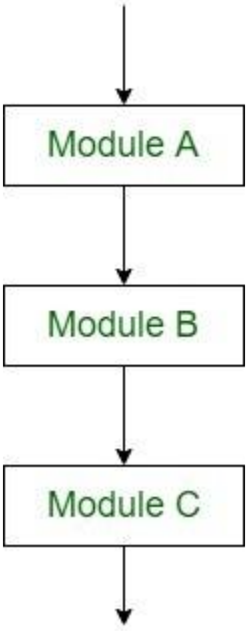
Control Structures

Control Structures are just a way to specify flow of control in programs. Any algorithm or program can be more clear and understood if they use self-contained modules called as logic or control structures. It basically analyzes and chooses in which direction a program flows based on certain parameters or conditions. There are three basic types of logic, or flow of control, known as:

1. Sequence logic, or sequential flow
2. Selection logic, or conditional flow
3. Iteration logic, or repetitive flow

Sequential Logic (Sequential Flow)

Sequential logic as the name suggests follows a serial or sequential flow in which the flow depends on the series of instructions given to the computer. Unless new instructions are given, the modules are executed in the obvious sequence. The sequences may be given, by means of numbered steps explicitly. Also, implicitly follows the order in which modules are written. Most of the processing, even some complex problems, will generally follow this elementary flow pattern.



Examples: Simple Sequential Programs

1. Write a C program to find the sum of individual digits of a positive 3 digit integer.

```
//Program to find the sum of digits of a 3-digit number
#include
main()
{   int num, sum;
    printf("Enter a 3-digit number : ");
    scanf("%d" , &num);
    sum = num/100;
     num%=100;
    sum+=num/10;
    num%=10;
    sum+=num;
    printf("Sum of digits = %d\n" , sum);
}
```

2. Write a C program to covert Fahrenheit to Celsius.

```
//Program to convert Temperature from Fahrenheit to Celsius Scale
#include
main()
{   float fah, cel;
    printf ("Enter temperature in Fahrenheit scale : ");
    scanf ("%f", &fah);
    cel = 5*(fah-32)/9;
    printf("%f Fahrenheit = %f Celsius\n", fah, cel);
}
```

3. WAP to swap the values of 3 variables without using 4th variable

```
//Program to swap the value of 3 integer variables without using 4th variable
#include
main()
{   int a, b, c;
    printf ("Enter values for a, b, and c : ");
    scanf ("%d %d %d", &a, &b, &c);
    a = a+b+c;
    c = a-(b+c);
    b = a-(b+c);
    a = a-(b+c);
    printf("a = %d, b = %d, C = %d \n", a, b, c);
}
```

4. WAP to calculate the hours, minutes and seconds by giving seconds as input.

```
//Program to convert seconds into hours, minutes and seconds
#include
main()
{   unsigned long int sec, hour, min;
    printf("Enter time in seconds : ");
    scanf ("%d", &sec);
        //There is a scope for simplifying and enhancing the logic
    hour = sec/3600;
    min = (sec/60)-(hour*60);
    sec -= (hour*3600)+(min*60);
    printf("Time = %d hour(s) %d min(s) %d sec(s)\n", hour, min, sec);
}
```

5. WAP to reverse a 3-digit number

```
/* Program to reverse a 3-digit number
    For Ex : If input - 123, output = 321    */
#include
main()
{   int num, a, b, c, rev;
    printf("Enter a 3-digit number : ");
    scanf("%d", &num);
    a = num%10;
    num/=10;
    b = num%10;
    num/=10;
    c = num;
    rev = a*100 + b*10 + c;
    printf ("The reverse of entered number is %d \n", rev);
}
```

6. WAP to perform flooring and ceiling operation on a real number.

```
/*  Program to perform flooring and ceiling operation on a number without using inbulit funct
#include
main()
{
    float num;
        int floor, ceil;
    printf ("Enter a real number : ");
    scanf ("%f", &num);
    floor = (int)num;
    ceil = floor + 1;
    printf ("Real number = %f", num);
        printf ("\nFloor = %d", floor);
        printf ("\nCeil = %d", ceil);
}
```

7. WAP to enter a character and display its ASCII equivalent in decimal and hexadecimal.

```
//Program to display the ASCII code of a character in decimal as well as hexadecimal
#include
main()
{   char ch;
    printf("Enter a Character : ");
    scanf("%c", &ch);
    printf("ASCII value of %c in decimal = %d \n", ch, ch);
    printf("ASCII value of %c in hexadecimal = %x \n", ch, ch);
}
```