# Conditional Programs

04 September 2021 07:59 PM

The most basic and simple kind of conditional operator is the if else. It is of four types

1. If statement:

This checks if a certain condition is true and then executes a block of statement otherwise not.

Syntax:

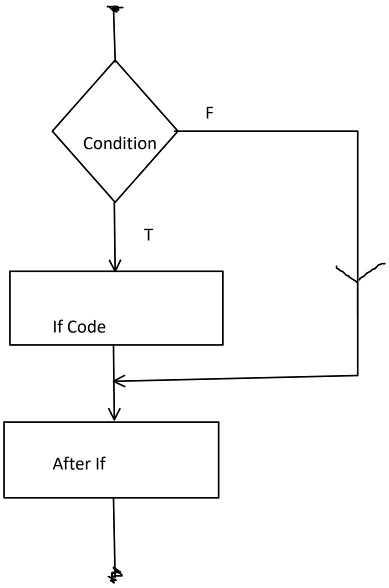
If(condition)

{

//Code to be executed

}

Here if you do not give the {} then the statement right after the if statement is considered for execution.



Example for if statement

```
#include<stdio.h>
int main(){
        int age;
        printf("\nEnter your age : ");
        scanf("%d", &age);
        if (age > 18)
        {
            printf("\nYou are eligible for vote.");
        }
        printf("\nThank you for using voting service.");
    }
}
Output:
Enter your age : 37
You are elligible for vote.
Thank you for using voting service.
```

2. If-else statement:

It is quite similar to the if statement the difference comes when condition is false in that case the block of commands in the else is executed.

Syntax:

If(condition)

{

//Code to be executed when condition is true

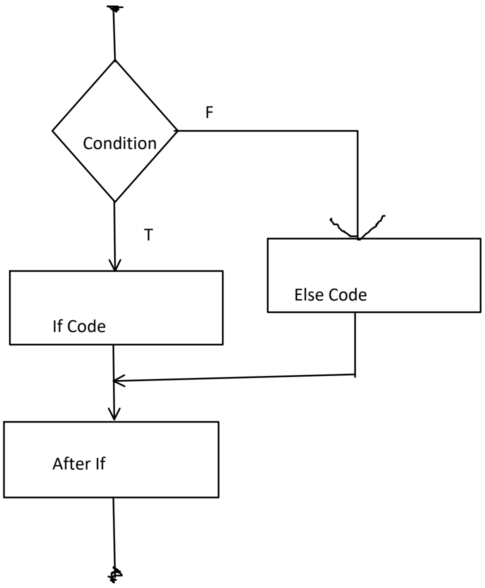
}

else

{

//Code to be executed when condition is false

}



```
#include <stdio.h>
int main()
{
    int age;
    printf("\nEnter your age : ");
    scanf("%d", &age);
    if (age > 18)
    {
        printf("\nYou are eligible for vote.");
    }
    else
    {
        printf("\nYou are not eligible for vote.");
    }
    printf("\nThank you for using voting service.");
    return 0;
}
```

Q. Write a program to input an amount and then print the total number of different notes.

For example:

Input:

Enter amt: 2442

Output:

Total number of notes =

2000 = 1

500 = 0

100 = 4

50 = 0

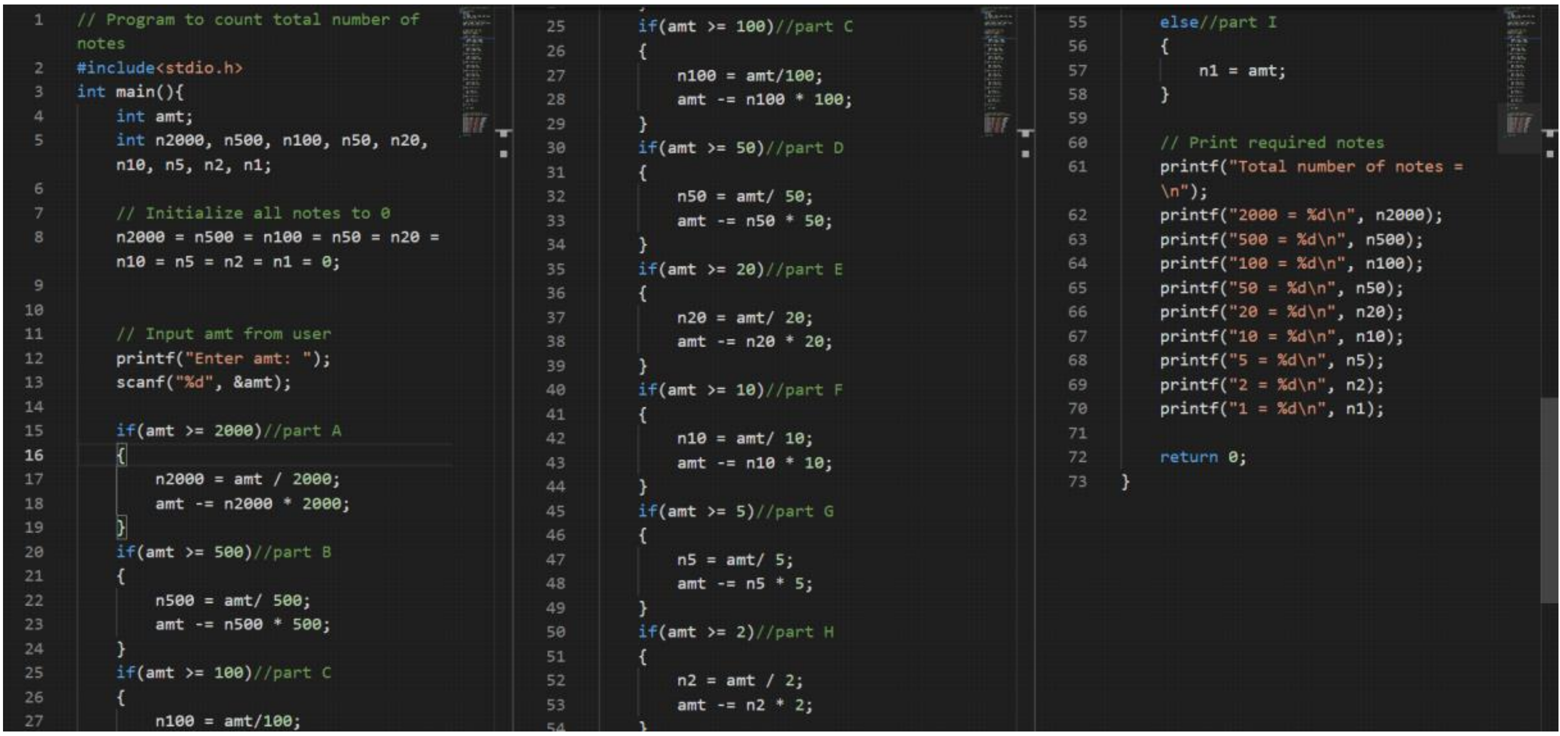
20 = 2

10 = 0

5 = 0

2 = 1

1 = 0



```c
// Program to count total number of notes
#include<stdio.h>
int main(){
    int amt;
    int n2000, n500, n100, n50, n20, n10, n5, n2, n1;

    // Initialize all notes to 0
    n2000 = n500 = n100 = n50 = n20 = n10 = n5 = n2 = n1 = 0;


    // Input amt from user
    printf("Enter amt: ");
    scanf("%d", &amt);

    if(amt >= 2000)//part A
    {
        n2000 = amt / 2000;
        amt -= n2000 * 2000;
    }
    if(amt >= 500)//part B
    {
        n500 = amt/ 500;
        amt -= n500 * 500;
    }
    if(amt >= 100)//part C
    {
        n100 = amt/100;
```

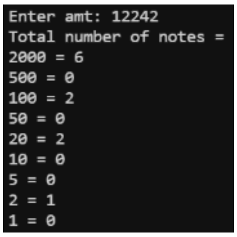
```c
    if(amt >= 100)//part C
    {
        n100 = amt/100;
        amt -= n100 * 100;
    }
    if(amt >= 50)//part D
    {
        n50 = amt/ 50;
        amt -= n50 * 50;
    }
    if(amt >= 20)//part E
    {
        n20 = amt/ 20;
        amt -= n20 * 20;
    }
    if(amt >= 10)//part F
    {
        n10 = amt/ 10;
        amt -= n10 * 10;
    }
    if(amt >= 5)//part G
    {
        n5 = amt/ 5;
        amt -= n5 * 5;
    }
    if(amt >= 2)//part H
    {
        n2 = amt / 2;
        amt -= n2 * 2;
    }
```

```c
    else//part I
    {
        n1 = amt;
    }

    // Print required notes
    printf("Total number of notes =\n");
    printf("2000 = %d\n", n2000);
    printf("500 = %d\n", n500);
    printf("100 = %d\n", n100);
    printf("50 = %d\n", n50);
    printf("20 = %d\n", n20);
    printf("10 = %d\n", n10);
    printf("5 = %d\n", n5);
    printf("2 = %d\n", n2);
    printf("1 = %d\n", n1);

    return 0;
}
```

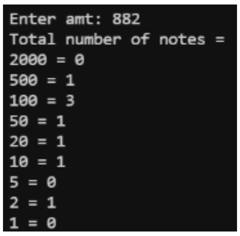
Basically, what we are doing over here is we input the amount from the user and then store it in a variable amount. We then check if the amount is greater than x then we divide amount buy x to get maximum x notes required. This is stored in another variable after the division we subtract the resultant amount of x notes from original amount performed as amt = amt – (nx*x).
Here x is 2000 or 500 or 200 or 100 or 50 or 20 or 10 or 5 or 2 or 1.

Conditional Programs

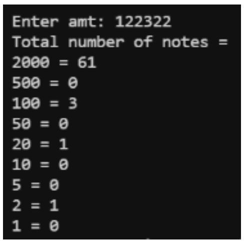
Audio recording started: 11:43 AM 05 September 2021



```
Enter amt: 12242
Total number of notes =
2000 = 6
500 = 0
100 = 2
50 = 0
20 = 2
10 = 0
5 = 0
2 = 1
1 = 0
```



```
Enter amt: 882
Total number of notes =
2000 = 0
500 = 1
100 = 3
50 = 1
20 = 1
10 = 1
5 = 0
2 = 1
1 = 0
```

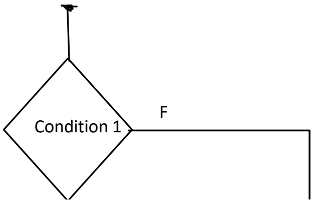


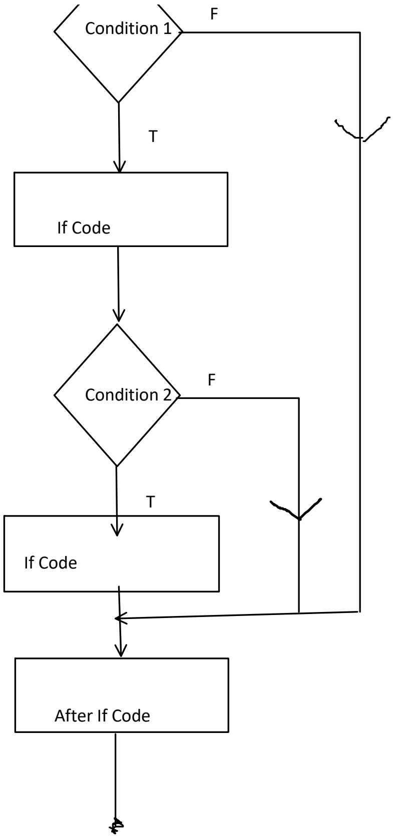
```
Enter amt: 122322
Total number of notes =
2000 = 61
500 = 0
100 = 3
50 = 0
20 = 1
10 = 0
5 = 0
2 = 1
1 = 0
```

3. Nested If statement:

Here an if statement is placed inside another if statement. the syntax for it is given below:

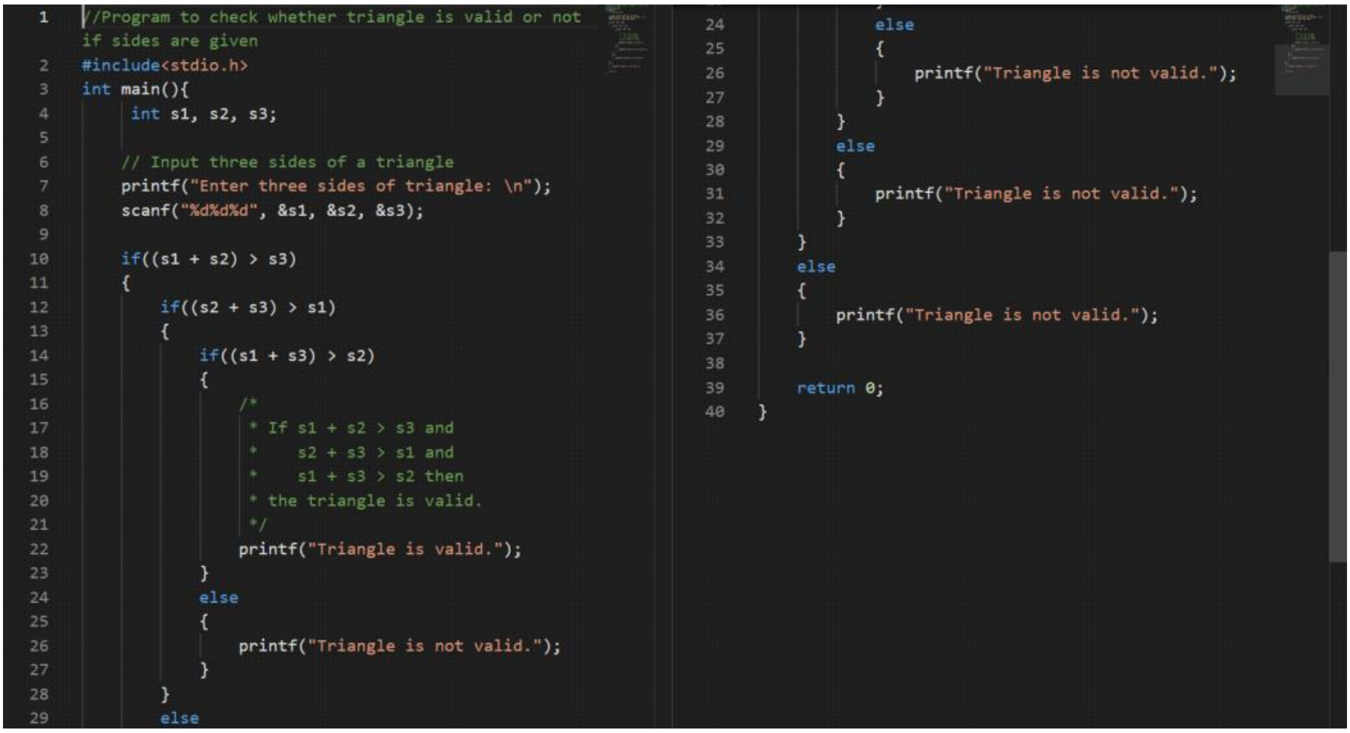
```
Syntax:
If(condition1)
{
//Code to be executed when condition1 is true
        If(condition2)

     {

     //Code to be executed when condition2 is true

     }

 }
```





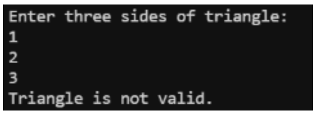
Q. Write a program to check if a triangle is valid when the user enters the three sides.

Hint: In a triangle if the sum of any two sides is greater than the 3rd then the triangle is said to be valid.

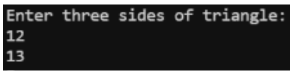


```
//Program to check whether triangle is valid or not if sides are given
#include<stdio.h>
int main(){
    int s1, s2, s3;

    // Input three sides of a triangle
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &s1, &s2, &s3);

    if((s1 + s2) > s3)
    {
        if((s2 + s3) > s1)
        {
            if((s1 + s3) > s2)
            {
                /*
                 * If s1 + s2 > s3 and
                 *    s2 + s3 > s1 and
                 *    s1 + s3 > s2 then
                 * the triangle is valid.
                 */
                printf("Triangle is valid.");
            }
            else
            {
                printf("Triangle is not valid.");
            }
        }
        else
        {
            printf("Triangle is not valid.");
        }
    }
    else
    {
        printf("Triangle is not valid.");
    }

    return 0;
}
```

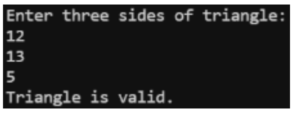
First we ask the user to enter the 3 sites in some varia ble say s1, s2 and s3. Then we check if s1 + s2 is greater than s3 AND s1 + s3 is greater than s2 AND s2 + s3 is greater than s1. Also if anyone of them is false then the triangle is invalid.



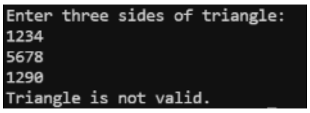
```
Enter three sides of triangle:
1
2
3
Triangle is not valid.
```



```
Enter three sides of triangle:
12
13
```



```
Enter three sides of triangle:
12
13
5
Triangle is valid.
```

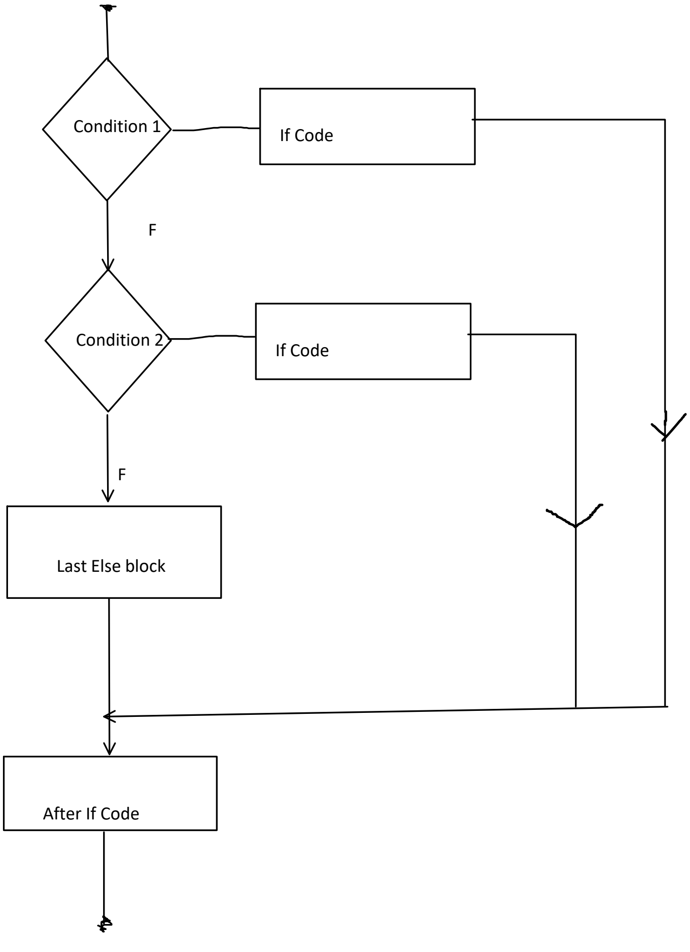


```
Enter three sides of triangle:
1234
5678
1290
Triangle is not valid.
```

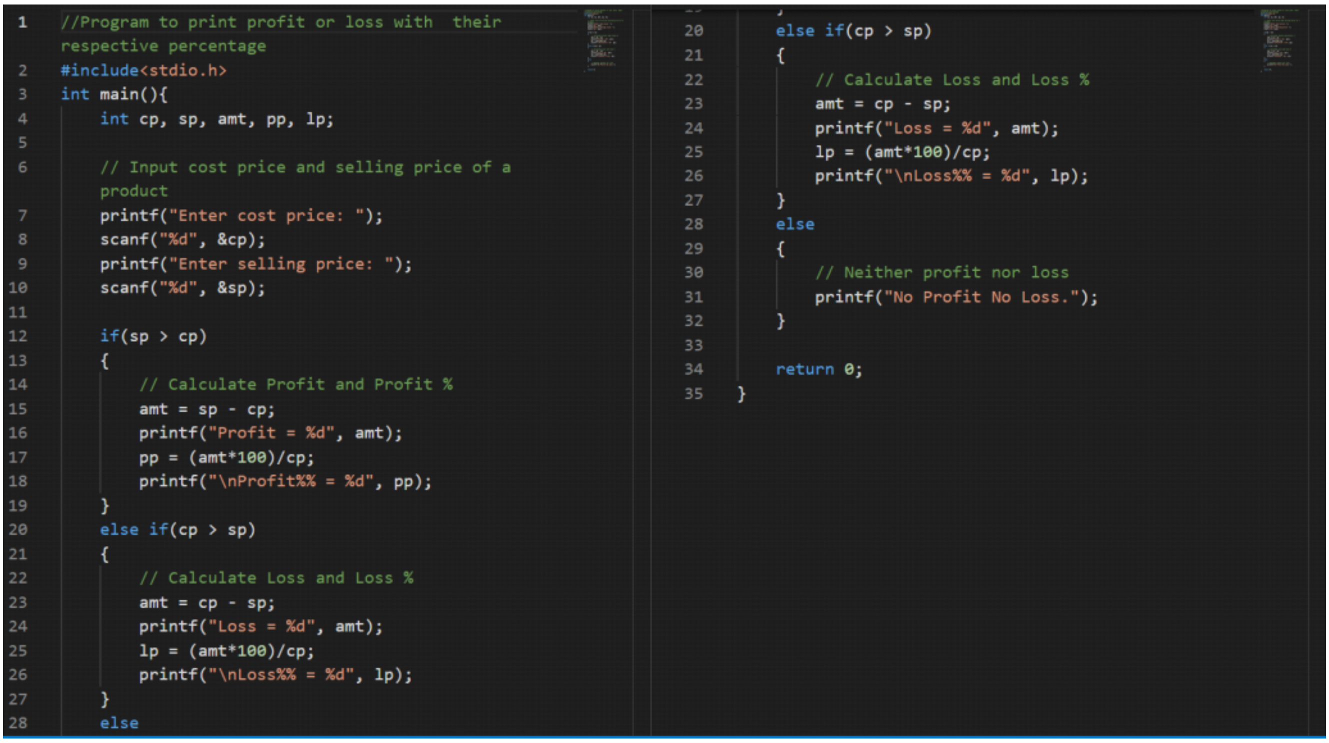
4. if-else-if ladder statement:

   If any of the conditions is true, then the statements in that particular if block is executed The statements in the last else block is executed if and only if none of the above conditions is true. The syntax for if else if ladder is given below.

```
if(condition1)
   {
   //Code to be executed when condition1 is true
    }
else if(condition2)
   {
   //Code to be executed when condition2 is true
   }
 else
   {
   //Code to be executed when all the conditions is false
   }
```

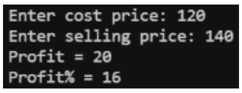


Q. Write a program to print profit or loss with the respective percentage when the user enters the cost price and the selling pr ice.

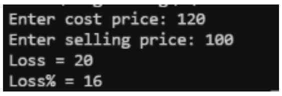


```c
//Program to print profit or loss with  their respective percentage
#include<stdio.h>
int main(){
    int cp, sp, amt, pp, lp;

    // Input cost price and selling price of a product
    printf("Enter cost price: ");
    scanf("%d", &cp);
    printf("Enter selling price: ");
    scanf("%d", &sp);

    if(sp > cp)
    {
        // Calculate Profit and Profit %
        amt = sp - cp;
        printf("Profit = %d", amt);
        pp = (amt*100)/cp;
        printf("\nProfit%% = %d", pp);
    }
    else if(cp > sp)
    {
        // Calculate Loss and Loss %
        amt = cp - sp;
        printf("Loss = %d", amt);
        lp = (amt*100)/cp;
        printf("\nLoss%% = %d", lp);
    }
    else
    {
        // Neither profit nor loss
        printf("No Profit No Loss.");
    }

    return 0;
}
```

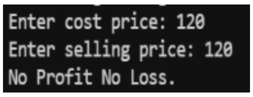
In this program we are using the basic mathematical knowledge of calculating a profit and loss as we know profit will happen only once selling price is greater than cost price and loss will happen only when cost price is greater than selling price. hence three things can happen either cost price is greater than selling price or selling price is greater than cost price or both of them are equal therefore we use if-else-if ladder over here.



```
Enter cost price: 120
Enter selling price: 140
Profit = 20
Profit% = 16
```



```
Enter cost price: 120
Enter selling price: 100
Loss = 20
Loss% = 16
```



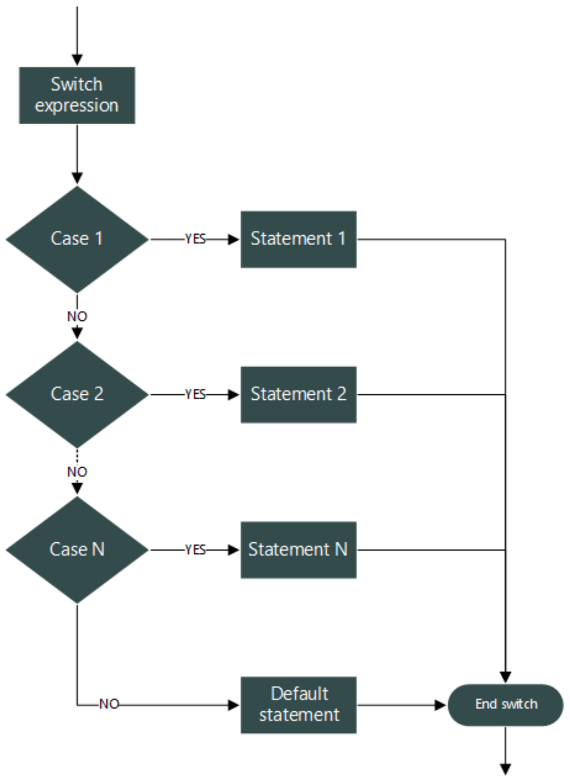
```
Enter cost price: 120
Enter selling price: 120
No Profit No Loss.
```

5. Switch Statement:

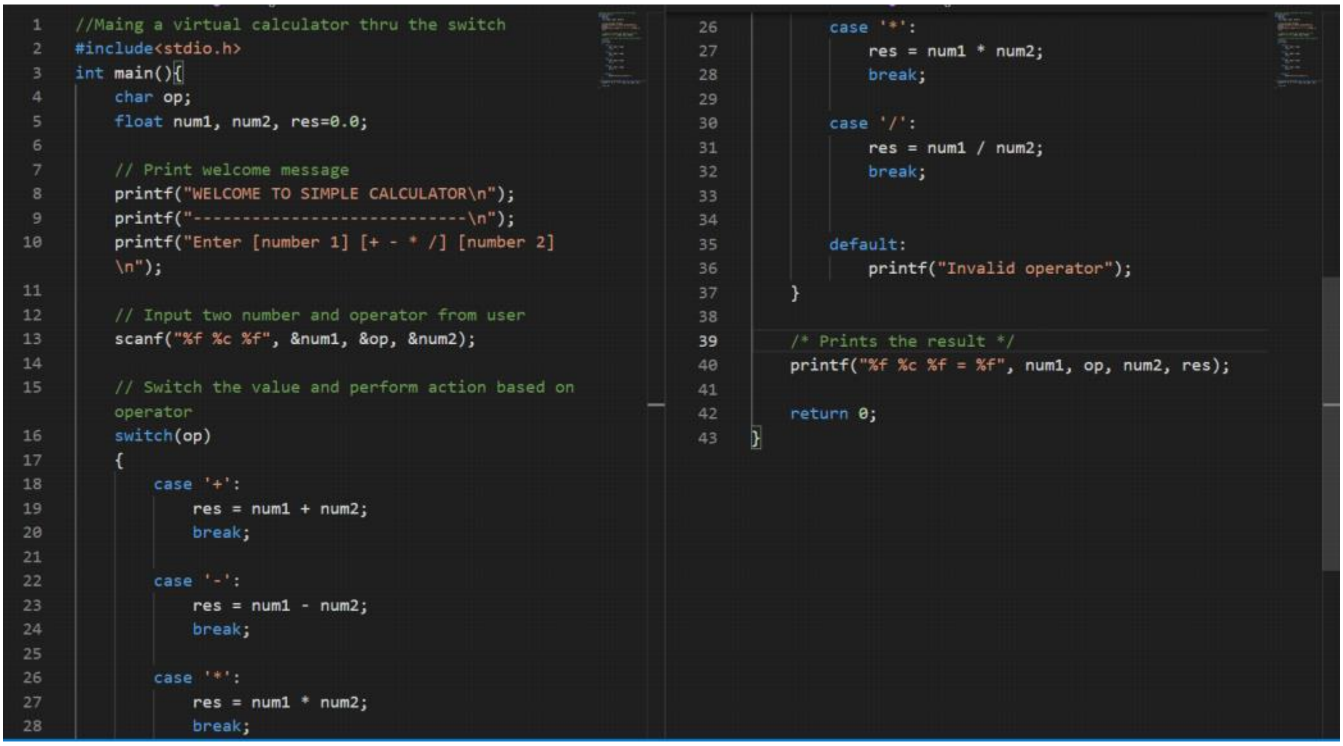
The switch statement is used when the else-if ladder becomes too long and complex. Switch is a multi-branch statement. It is one of the only statement that allows the value to change control of execution.

Syntax:

```
 switch(variable)
{
   case value1: //code to be executed
         break;
      case value2: //code to be executed
         break;
      case value3: //code to be executed
         break;
      case value4: //code to be executed
         break;
   default: //code to be executed when none of the above matches
}
```

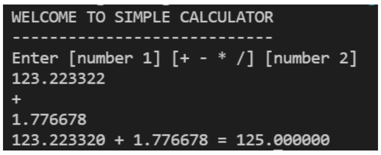


Q. Write a program to make a virtual character with using only switch statement.

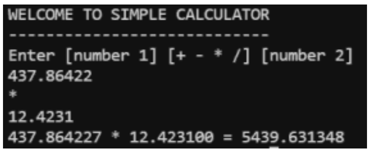


```c
//Maing a virtual calculator thru the switch
#include<stdio.h>
int main(){
    char op;
    float num1, num2, res=0.0;

    // Print welcome message
    printf("WELCOME TO SIMPLE CALCULATOR\n");
    printf("----------------------------\n");
    printf("Enter [number 1] [+ - * /] [number 2]\n");

    // Input two number and operator from user
    scanf("%f %c %f", &num1, &op, &num2);

    // Switch the value and perform action based on operator
    switch(op)
    {
        case '+':
            res = num1 + num2;
            break;

        case '-':
            res = num1 - num2;
            break;

        case '*':
            res = num1 * num2;
            break;

        case '/':
            res = num1 / num2;
            break;


        default:
            printf("Invalid operator");
    }

    /* Prints the result */
    printf("%f %c %f = %f", num1, op, num2, res);

    return 0;
}
```

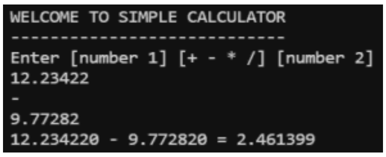
In this program we take three input from the user, two numbers and a character in the given format and store them in some variable num1, op and num2. Next we switch the value of op for four possible cases, i.e., ‘+’, ‘-’, ‘*’, ‘/’. For ‘+’ case perform addition and store result in some variable, i.e., res = num1 + num2. Similar things happen for the other 3 cases. If none of the above cases matches we print “invalid operator”. Finally res is printed.



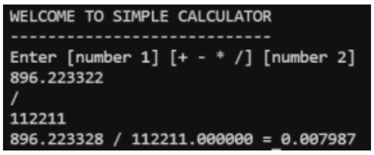
```
WELCOME TO SIMPLE CALCULATOR
----------------------------
Enter [number 1] [+ - * /] [number 2]
123.223322
+
1.776678
123.223320 + 1.776678 = 125.000000
```



```
WELCOME TO SIMPLE CALCULATOR
----------------------------
Enter [number 1] [+ - * /] [number 2]
437.86422
*
12.4231
437.864227 * 12.423100 = 5439.631348
```



```
WELCOME TO SIMPLE CALCULATOR
----------------------------
Enter [number 1] [+ - * /] [number 2]
12.23422
-
9.77282
12.234220 - 9.772820 = 2.461399
```



```
WELCOME TO SIMPLE CALCULATOR
----------------------------
Enter [number 1] [+ - * /] [number 2]
896.223322
/
112211
896.223328 / 112211.000000 = 0.007987
```

Some important points about the switch statement:

1. The break statement terminates the sequence of flow and ends the switch case. If this break statement is absent then the default case will happen always.
2. Duplicate case values are not allowed.
3. The default statement is optional. We can omit it if we want.