# ABES Engineering College, Ghaziabad.

**Affiliated to Dr. A.P.J Abdul Kalam Technical University, Lucknow.**

**Department of CSE-Data Science**

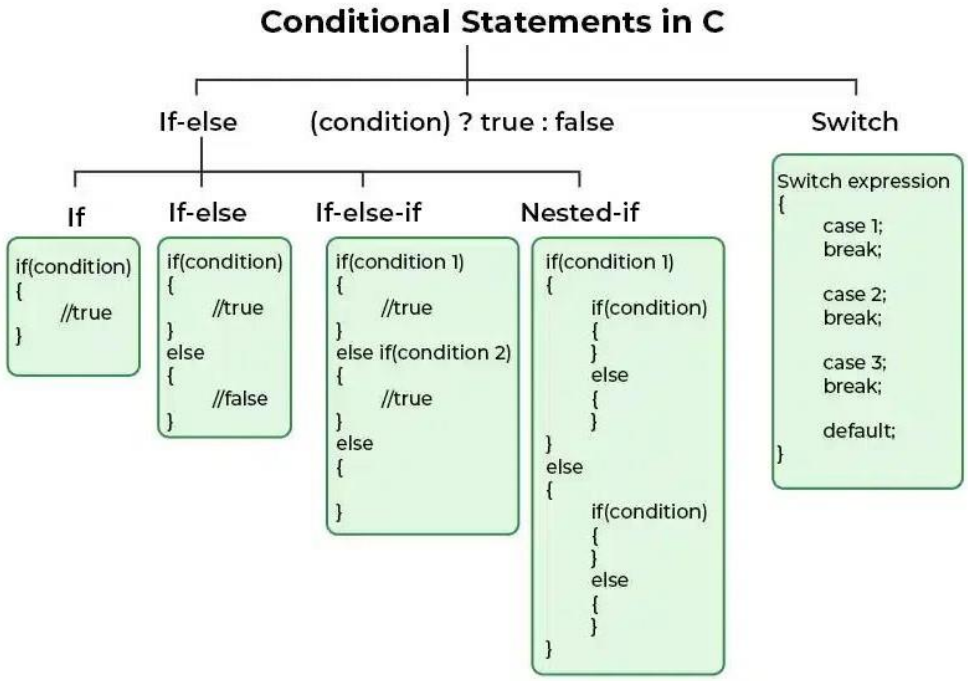
| | |
|---|---|
| **Title** | **Lecture Notes** |
| **Subject** | **Programming for Problem Solving** |
| **Topics** | **If else and switch** |
| **Lecture Date** | **May 2024** |
| **Faculty Name** | **Mr. Dilip Kr. Bharti** |
| **Section** | **First Year Section-B** |



## Decision Making in C (if , if..else, Nested if, if-else-if )

The conditional statements (also known as decision control structures) such as if, if else, switch, etc. are used for decision-making purposes in C programs. They are also known as Decision-Making Statements and are used to evaluate one or more conditions and make the decision whether to execute a set of statements or not. These decision- making statements in programming languages decide the direction of the flow of program execution.

## Types of Conditional Statements in C




Following are the **decision-making statements** available in C:

1. if Statement
2. if-else Statement

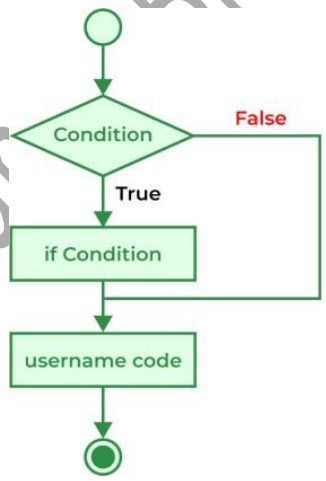
3. Nested if Statement
4. if-else-if Ladder
5. switch Statement
6. Conditional Operator
7. Jump Statements:
   - Break
   - Continue
   - Goto

**1. if in C**

The if statement is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not, it means if a certain condition is true then a block of statements is executed otherwise not.

**Syntax:**

```
If(condition)
{
}
```




**Example of if in C**

```
// C program to illustrate If statement
#include <stdio.h>int
  main()
{
    int i = 10; if
     (i > 15)
 {
```

```
        printf("10 is greater than 15");
    }
     printf("I am Not in if");
}
```

**Output:**

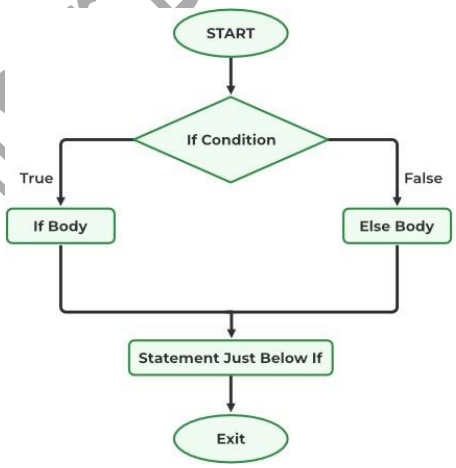
```
I am Not in if
```

### 2. if-else in C

If the given condition is true, then the code inside the if block is executed, otherwisethe code inside the else block is executed.

**Syntax:**

```
If(condition)
{
}
else
{
}
```






**//Example of if-else in C**

```
// C program to illustrate If-else statement
#include <stdio.h>int
 main()
{
        int a=10;
       if (a%2==0)
```

```
{
        printf("a is even");
```

```
    }
     else {
         printf("a is odd");
    }

    return 0;
}
```

**Output:**

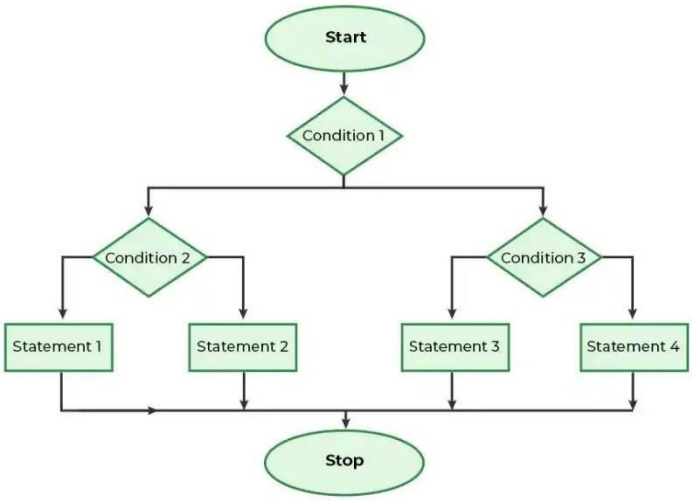
a is even

### 3. Nested if-else

Nested if statements mean an if statement inside another if statement.

**Syntax:**

```
if (condition1)
{
   // Executes when condition1 is trueif
   (condition2)
   {
      // Executes when condition2 is true
   }
   else
   {
        // Executes when condition2 is false
}
```




// **Example of Nested if-else in C**

## // Find the given year is leap or not

```
#include<stdio.h>int
main()
{
  int y;
  printf("enter year");
  scanf("%d",&y);
  if (y % 4 == 0)
{
  if (y % 100 == 0)
  {
    if (y % 400 == 0)
    {
      printf("%d is a leap year",y);
    } else
    {
      printf("%d is not a leap year",y);
    }
  } else
```

```
    {
     //if year is divisible by 4 but not 100, it is a leap year
     printf("%d is a leap year",y);
    }
  }
 else
     {
   printf("%d is not a leap year",y);
     }
return 0;
}
```

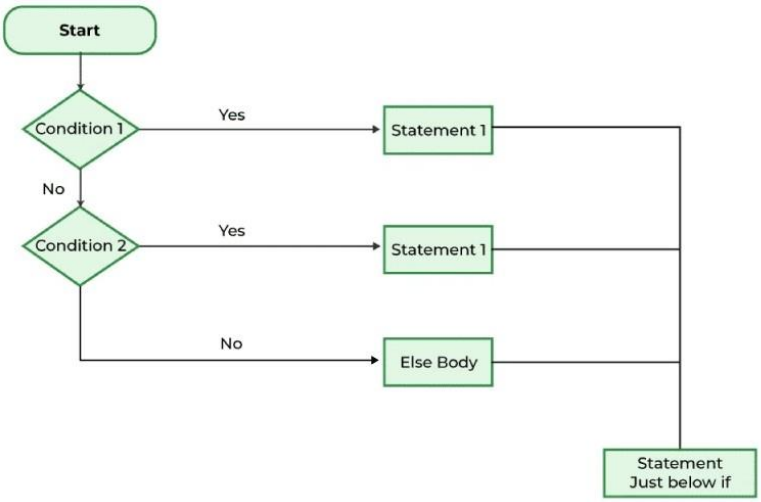
**4. if-else-if Ladder**

The if else if statements are used when the user has to decide among multiple options. The C if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. **if-else-if ladder is similar to the switch statement.**

**Syntax:**

```
if (condition)
  statement;
else if (condition)
 statement;
..
else
   statement;
```

**Flowchart:**




## //Example of if -else if ladder in C

**// number is positive, negative or not**

```
#include <stdio.h>int
  main()
{
    int A;
      printf("Enter the number A: ");
    scanf("%d", &A);
      if (A > 0)
        printf("%d is positive.", A);else
    if (A < 0)
        printf("%d is negative.", A);else
    if (A == 0)
        printf("%d is zero.", A);
      return 0;
}
```

**5. Switch Statement**

The switch case statement is an alternative to the if else if ladder that can be usedto execute the conditional code based on the value of the variable specified in the

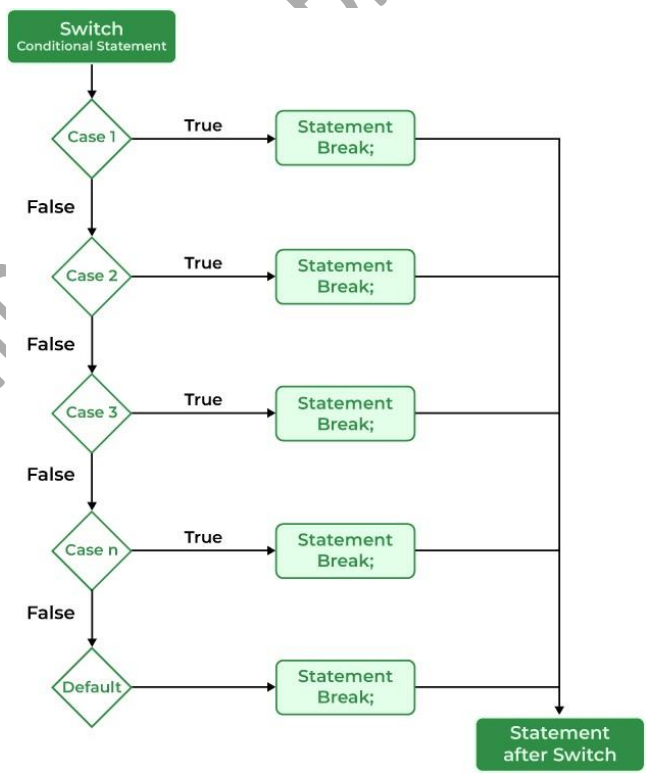
switch statement. The switch block consists of cases to be executed based on thevalue of the switch variable.

Syntax

```
switch (expression) {
    case value1:
        statements;
    case value2:
        statements;
    ....
    ....
    ....
    default:
        statements;
}
```

**Note: The switch expression should evaluate to either integer or character. Itcannot evaluate any other data type.**

## Flowchart:



**//Example of switch**

## // whether an alphabet is a vowel or not using switch

```
#include <stdio.h>int
main()
{
    char ch;
    printf("Enter any alphabet: ");
    scanf("%c", &ch);
    switch(ch)
    {
        case 'a':
            printf("Vowel");
            break;
        case 'e':
            printf("Vowel");
            break;
        case 'i':
            printf("Vowel");
            break;
        case 'o':
            printf("Vowel");
            break;
        case 'u':
            printf("Vowel");
            break;
        case 'A':
            printf("Vowel");
            break;
        case 'E':
```

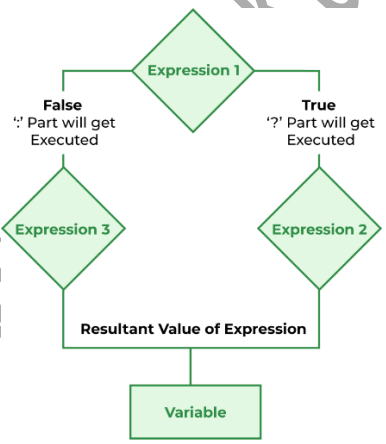
```
            printf("Vowel");
            break;
        case 'I':
            printf("Vowel");
            break;
        case 'O':
            printf("Vowel");
            break;
        case 'U':
            printf("Vowel");
            break;
        default:
            printf("Consonant");
    }
    return 0;
}
```

### 6. Conditional Operator

The conditional operator is used to add conditional code in our program. It is similarto the if-else statement. It is also known as the ternary operator as it works on threeoperands.

**Syntax:**

(condition) ? [true_statements] : [false_statements];



**//Example on conditional operator**

## //greater number among 2 numbers using conditional operator

```
#include<stdio.h>int
 main()
{
  int a,b;
  printf("enter 2 numbers");
  scanf("%d%d",&a,&b);
  (a>b)? printf("a is greater") :printf("b is greater");
}
```

### 7. Jump Statements

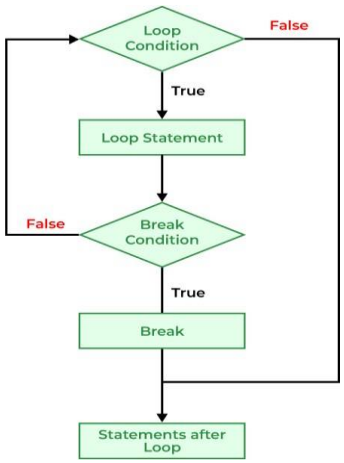
These statements are used in C for the unconditional flow of control throughout thefunctions in a program.

**break**

This loop control statement is used to terminate the loop. As soon as the break statement is encountered from within a loop, the loop iterations stop there, and controlreturns from the loop immediately to the first statement after the loop.

Syntax:

break;

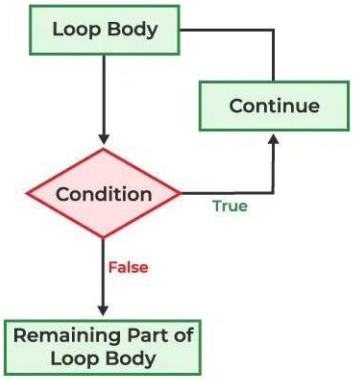



**continue**

When the continue statement is executed in the loop, the code inside the loop followingthe continue statement will be skipped and the next iteration of the loop will begin.

**Syntax:**

continue;




**goto**

The goto statement in C also referred to as the unconditional jump statement canbe used to jump from one point to another within a function.

**Syntax:**

Syntax1 | Syntax2

- - - - - - - - - - - - - - - - - -

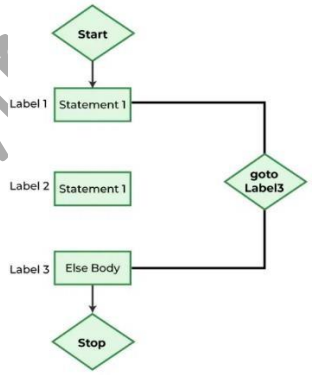
goto label; | label:

. | .

. | .

. | .

label: | goto label;



## // Programs on if-else

### Write a program to find out the greatest number out of three numbers

```
#include <stdio.h>int
main()
{
    int A, B, C;
     printf("Enter the numbers A, B and C: ");
    scanf("%d %d %d", &A, &B, &C);
     // finding max using compound expressionsif (A
    >= B && A >= C)
        printf("%d is the largest number.", A);else if
     (B >= A && B >= C)
        printf("%d is the largest number.", B);else
        printf("%d is the largest number.", C);return
     0;
}
```

**/*If three sides of triangle are input through keyboard, draw a flowchart to check whether a triangle is isosceles, equilateral scalene or right-angled triangle. Also write a program in C for the same**. */

```
#include<stdio.h>int
main()
{
   int side1, side2, side3; printf("Enter
   sides of triangle:");
   scanf("%d%d%d",&side1,&side2,&side3);if(side1 ==
   side2 && side2 == side3)
      printf("The Given Triangle is equilateral");
   else if (side1 == side2 || side2 == side3 || side3 == side1)
```

```
        printf("The given Triangle is isosceles");
    else    if((side1*side1)    +    (side2*side2)    ==    (side3*side3)    ||  (side1*side1)    +
 (side3*side3) == (side2*side2) || (side2*side2) + (side3*side3) == (side1*side1))
          printf("It is a right angle triangle!\n");else
        printf("The given Triangle is scalene");return
    0;
 }
```

**/*A Certain grade of steel is graded according to the following conditions:Hardness must be greater than 50.**

**Carbon content must be less than 0.7**

**Tensile strength must be less than 5600**

**The grades are as follows:**

**Grade is 10 if all the three conditions are met.Grade is 9 if condition (i) and (ii) are met.**

**Grade is 8 if condition (ii) and (iii) are met.**

**Grade is 7 if condition (i) and (iii) are met.**

**Grade is 6 if only one condition is met.**

**Grade is 5 if none of the conditions are met.**

**Write a program, which will require the user to give values of hardness, carboncontent and tensile strength of the steel under consideration and output the grade of the steel. */**

```
#include<stdio.h>int
main(){
   int h,t;
   double c;
   printf("Enter the value of hardness");
   scanf("%d",&h);
   printf("Enter the value of carbon");
   scanf("%lf",&c);
   printf("Enter the value of tensile");
```

```
    scanf("%d",&t);
    if(h>50 &&c<0.7 && t<5600)
        printf("Grade 10");
    else if(h>50 &&c<0.7)
        printf("Grade 9");
     else if(c<0.7 && t<5600)
        printf("Grade 8");
     else if(h>50 && t<5600)
        printf("Grade 7");
     else if(h>50 ||c<0.7 || t<5600)
        printf("Grade 6");
     else
            printf("Grade 5");
     return 0;
}
```

**Note: Use the double data type for finding the value of carbon content. If youuse float data type, you will get a wrong answer.**

**/*Write a program in C to print grades as per following criteria for obtained percentage of marks M out of 100:**

| Obtained Percent Marks (M) | Grade |
|---|---|
| 90 <M≤ 100 | A+ |
| 80 <M≤ 90 | A |
| 70 <M≤ 80 | B+ |
| 60 <M≤ 70 | B |
| 50 <M≤ 60 | C |
| M≤ 50 | F |

```
*/ #include<stdio.h>
    int main()
```

```
{
        int marks;
        float M;
printf("Enter marks not more than 100");
scanf("%d", &marks);
if(marks>100)
        printf("Wrong marks entered");
else
{
        M=(marks/100)*100;
if(M>90 && M<=100)
        printf("A+");
else if(M>80 && M<=90)
        printf("A");
else if(M>70 && M<=80)
        printf("B+");
else if(M>60 && M<=70)
        printf("B");
else if(M>50 && M<=60)
        printf("C");
else
        printf("F");
}
return 0;
}
```

## Switch Statement in C

# Introduction

The switch case statement is a flow control statement in which we can define a switch variable and then execute different code based on the value of the switch variable. **It isan alternative of if else if ladder.**

Switch case statement evaluates a given expression and based on the evaluated value (matching a certain condition), it executes the statements associated with it. Basically, it is used to perform different actions based on different conditions(cases).

**The syntax of switch statement in c language is given below:**

```
switch(expression)
{
case value1:
 //code to be executed;
 break;  //optional

case value2:
 //code to be executed;
 break;  //optional
 ......

default:
 code to be executed if all cases are not matched;
}
```

**Rules for switch statement in C language**

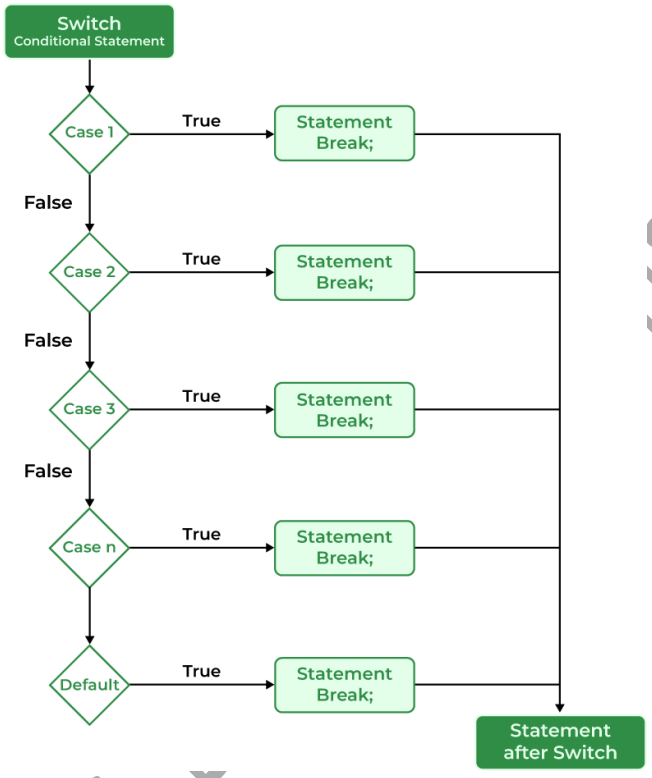
1. The **switch expression** must be of an **integer or character type.**
2. The **case value** must be an **integer or character constant.**
3. The case value can be used only inside the switch statement.
4. The break statement in switch case is not must. It is optional. **If there is no break statement found in the case, all the cases will be executed present after the matched case.** It is known as fall through the state of C switch statement.

Example:

```
int   x,y,z;
char  a,b;
float f;
```

| Valid Switch | Invalid Switch | Valid Case | Invalid Case |
|---|---|---|---|
| switch(x) | switch(f) | case 3; | case 2.5; |
| switch(x>y) | switch(x+2.5) | case 'a'; | case x; |
| switch(a+b-2) | | case 1+2; | case x+2; |
| switch(func(x,y)) | | case 'x'>'y'; | case 1,2,3; |

**Flowchart of switch statement in C**




## How switch Statement Work?

The working of the switch statement in C is as follows:

**Step 1:** The switch variable is evaluated.
**Step 2:** The evaluated value is matched against all the present cases.
**Step 3A:** If the matching case value is found, the associated code is executed.
**Step 3B**: If the matching code is not found, then the default case is executed if present.**Step 4A:** If the break keyword is present in the case, then program control breaks out of the switch statement.

**Step 4B:** If the break keyword is not present, then all the cases after the matching caseare executed.
**Step 5:** Statements after the switch statement are executed.

## case in switch case

The case keyword is used to define the different cases and their associated code in theswitch statement.

### break in switch case

**This keyword is used to stop the execution inside a switch block.** It helps to terminate the switch block and break out of it. **When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.**

**The break statement is optional.** If omitted, execution will continue on into the nextcase. The flow of control will fall through to subsequent cases until a break is reached.

**Example of switch case without break**

### // C Program of switch case without break

```
#include <stdio.h>int
main()
{
    int var = 2;
    // switch case without break
    switch (var)
 {
      case 1:
          printf("Case 1 is executed.\n");case
      2:
          printf("Case 2 is executed.\n");case
      3:
          printf("Case 3 is executed.");case
       4:
          printf("Case 4 is executed.");
    }
    return 0;
```

}

**Output:**

Case 2 is executed.
Case 3 is executed. Case 4 is executed.

**default in switch case**

The default keyword is used to specify the set of statements to execute if there is nocase match. It is optional to use the default keyword in a switch case. Even if the switch casestatement does not have a default statement, it would run without any problem.

**Important Points About Switch Case Statements**

1. Switch expression should result in a constant value.
2. Expression value should be only of int or char type.
3. Case Values must be Unique.
4. Nesting of switch Statements is allowed.
5. The default block can be placed anywhere(beginning, middle or end) in switchstatement.

**// C program to print the day using switch**

```
#include <stdio.h>int
main()
{
        int day = 2;
        printf("The day with number %d is ", day);switch
 (day) {
   case 1:
        printf("Monday");
        break;
 case 2:
        printf("Tuesday");
        break;
case 3:
        printf("Wednesday");
        break;
```

```
        case 4:
                printf("Thursday");
                break;
        case 5:
                printf("Friday");
                break;
        case 6:
                printf("Saturday");break;
        case 7:
                printf("Sunday");
                break;
        default:
                printf("Invalid Input");
                break;
        }
                return 0;
    }
```

**Output:**

The day with number 2 is Tuesday

**//Simple Calculator using switch case in C**

## // C Program to create a simple calculator using switch statement

```
#include<stdio.h>
#include<conio.h> int
main()
{
int x,y,z;
char choc;
printf(" \n Press\n a for additon\n s for subtraction\n m for multiplication\n d fordivison\n r for remainder");
printf("\nenter your choice\n");
scanf("%c",&choc); switch(choc)
{
case 'a':
```

```
        printf("enter two numbers\n");
        scanf("%d%d",&x,&y); z=x+y;
        printf("addition of two numbers is \t%d",z);break;
case 's':
        printf("enter two numbers\n");
        scanf("%d%d",&x,&y);
        z=x-y;
        printf("subtraction of two numbers is \t%d",z);break;
case 'm':
        printf("enter two numbers\n");
        scanf("%d%d",&x,&y); z=x*y;
        printf("Multipication of two numbers is \t%d",z);break;
case 'd':
        printf("enter two numbers\n");
        scanf("%d%d",&x,&y); z=x/y;
        printf("division of two numbers is \t%d",z);break;
case 'r':
        printf("enter two numbers\n");
        scanf("%d%d",&x,&y); z=x%y;
        printf("remainder of two numbers is \t%d",z);break;
default:
        printf("illegal choice");
}
return 0;
}
```

**Output:**

```
Press
 a for additon
```

s for subtraction
m for multiplicationd
for divison
r for remainder
enter your choicer
enter two numbers5
4
remainder of two numbers is 1

**Example 2:**

**/* program on calculate value of y using Switch.if N=1, y=ax%b**

**if N=2, y=axx+bbif N=3 y=-bx**

**if N=4 y=-a+x/b**

**enter the value of N,a,x,b1 2 3 4**

**Output is value of y is 2.000000 */**

```
#include<stdio.h>
#include<conio.h> int
main()
{
int N,x,a,b;
float y;
printf(" \n find value of y\n if N=1, y=ax%b\n if N=2, y=axx+bb\n if N=3 y=-bx\n ifN=4 y=-a+x/b");
printf("\nenter your choice\n");
scanf("%d",&N);
switch(N)
{
case 1:
        printf("enter the value of x,a,b\n");
        scanf("%d%d%d",&x,&a,&b); y=a*x%b;
        printf("value of y is \t%f",y);
        break;
```

```
case 2:
        printf("enter the value of x,a,b\n");
        scanf("%d%d%d",&x,&a,&b);
        y=a*x*x+b*b;
        printf("value of y is \t%f",y);
        break;
case 3:
        printf("enter the value of x,a,b\n");
        scanf("%d%d%d",&x,&a,&b); y=(-
        1)*b*x;
        printf("value of y is \t%f",y);
        break;
case 4:
        printf("enter the value of x,a,b\n");
        scanf("%d%d%d",&x,&a,&b); y=a+x/b;
        printf("value of y is \t%f",y);
        break;
default:
        printf("wrong choice");
}
return 0;
}
```

**Advantages of C switch Statement**

1. Easier to read than if else if.
2. Easier to debug and maintain for a large number of conditions.
3. Faster execution speed.
4. Ideal for menu-driven programming and multiple-choice scenarios.
5. Switch statements allow for a default case to handle situations where none of thespecified cases match the given value.

**Disadvantages of C switch Statement**

1. Switch case can only evaluate int or char type.
2. **No support for logical expressions.**
3. Have to keep in mind to add a break in every case.
4. A variable or expression can't be used as a case constant

| **Switch** | **if else if** |
|---|---|
| It executes the different cases on the basis of the value of the switch variable. | It executes the different blocks based on the condition specified. |
| It can only evaluate the int or char type expressions. | It can evaluate any type of expression. |
| Faster and easier to read for the large number of conditions. | It can get messy when there are lots of conditions. |
| It contains a single expression which can be either a character or integer variable. | it contains either logical or equality expression. |
| Cases in a switch statement are easy to maintain and modify. Therefore, we can say that the removal or editing of any case will not interrupt the execution of other cases. | Editing is not easy in the 'if-else'statement. |
| If we have multiple choices then the switch statement is the best option as the speed of the execution will be much higher than 'if-else'. | If there are multiple choices implemented through 'if-else', then the speed of the execution will be slow. |
| If the value does not match with any case,then by default, default statement is executed. | If the condition is not true, then by default, else block will be executed. |
| we need to use a single statement for numerous decisions. | we need to use multiple statements for numerous decisions. |
| Here, each case will be executed one after the other, if the break is not used after ending of each case. | One statement will be executed. It can be if or else. |
| //Example of switch | // Example of if else if |

**Different types of control statement in C**

## Types of Control Statements in C

The primary types of control statements in C are:

**Sequence Statements**

## Decision-making control statements

1. Simple if statement
2. If-else statements
3. Nested if-else statements
4. else-if ladder

## Jump Statements

break

goto

continue

**Loop control statements in C**

1. While Loop
2. Do-while Loop
3. For Loop

**Previous year Question Papers2017-18**

**(RCS-101)**

1. Draw a flow chart to find the greatest number among three numbers. 2

**2017-18 (RCS-201)**

**NIL**

**2018-19(KCS-101)**

1. A Certain grade of steel is graded according to the following conditions: Hardness must be greater than 50.
   Carbon content must be less than 0.7 Tensile
   strength must be less than 5600

The grades are as follows:
Grade is 10 if all the three conditions are met.Grade
is 9 if condition (i) and (ii) are met.
Grade is 8 if condition (ii) and (iii) are met.Grade
is 7 if condition (i) and (iii) are met. Grade is 6 if
only one condition is met.
Grade is 5 if none of the conditions are met.
Write a program, which will require the user to give values of hardness, carboncontent and tensile strength of the steel under consideration and output the grade of the steel. 10

2. What is case control structure in C? What is the reason for using break statementat the end of each case in case control block? 10

**2018-19(KCS-201)**

1. Correlate else if ladder and switch case statement. 2
2. Explain the use of default in switch statement. Write a program that takes two operands and one operator from the user and perform the operation and print theresult by using switch statement. 10

**2019-20(KCS-101)**

1. What are different conditional statements in C programming? Explain withproper example of each. 10
2. If three sides of triangle are input through keyboard, draw a flowchart to check whether a triangle is isosceles, equilateral scalene or right-angled triangle. Alsowrite a program in C for the same. 10

**2020-21(KCS-101T)**

1. Explain the need of break in switch statement with example. 2
2. Write a program to find out the greatest number out of three numbers. 10
3. Explain different type of control statements used in C programming with example. 10

**2021-22(KCS-101T)**

1. Write limitations of switch case. 2

2. Compare if-else-if ladder and switch case. Write a menu driven program to perform basic functions of calculator. 10

**2021-22(KCS-201T)**

1. Write advantages of Switch statement. 2
2. What is use of break in switch case? Write a program to develop a calculator using case in character format. 10

**2022-23(BCS-101)NIL**

**2022-23(BCS-201)**

1 Write the output of following

code:#include <stdio.h>

```
int main()
{
 int a = -10, b = 20;if(a
 > 0 && b < 0)a++;
 else if(a < 0 && b < 0)a--;
 else if(a < 0 && b > 0)b--;
else
 b--;
 printf("%d\n",a + b);
 return 0;
 }
```

2

2 Write a program in C to print grades as per following criteria for obtained

percentage of marks M out of 100:

| Obtained Percent Marks (M) | Grade |
|---|---|
| 90 <M≤ 100 | A+ |
| 80 <M≤ 90 | A |
| 70 <M≤ 80 | B+ |
| 60 <M≤ 70 | B |
| 50 <M≤ 60 | C |
| M≤ 50 | F |

7

**2023-24(BCS-101)**

1 Write a Program to discuss the use of break in Switch Statement. 7