\n");
    scanf("%d",&n);
    val=x;
    x=x*3.14159/180;
    nume=1;
    deno=1;
    sum=1;
    for(i=2;i<=n;i++)
    {
        nume=nume*x*x;
        deno=deno*(i*2-3)*(i*2-2);
        sign=sign*(-1);
        term=nume/deno*sign;
        sum=sum+term;
    }
    printf("cos(%f)=%f",val,sum);
    return 0;
}

```