- Switch-statement in C:-

→ Syntax of switch statement in C:-

```
Switch (expression)
{
case value1: statement_1;
        break;
case value2: statement_2;
        break;
  .
  .
  .
case value_n: statement_n;
        break;
default: default_statement;
}
```

* In a switch statement, the "case value" must be of "char" and "int" type.
* There can be one or N number of cases.
* The values in the case must be unique.
* Each statement of the case can have a break statement. It is optional.
* The default statement is also optional.

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

```
//switch variable
int var=1;

// switch statement
switch (var) int-
{
 case 1:
    printf("Case 1 is Matched.");
    break;
 case 2:
    printf ("case 2 is matched:");
    break;
 case 3:
    printf (" Case 3 is matched.");
    break;
 default:
    printf (" Default case is matched.");
    break;
 }
 return 0;
 }
```

Output:-

Case 1 is Matched

* If the break keyword is not present, then all the cases after the matching case are executed.

```
/* C program to demostrate the behaviour 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.

→ Important points about switch case statements:-

* Switch expression should result in a constant value.
* Expression value should be only of int or char type.
* Case values must be unique.
* Nesting of switch statements.
  ⇒ Nesting is allowed.
  ⇒ should be avoided – complex and less readable.
* The default block can be placed anywhere.

Ex.-1 // C program to print the day using switch

```
#include<stdio.h>
//Driver code
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.

ex.-2

```
/* C program to create a simple calculator
   using switch */
// statement
```

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

//driver code.
int main()
{
  //switch variable
  char choice;
  //operands
  int x, y;
  while(1)
  {
    printf("Enther the operator (+,-,*,/) \n
        Enter x to exit \n");
    scanf("%c", &choice);
    //for exit
    if (choice=='x')
    {
      exit(0);
    }
    printf("Enter the two numbers:");
    scanf("%d %d", &x, &y);
    //switch case with operation with each operator
    switch(choice)
    {
    case'+':
```

```
        printf("%d + %d = %d \n", x, y, x+y);
        break;
    case '-':
        printf("%d - %d = %d \n", x, y, x-y);
        break;
    case '*':
        printf("%d * %d = %d \n", x, y, x*y);
        break;
    case '/':
        printf("%d / %d = %d \n", x, y, x/y);
        break;
    default:
        printf("Invalid operator Input \n");
    }
}
return 0;
}
```

Output:-

```
Enter the operator (+, -, *, /)
Enter x to exit
+
Enter the two numbers : 100 200
100 + 200 = 300
```

* Switch case, easier to read and debug, faster execution, maintain for a large number of condition.

* Disadvantage :-

- can only evaluate int or char type.
- No support for logical expressions.
- Have to keep in mind to add a break in every case