

# ਆਈ.ਕੇ. ਗੁਜਰਾਲ ਪੰਜਾਬ ਟੈਕਨੀਕਲ ਯੂਨੀਵਰਸਿਟੀ (ਅਕਾਦਮਿਕ ਵਿਭਾਗ)



Ref. No.: IKG-PTU/DA/ 18002

Date : 18/10/2018

ਡਾਇਰੈਕਟਰ / ਪ੍ਰਿੰਸੀਪਲ / ਫੈਕਲਟੀ ਇੰਚਾਰਜ  
ਸਮੂਹ ਖੇਤਰੀ ਕੈਂਪਸ ਅਤੇ ਐਫੀਲੇਡਿਟਡ ਕਾਲਜ  
ਆਈ.ਕੇ.ਗੁਜਰਾਲ ਪੰਜਾਬ ਟੈਕਨੀਕਲ ਯੂਨੀਵਰਸਿਟੀ।

ਵਿਸ਼ਾ :- ਬੀ.ਟੈਕ. ਸਮੈਸਟਰ ਪਹਿਲਾ ਅਤੇ ਦੂਜਾ ਬੈਚ 2018 Paper Code BTPS101-18 ਦੇ ਮਾਡਲ ਪ੍ਰਸ਼ਨ ਪੱਤਰ ਸਮੇਤ ਉੱਤਰ ਕੁੰਜੀਆਂ ਭੇਜਣ ਸਬੰਧੀ।

ਉਪਰੋਕਤ ਵਿਸ਼ੇ ਦੇ ਸਬੰਧ ਵਿੱਚ ਆਪ ਜੀ ਨੂੰ ਬੀ.ਟੈਕ. ਸਮੈਸਟਰ ਪਹਿਲਾ ਅਤੇ ਦੂਜਾ ਬੈਚ 2018 Paper Code BTPS101-18 ਦੇ ਮਾਡਲ ਪ੍ਰਸ਼ਨ ਪੱਤਰ ਸਮੇਤ ਉੱਤਰ ਕੁੰਜੀਆਂ ਇਸ ਪੱਤਰ ਨਾਲ ਨੱਥੀ ਕਰਕੇ ਭੇਜੀਆਂ ਜਾਂਦੀਆਂ ਹਨ। ਆਪ ਜੀ ਨੂੰ ਬੇਨਤੀ ਕੀਤੀ ਜਾਂਦੀ ਹੈ ਕਿ ਇਹ ਮਾਡਲ ਪ੍ਰਸ਼ਨ ਪੱਤਰ ਸਮੇਤ ਉੱਤਰ ਕੁੰਜੀਆਂ ਸਬੰਧਤ ਵਿਦਿਆਰਥੀਆਂ ਅਤੇ ਅਧਿਆਪਕਾਂ ਤੱਕ ਪੁੱਜਦੇ ਕੀਤੇ ਜਾਣ ਤਾਂ ਜੋ ਸਬੰਧਤ ਵਿਦਿਆਰਥੀ ਇਮਿਤਹਾਨਾਂ ਦੀ ਤਿਆਰੀ ਸੁਚੱਜੇ ਢੰਗ ਨਾਲ ਕਰ ਸਕਣ। ਇਸ ਸਬੰਧੀ ਕਿਸੇ ਵੀ ਪ੍ਰਕਾਰ ਦੇ ਸੁਝਾਅ / ਫੀਡਬੈਕ ਯੂਨੀਵਰਸਿਟੀ ਦੇ ਅਕਾਦਮਿਕ ਵਿਭਾਗ ਨੂੰ ਭੇਜੇ ਜਾ ਸਕਦੇ ਹਨ।

ਨੱਥੀ : ਉਕਤ ਅਨੁਸਾਰ।

(ਰਜਿੰਦਰ ਡੋਗਰਾ)  
ਡਿਪਟੀ ਰਜਿਸਟਰਾਰ  
18/10/2018

ਇਸ ਦਾ ਇੱਕ ਉਤਾਰਾ :

1. ਇੰਚਾਰਜ ਸਕੱਤਰੇਤ ਉਪ ਕੁਲਪਤੀ ਜੀ ਨੂੰ ਸੂਚਨਾ ਹਿੱਤ।
2. ਕੰਟਰੋਲਰ ਪ੍ਰੀਖਿਆਵਾਂ ਜੀ ਨੂੰ ਬੀ.ਟੈਕ. ਸਮੈਸਟਰ ਪਹਿਲਾ ਅਤੇ ਦੂਜਾ ਬੈਚ 2018 ਦੇ ਮਾਡਲ ਪ੍ਰਸ਼ਨ ਪੱਤਰ ਸਮੇਤ ਉੱਤਰ ਕੁੰਜੀਆਂ ਇਸ ਪੱਤਰ ਨਾਲ ਨੱਥੀ ਕਰਕੇ ਸੂਚਨਾ ਅਤੇ ਲੋੜੀਂਦੀ ਕਾਰਵਾਈ ਹਿੱਤ ਭੇਜੇ ਜਾਂਦੇ ਹਨ ਜੀ।
3. ਡਿਪਟੀ ਕੰਟਰੋਲਰ, ਗੁਪਤ ਵਿਭਾਗ (Confidential Branch) ਜੀ ਨੂੰ ਸੂਚਨਾ ਅਤੇ ਲੋੜੀਂਦੀ ਕਾਰਵਾਈ ਹਿੱਤ।
4. ਡਿਪਟੀ ਕੰਟਰੋਲਰ, ਆਈ.ਟੀ.ਐਸ. ਵਿਭਾਗ ਨੂੰ ਯੂਨੀਵਰਸਿਟੀ ਵੈਬਸਾਈਟ "ਨੋਟਿਸ ਬੋਰਡ" ਉਪਰ ਅਪਲੋਡ ਕਰਨ ਹਿੱਤ ਭੇਜੇ ਜਾਂਦੇ ਹਨ ਜੀ।

(ਰਜਿੰਦਰ ਡੋਗਰਾ)  
ਡਿਪਟੀ ਰਜਿਸਟਰਾਰ  
18/10/2018

Roll No. \_\_\_\_\_

Total No. of Pages 01

Total No. of Questions: 09

**B.Tech. (Sem- 1&2) Dec, 2018**  
**Programming for Problem Solving**  
**Subject code: BTPS101-18**

Time: 3 hrs.

Max. Marks: 60

Instructions to Candidates:

1. Section A is compulsory consisting of ten questions carrying TWO marks each.
2. Section B & C have four questions each carrying EIGHT marks each
3. Attempt any five questions from Section B& C, selecting at least two from each section.

**SECTION-A**

**1. Write briefly:**

- a. What is an algorithm?
- b. List various input and output devices.
- c. What is the significance of operating system?
- d. Why switch statement is used?
- e. Write down the syntax of function declaration also give an example.
- f. Draw block diagram of a computer system.
- g. Differentiate syntax and logical errors.
- h. Compute the binary equivalent of 357.
- i. List jumping statements.
- j. Write down the syntax of else-if ladder.
- k. Differentiate RAM and ROM.

**SECTION-B**

2. List and explain string library functions in detail.
3. Write the following programs
  - a. To determine whether a number is palindrome.
  - b. To compute transpose of a matrix.
4. Differentiate call by value and call by reference with programming example.
5. Write a program to search an element from a list of numbers.

**SECTION-C**

6. What do you mean by recursion? Give programming illustration to represent the concept of recursion.
7. Write a program to sort a list of numbers using bubble sort method.
8. Explain the following operators with example
  - a. Relational
  - b. Conditional
  - c. Logical
  - d. Arithmetic
9. Explain the concept of array of structure with programming illustration.



## Section-A

Ans 1

Ans (a) An algorithm is a detailed series of instructions for carrying out an operation or solving a problem.

In non technical approach, we use algorithms in everyday tasks, such as a recipe to bake a cake or a do-it yourself handbook. Technically, computers use algorithms to list the detailed instructions for carrying out an operation.

Ans (b) Various Input devices are :-

1. Mouse
2. Keyboard
3. Joystick
4. Scanner
5. Microphone

Various output devices are :-

1. Monitor
2. Printer
3. Speaker
4. Projector
5. Head Phone

Ans (C) The operating system controls computer system resources and coordinates the flow of data to and from the micro processor. At the same time it also controls the flow of data between the input and output devices such as keyboard & monitor.  
It also acts as

1. Processor management
2. Memory management
3. Input output management
5. File management.

Ans (d) Use of switch statement

Switch statement is a control statement that allows us to choose only one choice among the many given choices. The expression in switch evaluates to return an integral value, which is then compared to the values present in different cases. It executes that block of code which matches the case value.

Ans (e) Syntax For Function Declaration

Return type FunctionName (Parameter types);

// main code

Return type FunctionName (function parameters)

{

// statements that execute when called

Return Value;

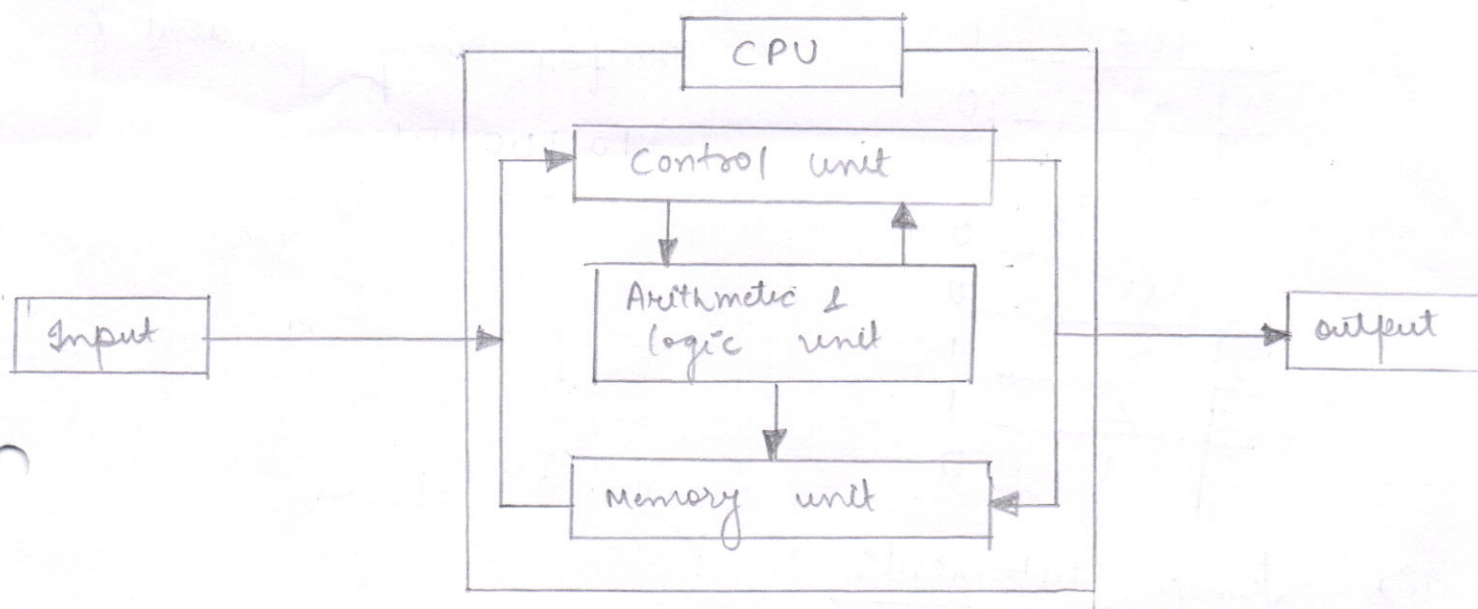
}



### example

```
int FindMaximum (int, int );  
Void Main ( )  
{  
    int MaxNumber = Find Maximum (5,7 );  
}  
int FindMaximum ( int number1, int number2  
{  
    int maximum = Number2;  
    if ( Number1 > Number2 )  
        maximum = Number1;  
    Return Maximum;  
}
```

### Ans(f) Block Diagram of Computer



### Ans(g)

#### Syntax Error

A syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in a Particular Programming language

#### Logical error

A logical error is an error in the Program that causes it to operate incorrectly but not to terminate abnormally.

- In compiled languages the compiler indicates the syntax error with the location and what the error is.
- It is easier to identify a syntax error
- A syntax error occurs due to fault in the syntax of the program

- The programmer has to detect the error by himself
- It is comparatively difficult to identify a logical error.
- It occurs due to fault in the algorithm.

Ans (h) Binary Equivalent of 357

|   |     |   |
|---|-----|---|
| 2 | 357 |   |
| 2 | 178 | 1 |
| 2 | 89  | 0 |
| 2 | 44  | 1 |
| 2 | 22  | 0 |
| 2 | 11  | 0 |
| 2 | 5   | 1 |
| 2 | 2   | 1 |
|   | 1   | 0 |

Therefore Binary Equivalent is

101100101

Ans (i) Jump statements

1. Goto Statement :- It unconditionally transfers program control to a labelled statement where the label identifier is in the scope of the function containing the goto statement.
2. Continue Statement - It passes control to the end of the immediately enclosing while, do, or for statement.



3. Break statement :- The break statement terminates execution of the immediately enclosing while, do, for, or switch statement.

4. The return statement :- It terminates execution of a function and returns control to the calling function with or without a return value.

Ans(j)      Syntax of If else ladder

```
if ( condition 1 )  
{  
    statement 1 ;  
}  
else if ( condition 2 )  
{  
    statement 2 ;  
}  
else if ( condition 3 )  
{  
    statement 3 ;  
}  
else  
{  
    statement 4 ;  
}  
}
```

Ans(k)

RAM

- RAM stands for Random Access Memory.
- It is volatile memory.
- RAM : DRAM & SRAM
- It requires flow of electricity to retain data

ROM

- ROM stands for Read only memory.
- It is Non volatile memory.
- ROM : PROM & EPROM
- It does not require electricity to retain data

- RAM is the memory available for the operating system Program and Process to use when the Computer is running

- ROM is memory that comes with your computer that is Pre-written to hold the instructions for booting up the computer



## SECTION - B

2. List and Explain library functions of String in Detail.

There are so many string related library functions which operate on the string and produces certain results. Some of the commonly used string library functions

- strlen(): This function returns the length of string

Example :- `strlen(name);`

This will return the length of string stored in variable `name[]`.

- strcat(): This function concatenates two strings.

Syntax :- `strcat(first string, second string);`

Concatenation will be done in the order in which they are written.

- strcpy(): This function copies the value of the second string to first string.

Syntax :- `strcpy(name1, name);`

This will copy the string stored in variable `name[]` to the variable `name1[]`.

- strcmp() :- It compares two strings.

`strcmp(name, name1);`

It will return 0, if Both the strings are same.

It return the value less than 0 if `name[] < name1[]`.  
otherwise it returns Value greater than 0.

- strlower() :- It changes all the Characters of the string to Lower Case

`strlower(name);`

It will convert the whole string stored in `name[]` to lowercase.

- strupr() :- It changes all the Characters of the string to upper case

`strupr(name);`



- strchr() :- It returns the location or the pointer of the first occurrence of a character in a string.  
`strchr(name, ch);`

It returns the location of the first occurrence of character in variable `ch` in the string `name[]`.

It returns NULL if character is not found.

- strstr() :- It returns the location or the pointer of the first occurrence of one string in another.  
`strstr(name, name1);`

Program for implementing these library function.

```
#include <iostream.h>
#include <string.h>
```

```
main()
{
    char *x;
    char name[50], name1[50], name2[50], ch = 'B';
    cout << "Enter two strings";
    cin >> name >> name1;
    cout << "length of name string is" << strlen(name);
    strcat(name, name1);
    cout << name;
    strcpy(name2, name);
    cout << name2;
    if (strcmp(name, name1))
    {
        cout << "Hello";
    }
    cout << strlen(name);
    cout << strcpy(name);
    return 0;
    getch();
}
```

output

```
Enter two strings
The bell rings bell
length of name string is 14
The ball rings bell
The ball rings
The ball rings
THE BALL RINGS
```



(3) Write a program to determine whether a number is palindrome.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{ int a, n, b, temp = 0;
```

```
clrscr();
```

```
Cout << "Enter any Number";
```

```
Cin >> n;
```

```
b = n;
```

```
while (n > 0)
```

```
{ a = n % 10;
```

```
temp = temp * 10 + a;
```

```
n = n / 10;
```

```
}
```

```
if (temp == b)
```

```
{ Cout << "Palindrome Number";
```

```
}
```

```
else
```

```
{ Cout << "Not Palindrome Number";
```

```
}
```

```
getch();
```

```
}
```

Output

Enter any Number 143

Not Palindrome Number.

(B) Write a program to compute transpose of matrix.

```
#include <iostream.h>
#include <conio.h>
main()
{
    clrscr();
    int arr[10][10], m, n, trans[10][10];

    Cout << "Enter the Rows and Columns ";
    Cin >> m >> n;

    Cout << "Enter the elements of matrix";
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<n; j++)
        {
            Cin >> arr[i][j];
        }
    }

    for (int i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
        {
            Cout << "\t" << arr[i][j];
        }
        Cout << "\n";
    }

    for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
        {
            transpose[j][i] = arr[i][j];
        }
    }

    Cout << "Transpose of Matrix is";
```



```

for(i=0; i<m; i++)
{
    for(j=0; j<n; j++)
    {
        cout<<"\t" << trans[i][j];
    }
    cout<<"\n";
}
getch();

```

3.

Output :- Enter the Rows and Columns  
                   3 2  
           Enter the elements of matrix

3  
4  
2  
1  
5  
7

3 4  
2 1  
5 7

// Matrix (Input Matrix)

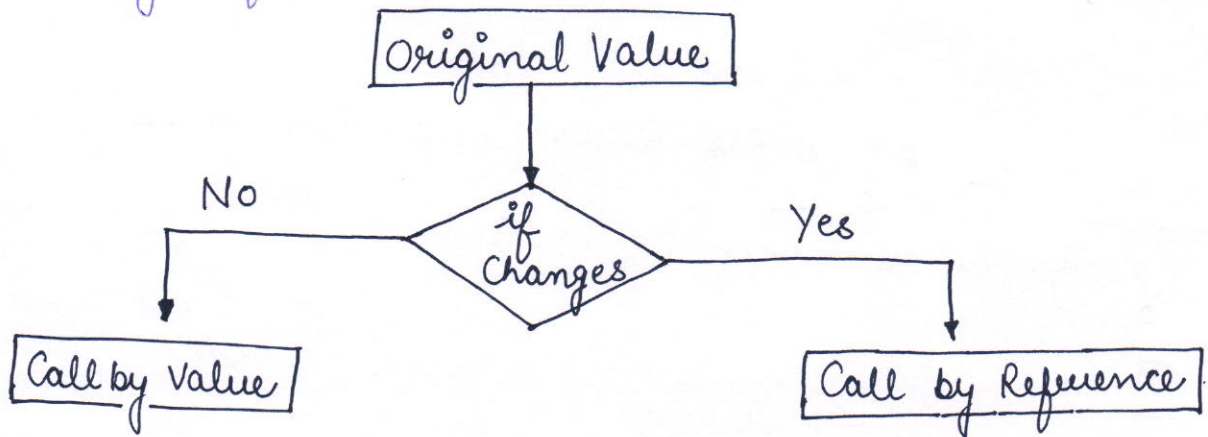
Transpose of Matrix is

3 2 5  
4 1 7

#### 4. Differentiate Call by Value and Call by reference with programming Example.

If a function take any arguments, it must declare variables that accept the values as a argument. These variables are called formal parameters of the function. There are two ways to pass value or data to function :-

- Call by value
- Call by Reference



Call by Value :- Original Value cannot be changed or modified.

Example :-

```
#include <iostream.h>
#include <conio.h>
void swap(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

```
void main()
{
    int a = 10, b = 20;
    clrscr();
    swap(a, b);
    cout << "a is" << a;
    cout << "b is" << b;
    getch();
}
```

Output :-

```
a is 20
b is 10
```

// passing value to the function



Call by Reference:- original value is changed or modified  
Here, address of the value is passed in function, so actual and formal arguments share the same address space  
Hence, any value changed inside the function, is reflected inside as well as outside the function.

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void swap (int *a, int *b)
```

```
{ int temp;
```

```
temp = *a;
```

```
*a = *b;
```

```
*b = temp;
```

```
}
```

```
void main()
```

```
{ clrscr();
```

```
int a=10, b=20;
```

```
swap(&a, &b); // passing address space to the function
```

```
cout << "a is " << a;
```

```
cout << "b is " << b;
```

```
getch();
```

```
}
```

Output

a is 20

b is 10

5. Write a program to search an Element from a list of number:

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main()
```

```
{ clrscr();
```

```
int arr[100], n, i, k
```

```
cout << "Total Elements ";
```

```
cin >> n;
```

```
cout << "Enter all the elements of list of numbers";
```

```
for(i=0; i<n; i++)
```

```
{ cin >> arr[i];
```

```
}
```

```
cout << "Enter the Number that you want to search";
```

```
cin >> k;
```

```
for(i=0; i<n; i++)
```

```
{ if (arr[i] == k)
```

```
{ cout << "Index of " << k << " is " << i;
```

```
break;
```

```
}
```

```
if (i == n)
```

```
{ cout << k << " is not present in the list";
```

```
}
```

```
getch();
```

```
}
```



Output :-

Total Elements 7

Enter all the Elements of list of numbers

1

2

3

5

70

15

23

Enter the number that you want to search 70

Index of 70 is 4.

## Section-C

6. Recursion in computer science is a method of solving a problem where the solution depends on solution to smaller instances of the same problem. It is a method of breaking down a problem into smaller and smaller subproblems until we reach a base condition. The general syntax of a recursive function :-

```
return_type func(arg1, arg2, ...)  
{  
    if (base cond == true)  
    {  
        ...  
    }  
    else  
    {  
        func(arg1, ... .) // recursive call  
    }  
}
```

Recursive functions are very useful to solve many problems like calculating the factorial of number, Fibonacci series etc.

Example: To compute factorial of given number using recursive function

```
#include <stdio.h>  
int factorial(int i)  
{  
    if (i <= 1) // base condition  
    {  
        return 1;  
    }  
    else  
    {  
        return i * factorial(i-1); // recursive call  
    }  
}
```



```

int main()
{
    int i = 12;
    printf("Factorial of %d is %d", i, factorial(i));
    return 0;
}

```

When the above code is compiled and executed, it produces the following result:-

Factorial of 12 is 479001600

7. Bubble sort is simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. The algorithm compares each pair of adjacent items and swaps them if they are in wrong order and this same process continues till no swap is needed.

```

#include <stdio.h>
int main()
{
    int array[100], n, c, d, swap;
    printf("Enter the number of elements");
    scanf("%d", &n);
    printf("Enter %d integers", n);
    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);
}

```

```

for(c=0; c<n-1; c++)
{
    for(d=0; d<n-c-1; d++)
    {
        if(array[d] > array[d+1])
        {
            swap = array[d];
            array[d] = array[d+1];
            array[d+1] = swap;
        }
    }
}

```

```

printf("Sorted list in ascending order:");

```

```

for(c=0; c<n; c++)
    printf("%d", array[c]);

```

```

return 0;

```

```

}

```

8. A operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C operators are symbols that are used to perform mathematical or logical manipulations. Operators take part in a program for manipulating data and variables and form a part of mathematical or logical expressions.

C offers various types of operators having different functioning capabilities:-

i. Arithmetic Operators

|    |                                                             |              |
|----|-------------------------------------------------------------|--------------|
| +  | Adds two operands                                           | $A+B = 30$   |
| -  | Subtracts second operand from first                         | $A-B = 10$   |
| *  | Multiply both operands                                      | $A*B = 200$  |
| /  | Divides numerator by denominator                            | $B/A = 2$    |
| %  | Modulus operator and remainder of after an integer division | $B \% A = 0$ |
| ++ | Increment operator increases the integer value by one       | $A++ = 11$   |
| -- | Decrement operator decreases the integer value by one       | $A-- = 9$    |

## ii. Relational Operator

They are used to compare two or more conditions / constraints or to complement the evaluation of original condition is considered.

|    |                          |
|----|--------------------------|
| == | is equal to              |
| != | is not equal to          |
| >  | Greater than             |
| <  | less than                |
| >= | Greater than or equal to |
| <= | less than or equal to    |

## iii. Logical Operator

They are used to combine two or more conditions / constraints.

|    |                      |
|----|----------------------|
| && | logical AND operator |
|    | logical OR operator  |
| !  | logical NOT operator |



#### iv. Conditional Operator

It is ternary operator used to construct conditional expressions  
cond. ? true statement : false statement

Q. A structure is a composite datatype with a collection of variables. These variables can have different data types and collectively form a structure of composite datatype. An array of structure is a sequential collection of structures. With structures, you can store mixed record types and with an array supporting this, we can have a list of mixed record types.

```
#include <stdio.h>
#include <string.h>
struct student
```

```
{
    int rollno;
    char name[10];
```

```
};
int main()
```

```
{
    int i;
    struct student st[3];
    printf("Enter records of 3 students");
    for (i=0; i<3; i++)
    {
        printf("Enter Roll No:");
        scanf("%d", &st[i].rollno);
        printf("Enter Name:");
        scanf("%s", &st[i].name);
```

```
}
```

```
printf("\n Student Information list");
```

```
for (i=0; i<3; i++)
```

```
{
```

```
    printf("\n Rollno : %d , Name : %s", st[i].rollno, st[i].name);
```

```
}
```

```
return 0;
```

```
}
```