



Q-1(A) Objective Type Questions

(4)

(1) C language was developed by _____.

Dennis M. Ritchie

(2) What is Variable?

Variable is a storage area that used to store different types of values.

(3) _____ is group of characters.

String

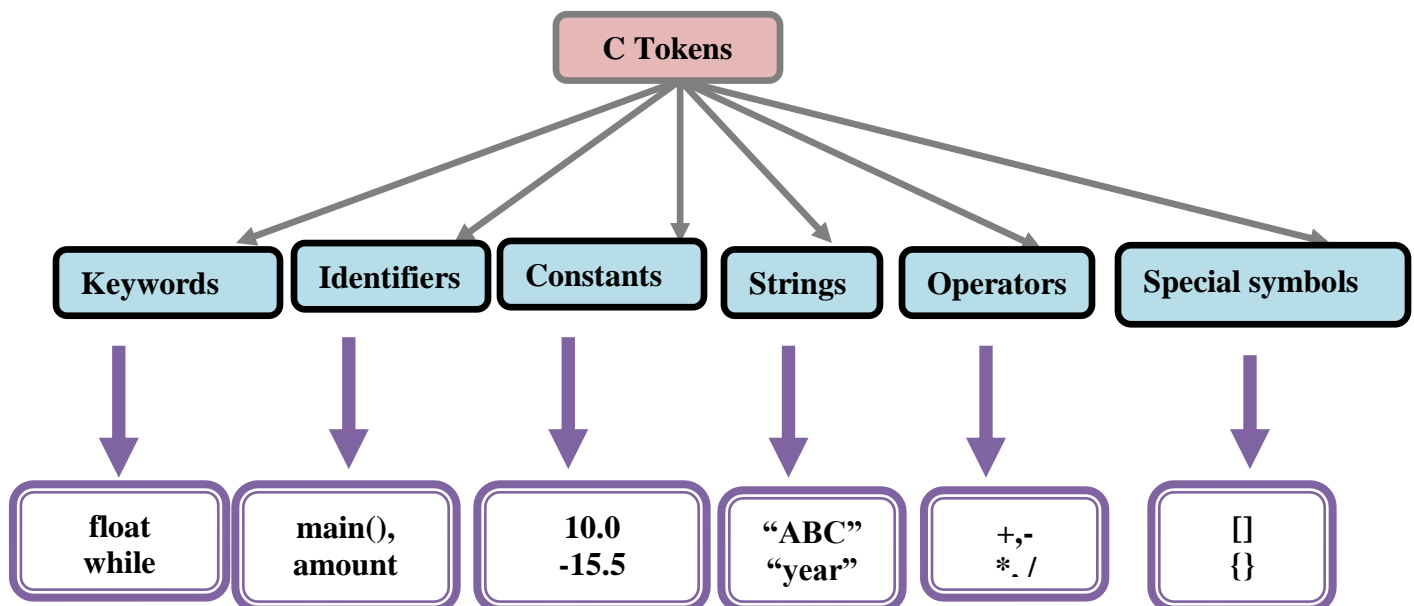
(4) What is Dry run?

A dry run is a process of manual compilation of the code to trace value of variable.

Q-1 (B) Answer in Brief .

(2)

(1) List out different tokens in C and explain any one in brief.





Detail :-

- ❖ In a C program, the smallest individual unit is known as C tokens.
- ❖ A token is a source program text which compiler does not break into further elements.
- ❖ These basic elements recognized by compilers are known as “tokens”.
- ❖ Tokens are further classified into keywords, constants, strings, operators and special symbols.

Strings :-

- ❖ In C language “bunch of characters “ is called string.
- ❖ In C language string can be represented by double quotes (“ ”).
- ❖ To indicate end of the string ‘\0’ (NULL) character can be used.
- ❖ C will automatically create an array of characters.
- ❖ Each character in the array will occupy 1 byte in the memory.
- ❖ In C language we can declare string value like following :-

Example: char string[]=”hello”.

h	e	l	l	o
---	---	---	---	---

2000 2001 2002 2003 2004

- ❖ While entering the string using scanf() following points should be kept in mind:
 - 1) The length of string can not be more than size of character array.
 - 2) The string can be fixed or dynamic.

Q-1 (C) Answer in Brief .

(3)

(1) Briefly explain structure of C program.

- ❖ The C program structure means “collection of functions and rules”.



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

❖ C program also support different sections which can be structure of the program.

***Sections:-**

- (1) Documentation Section :-This section is used to represent name of the program and comment line.
- (2) Link Section :- In this section we can support different header files that used for the program.
- (3) Definition section :- This section is used to define pre-processor directives.
- (4) Global section:- This section is used to declare global variables.
- (5) Main() :- It is called entry point of any C program.It is first function in every C program from where program execution will be start.
- (6) Sub – program :- This section contains user define functions.

*****STRUCTURE OF C PROGRAM *****

- Documentation Section
- Link Section
- Definition Section
- Global Declaration
- Main()
- {
 - Sub - Program section}

Example :-

```
//write a program to print "hello" on the screen.//documentation
#include<stdio.h> //link section
#include<conio.h>
```



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

```
Int y=20; //global section
Void main() // main section
{
    Int x=10; //sub – program section
    Printf("x=%d",x);
    Printf("y=%d",y);
    Getch();
}
```

Q-1 (D) Answer in Detail.

(5)

(1) List out different logic development techniques and explain flowchart with example.

- ❖ Logic is a tool to develop reasonable conclusions based on given set of data.
- ❖ There are following logic development techniques available in c:
 - Flow chart
 - Algorithm
- ❖ Flow chart is graphical representation of our data.
- ❖ It also give idea about sequence of flow.
- ❖ Flow chart can represent logic flow and process of data.

Flowchart Symbols:

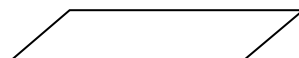
- Flow chart support following symbols to represent the logic.

1. **Terminator:**



- This symbol represents beginning and ending point in the program.

2. **Input/Output :**

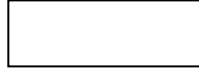


- It is used to represent input and output of the data.



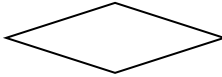
SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

3. Process :



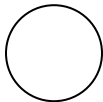
- This symbol is used to represent different process of operations.

4. Decision Symbol:



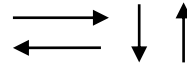
- This symbol is used to give condition and take decision.

5. Connector Symbol:



- It is used to represent logic flow of out data.

6. Data Flow Symbol / Flow Direction:



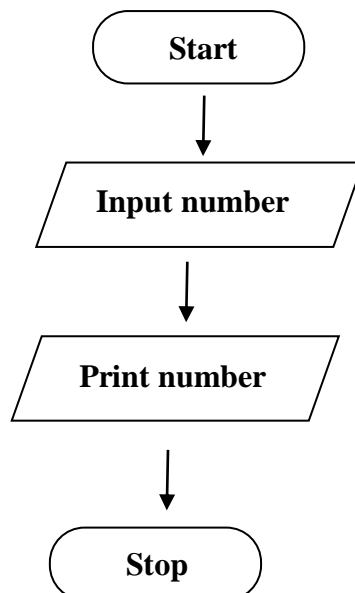
It is used to represent direction flow of our data.

❖ Advantages of flow chart :-

1. Using flow chart we can easily understand logic of code.
2. It represent clear view of the program.
3. It represent graphical behavior of the data.

Example:

A flowchart to print the number.





Q-2(A) Objective Type Questions

(4)

(1) What is Constant?

Constant is a fixed value that never change during execution of the program.

(2) Relational operators are also known as _____ operators.

Comparison

(3) List out different logical operators available in C.

AND

OR

NOT

(4) The range of Integer datatype is _____ to _____.

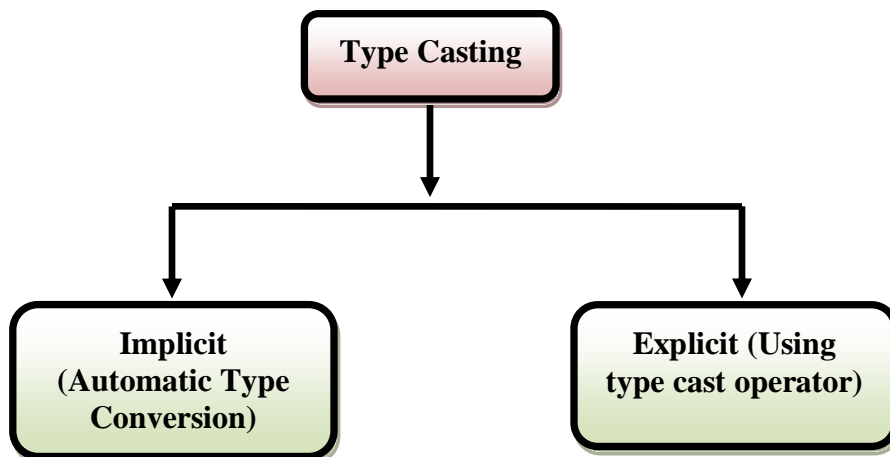
-32768 to +32767

Q-2 (B) Answer in Brief.

(2)

(1) What is type Casting ? Explain in brief.

- ❖ Type Casting is process of converting one data type into other data type.
- ❖ There are two types of type casting available.





1. Implicit Type Conversion :

- ❖ The implicit type casting is also known as automatic conversion.
- ❖ The implicit conversion is used to convert original data type into other data type.
- ❖ In this type of casting data should be converted from smaller data type to larger data type.

Conversion Hierachy

Long double	↑
Double	
Float	
Unsigned long int	
Long int	
Unsigned int	
Int	
Short	char

2. Explicit Type

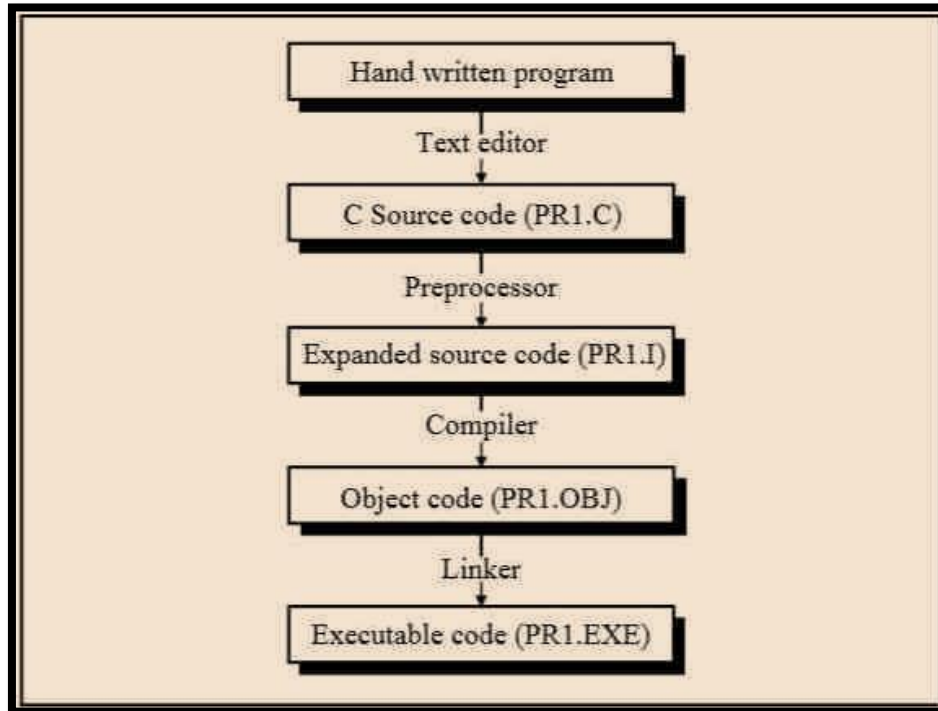
Conversion :

- ❖ Sometimes we have to convert different data type into single data type.
- ❖ But Explicit casting is not automatic conversion because we have to use type cast operator.
- ❖ We can change type of the data from smaller to larger and larger to smaller.
Ex :- `b = (float) a;`

Q-2 (C) Answer in Brief.

(3)

(1) Briefly Explain pre-processor in C.



- ❖ Pre – Processor directives are executed directly by compiler.
- ❖ Any pre-processor in C language always begins with #(Hash) Sign.
- ❖ C language support following pre-processor directives.

1. # include :

- # include is pre-processor that use to include header files in our program.
- If we not include particular header file then compiler give error related to “prototype”.
- **Example :-** #include<stdio.h>
- To declare any header file you have to use angular brackets (< >).

2. # define :

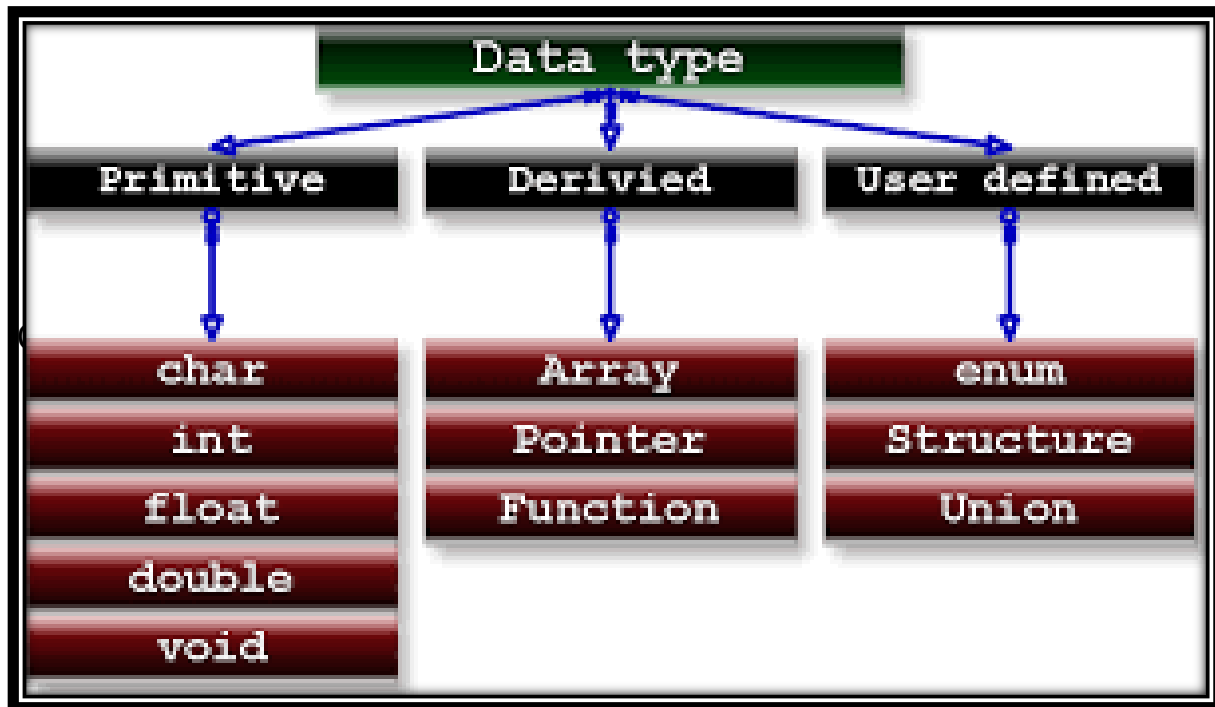
- # define pre – processor is used to create symbolic constant.
- You can not change value of constant once it is defined.
- You can not re-define the value of constant using # define.
- **Example :-** # define pi 3.14
- **Note :-** If you use # define then statement will not ends with ;

Q-2 (D) Answer in Detail.

(5)



(1) Write note on datatypes in C.



Detail :-

- ❖ Data types are used to store different types of values in variable.
- ❖ In C language , there are three types of data types available.

1. Primary/ Basic/ Fundamental Data types. (Integer, Character, Float, void)
2. Derived Data types. (Array, Pointer)
3. User Define Data types. (Structure, Union, enum, typedef)

1. Primary Data Type :

- The primary data types are also called fundamental or built-in or basic data types.
- The primary data types include mainly following :

(i) Integer :-

- This data type is used to store only integer value.
- It occupies 2 bytes of memory.



(ii)Float :- (Long)

- This data type is used to store integer value & Decimal Value.
- It occupies 4 bytes of memory.

(iii)Char :-

- This data type is used to store single or multiple characters.
- It occupies 1 byte of memory.

(iv)Double :-

- This data type is used to store integer and decimal values.
- It occupies 8 bytes of memory.

(v)long double :-

- This data type is used to store integer and decimal values with high range.
- It occupies 10 bytes of memory.

2. User Defined Data Type :

- User defined data type is a type of data that decided by userself.
- User defined data type include mainly the following.

(i)Typedef :-

- It provide facility to represent identifier of particular data type.

(ii)Enum :-

- It provide facility to declare multiple constant at the same time.

(iii)Structure:-

- It is used to create member of different data type at the same time.

(iv)Union:-

- It is used to create member of different data type at the same time.

Q-3(A) Objective Type Questions

(4)

(1)What is Loop?



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

Loop is used to execute block of code for number of times until given condition become false.

(2) List out Ternary operators available in C.

? (Question Mark)

: (Colon)

(3) Write down syntax of While loop.

While (condition)

{

<statements>

<increment / decrement>

}

(4) In Goto statement the label is followed by _____.

: (colon)

Q-3 (B) Answer in Brief.

(2)

(1) Explain Do...While loop with syntax and example.

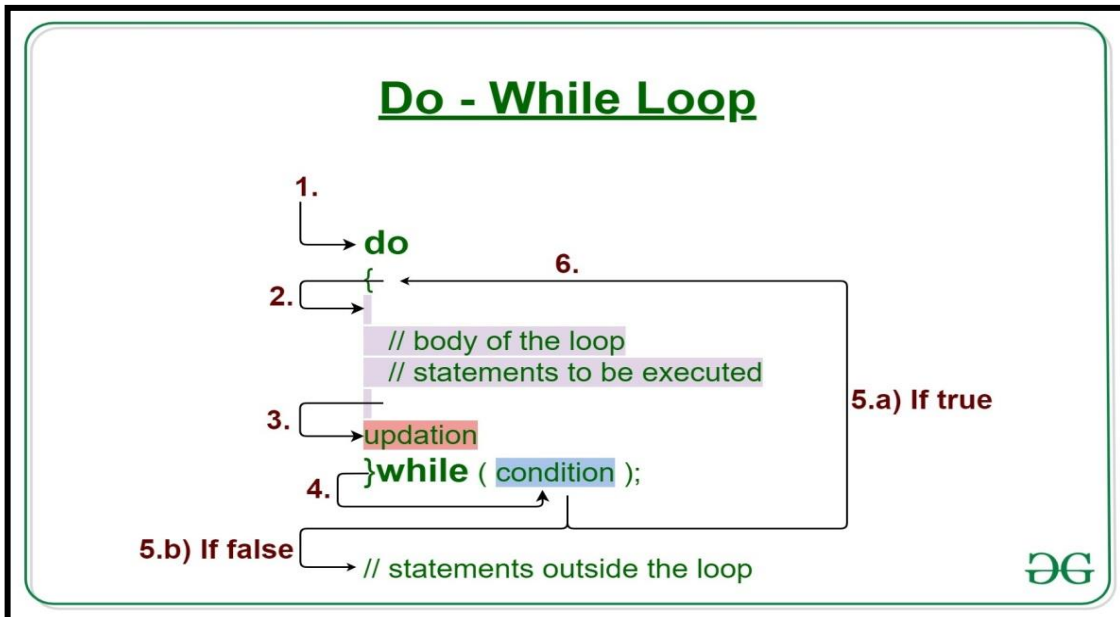
- ❖ Loop provide easy way to repeat the task and reduce the code.
- ❖ A loop is a group of statements that are executed until some condition satisfied.
- ❖ In C language we have to provide 3 criterias
 - (i) Starting value
 - (ii) Condition
 - (iii) Increment/Decrement
- ❖ In C language there are 3 types of looping statements.
 - (1) While loop [Entry Controlled loop]
 - (2) Do..while loop [Exit Controlled loop]
 - (3) For loop [Entry Controlled loop]

◆ Do..While loop:



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- Do..While loop is also known as Exit controlled loop.
- Do..While loop is exit controlled loop because the statement will be execute first and condition will be checked at last.
- In this loop , if condition become false then atleast once statement will be execute because condition will be checked at last.



○ **Syntax:**

```
do
{
    statements;
    (increment/decrement);

}while(condition);
```

○ **Example:**

```
int i=1;

clrscr();
do
{
    printf("\n %d",i);
```



```
    i++;  
}while(i<=10);
```

Q-3 (C) Answer in Brief.

(3)

(1) Explain If...Else with syntax and example.

- ❖ C language provide a very powerful control structure that is “if statement.”
- ❖ It is decision statement that use to take some decision.

✓ **if-else statement:**

- In this statement first of all , condition will be checked.
- If condition become true than statement following if will be execute.
- If condition become false then statement following else will be execute.

Syntax:-

```
    If(expression)  
    {  
        Statement-block;  
    }  
    else  
    {  
        Statement-block;  
    }  
    Next statement;
```

Example:-

```
int a=10,b=20;  
if(a>b)  
{  
    Printf(“a is max”);  
}  
Else  
{  
    Printf(“b is max”);  
}
```



Q-3 (D) Answer in Detail.

(5)

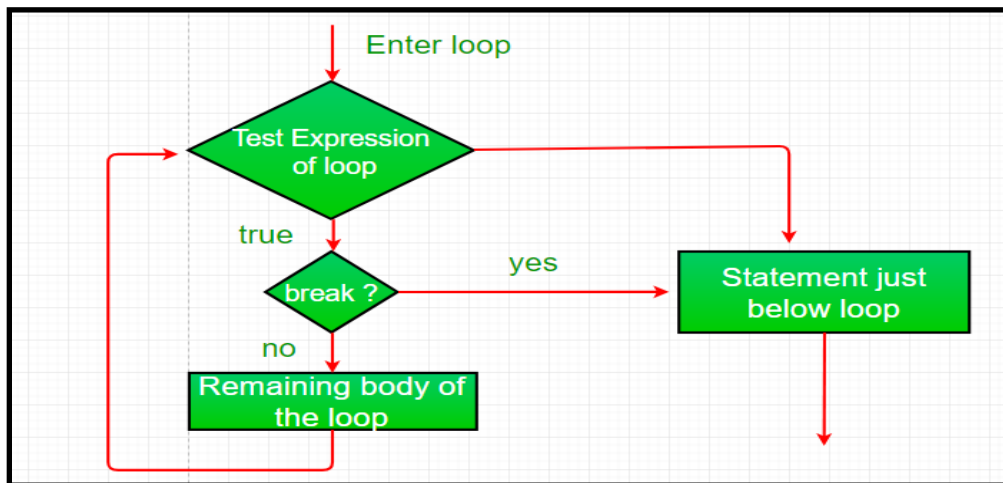
(1) write note on different jumping statements.

- ❖ The jumping statements are used to transfer the program control.
- ❖ There are following jumping statements available in C.

- (i) Break
- (ii) Continue
- (iii) Goto

◆ **break Statement:**

- Sometimes it is necessary to stop execution of the program.
- The break statement is unconditional statement.
- Break statement is used for terminating the loop or jump out from the loop.



Syntax:

Break;

Example:

```
int i=1;
clrscr();
While (i<=10)
{
    if(i==5)
```



```
        {
            break;
            printf("%d",i); // 5 times only
        }
    }
```

◆ continue Statement:

- The continue statement is used to pass the control to next iteration(loop).
- It is also unconditional statement.
- Continue does not terminate the loop but only skip the statement followed by if.

Syntax:

```
Continue;
```

Example:

```
Int I;
Clrscr();
For(i=1;i<=10;i++)
{
    If(i>4)
    {
        Continue;
    }
    Else
    {
        Printf("*\n");
    }
}
```

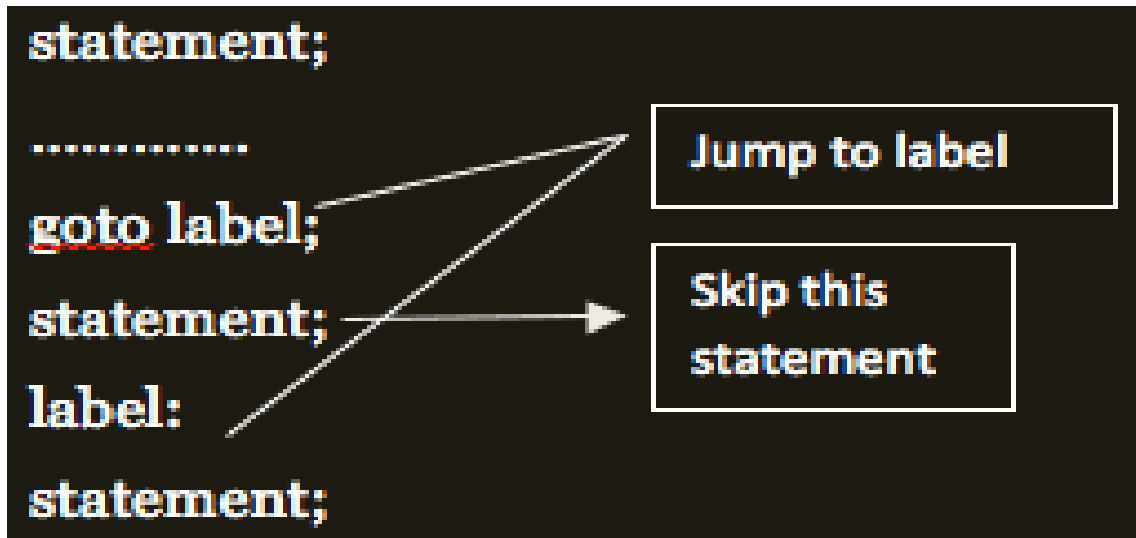
◆ Goto Statement:

- Goto statement is also unconditional statement.
- Goto is used to transfer the control (cursor) from one point to another point in the program.
- The goto statement is marked by label.



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- The label is variable name followed by colon(:).



Syntax:

```
Goto <variable name>;
```

Example:

```
Int i=10,j=20;
Clrscr();
If(i>j)
{
    goto l;
}
Else
{
    goto j;
}
l: printf("I is max");
j:printf("j is max");
```

```
Continue;
```

```
}
Else
{
```




```
        Printf("*\n");
    }
}
```

Q-4(A) Objective Type Questions

(4)

(1) What is nested loop?

In C, if one loop is created inside the another loop then it is called nested loop.

(2) Which type of value can be specified in switch statement?

Int , char

(3) _____ is known as exit controlled loop in c.

Do...while loop

(4) Write down syntax of For loop.

```
For(initialization ; condition ;increment / decrement>
{
    <statements>
}
```

Q-4(B) Answer in Brief.

(2)

(1) Write a program that perform nested if.

Example:-

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

void main()
{
    int n1,n2,n3;
    clrscr();
    printf("\n enter the n1,n2,n3");
    scanf("%d",&n1,&n2,&n3);
        if(n1>n2)
        {
            If(n1>n3)
```



```
        printf("\n n1 is max");
    else
        printf("\n n3 is max");
    }
    else
    {
        If(n2>n3)
            printf("\n n2 is max");
        else
            printf("\n n3 is max");
    }
    getch();
}
```

Q-4(C) Answer in Brief.

(3)

(1) Give difference between Break and Continue Statement.

Break	Continue
1) Break is used to terminate the block & get the control out of the loop.	1) Continue is used to get the control to the next iteration of the loop.
2) Break statement can be used in both switch case & loop.	2) Continue statement can be only used with looping statement.
3) Syntax : <break>;	3) Syntax : <continue>;

Q-4 (D) Answer in detail.

(5)

(1) Write note on switch case statement.

- ❖ The switch case statement is also known as multi condition statement.
- ❖ It provide facility to check the condition.
- ❖ Switch case make it easy to give the choice and take the decision.
- ❖ It also support multiple case values. But at a time only one case value Become true.
- ❖ If any case value match with the condition than finally statement following that case will be execute.



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- ❖ If no any case value match with the condition then finally the statement following default will be execute.

Break :-

- ❖ This statement is used to end or stop process of particular case.
- ❖ Without break the programme will continue to the next case.

Syntax:-

```
Switch(Condition)
{
    Case Expression 1:
        Statement block1;
        break;

    Case Expression 2:
        Statement block2;
        break;

    .....
    .....

    default:
        Statement block;
}
```

Example:

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

void main()
{
    int color=1;
    clrscr();
    printf("\n enter the color 1. RED 2. GREEN 3. BLUE");
    scanf("%d",&color);
```



```
switch(color)
{
    case 1:
        printf("\n RED");
        break;
    case 2:
        printf("\n GREEN");
        break;
    case 3:
        printf("\n BLUE");
        break;
    default:
        printf("\n NO COLOR");
}
getch();
}
```

Q-5(A) Objective Type Questions

(4)

(1) What is Function?

A function is a block of code that performs a specific task.

(2) Write down use of atof().

atof() function in C language converts string data type to float data type.

(3) Write down use of fabs().

fabs function returns the absolute value of a floating-point number.

(4) Which Library file must be include to perform string related functions.

<String.h >

Q-5 (B)) Answer in Brief .

(2)

(1) Explain getchar() and putchar() with Example.

getchar():



- This function is used to get a single character from standard input.

Syntax:

```
<variable> = getchar();
```

Example:

```
Void main()
{
    Char ch;
    Printf("enter character");
    Ch = getchar();
    Getch();
}
```

putchar():

- This function is used to print a single character to output stream.

Syntax:

```
Putchar(<character variable>)
```

Example:

```
Void main()
{
    Char ch;
    Printf("enter character");
    Ch = getchar();
    Putchar(ch);
    Getch();
}
```

Q-5 (C)) Answer in Brief.

(3)

(1) Give difference between call by value and call by reference.



<i>CALL BY VALUE</i>	<i>CALL BY REFERENCE</i>
CALLING FUNCTION SENDS COPIES TO DATA.	CALLING FUNCTION SENDS ADDRESS OF DATA.
THE FORMAL PARAMETERS ARE ORDINARY VARIABLES.	THE FORMAL PARAMETERS ARE POINTER VARIABLE.
ATMOST ONLY ONE VALUES CAN BE SENT BACK TO THE CALLING FUNCTION.	SEVERAL RESULTS CAN BE SENT BACK TO THE CALLING FUNCTION.
ACTUAL PARAMETERS ARE AFFECTED BY CHANGES MADE WITHIN THE FUNCTION.	DIRECT CHANGES ARE MADE TO THE ACTUAL PARAMETERS.

Q-5 (D)) Answer in Detail .

(5)

(1)Write note on different memory allocation functions.

- In c language there are following memory allocation functions available:

1.malloc():

- This function is used to allocate single block of requested memory.
- It always return NULL if memory is not sufficient.

Syntax:

`Ptr = (cast_type) malloc(byte_size)`

Example:

`Ptr = (int) malloc(100*sizeof(int))`

2 calloc():

- This function is used to allocate multiple block of requested memory.
- It always return 0(zero) if memory is not sufficient.

Syntax:

`Ptr = (cast_type) calloc(number ,byte_size)`



Example:

```
Ptr = (float) calloc(25,sizeof(float))
```

3.realloc():

- If memory is not sufficient for malloc & calloc at that time you can use realloc() to re – allocate the memory space.
- It always return 0(zero) if memory is not sufficient.

Syntax:

```
Ptr = realloc(ptr , newsize)
```

Example:

```
Ptr = realloc(p,n*sizeof(int))
```

4.free():

- The memory allocated by malloc and calloc will be released or free by calling free().
- It occupies memory until you exit your program

Syntax:

```
Free(<pointer variable>)
```

Example:

```
Void main()
{
    Char *str;
    Clrscr();
    Str = "tutorialpoint";
    Printf("string=%s,address=%u\n",str,str);
    Free(str);
    getch();
}
```



}

Q-6(A) Objective Type Questions

(4)

(1) Writedown use of fflush().

fflush() is used to clear or flush the output buffer of the stream.

(2) Write down syntax for calloc().

ptr = (castType*)calloc(n, size);

(3) _____ is used to return length of any given string.

Strlen()

(4) Writedown use of strcmp().

The strcmp() compares two strings and returns an integer value based on the result.

Q-6 (B)) Answer in Brief.

(2)

(1) Explain toupper() and tolower() with Example.

Toupper():

- This function is used to convert lowercase character into uppercase.

Syntax:

toupper(<character>)

Example:

```
#include<errno.h>
```

```
Void main()
```

```
{
```

```
    Char ch= 'a';
```

```
    Clrscr();
```

```
    Int c=toupper(ch);
```




SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

```
        Printf("%c",c);
        getch();
    }
```

Tolower():

- This function is used to convert uppercase character into lowercase.

Syntax:

```
tolower(<character>)
```

Example:

```
#include<errno.h>

Void main()
{
    Char ch= 'A';
    Clrscr();
    Int c=tolower(ch);
    Printf("%c",c);
    getch();
}
```

Q-6 (C) Answer in Brief.

(3)

(1) Explain Recursion with suitable example.

- Recursion is the process of repeating items in a self-similar way. ...
- The C programming language supports recursion, i.e., a function to call itself.
- But while using recursion, programmers need to be careful to define an exit condition from the function, otherwise it will go into an infinite loop.

Example:

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

void main()
```



```
        {
            clrscr();
            Recursion();
        }
        printf("\n\n Enter Number : ");
scanf("%d",&no
        void Recursion()
        {
            int no;
            printf("\nRecursion... ");

        );
        if (no==3)
            exit(0);
        else
            Recursion();
        }
```

Q-6 (D)) Answer in Detail.

(5)

(1) Explain different Storage classes in C.

❖ Storage class determines the scope (region of the program in which a variable is available for use.) and lifetime (duration in which a variable exist in the memory during execution of a program.). Storage class tells:

- Where the variable is stored.
- Initial value of the variable.
- Scope of the variable. Scope specifies the part of the program which a variable is accessed.
- Life of the variable.

There are four types of storage class:

1. Automatic Storage class
2. Register Storage class
3. Static Storage class



4. External Storage Class

1. Automatic Storage Class:

- The variables that are local to the function are known as automatic variables i.e. declared within the function.
- It is the default storage class for the variables declared in a function.

- ◆ Keyword : auto
- ◆ Storage Location : Main memory
- ◆ Initial Value : Garbage Value
- ◆ Life : Local (function in which it is declared).
- ◆ Scope : Local to the block in which variable is declared.

2. Register Storage Class:

- Register variables are declared inside the function.
- When the variables are required very often during the program, they may be declared as register variable.

- ◆ Keyword : register
- ◆ Storage Location : CPU Register
- ◆ Initial Value : Garbage
- ◆ Life : Local to the block in which variable is declared.
- ◆ Scope : Local to the block.

3. Static Storage Class:

- Static storage class can be used only if we want the value of a variable to persist between different function calls.
- If the control comes back to the same function again, the static variables have the same values they had last time around.
- By default, global variables are static variables.

- ◆ Keyword : static
- ◆ Storage Location : Main memory



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- ◆ Initial Value: Zero and can be initialize only once.
- ◆ Life : depends on function calls and the whole application or program.
- ◆ Scope : Local to the block.

4. External Storage Class:

- The variables of external storage class can be referred to as 'global or external variables.'
- They are declared outside the functions and can be invoked at anywhere in a program.
- The only drawback of declaring the variables as extern is it wastes lot of memory because these variables remain active throughout the life of the program.
 - ◆ Keyword : extern
 - ◆ Storage Location : Main memory
 - ◆ Initial Value : Zero
 - ◆ Life : Until the program ends.
 - ◆ Scope : Global to the program.

Q-7(A) Objective Type Questions

(4)

(1) Write down use of exit().

The exit() function is used to terminate program execution and to return to the operating system.

(2) .(dot) operator can be used to access _____ of structure.
members

(3) Array can be initialized at _____ & _____.
Compile time & runtime

(4) What is nested structure?



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

If one structure declare inside another structure then it is called nested structure.

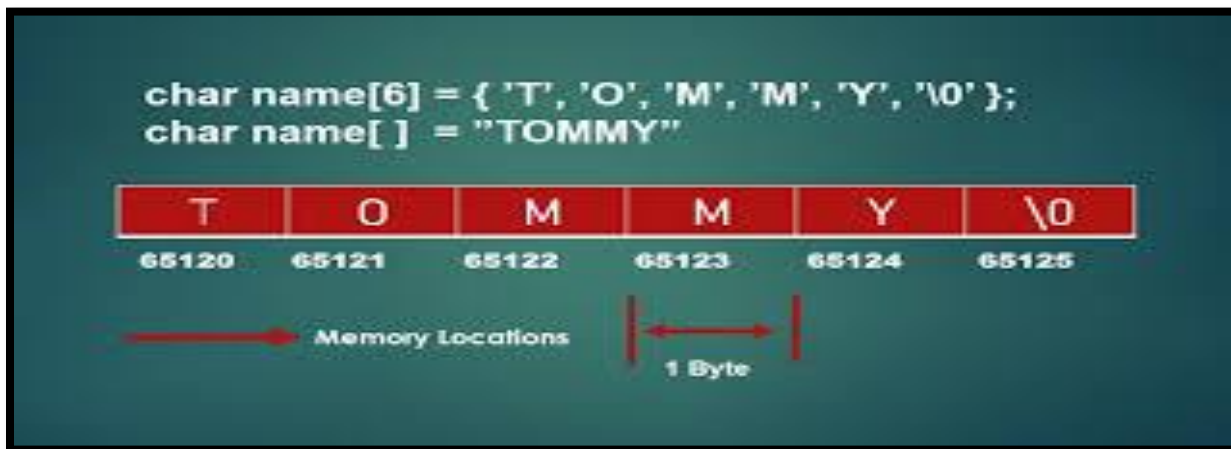
Q-7 (B) Answer in Brief.

(2)

(1) Write note on String array.

- ❖ The array of character is implemented in c which is known as string.
- ❖ The declaration of string array or character array can be like :
Char str[5] = { 'a','b','c','d','e','\0' }
- ❖ A NULL char '\0' must be added at the end of array string.
- ❖ The character array need one more space to store NULL char.
- ❖ C also provide facility to declare character array without specifying number of elements.
- ❖ In this case array size will be declare automatically.

❖ String Array Memory Allocation :-



Example :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    char arr[3][12]= { "Rose", "India", "technologies" };
    printf("Array of String is = %s,%s,%s\n", arr[0], arr[1], arr[2]);
}
```



```
    getch();  
}
```

Q-7 (C) Answer in Brief.

(3)

(1) Write down code to get 5 elements from user and count how many are odd and how many are even.

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
int a[5],i,even=0,odd=0;  
clrscr();  
for(i=0;i<5;i++)  
{  
    printf("\n five enter number");  
    scanf("%d",&a[i]);  
if(a[i]%2==0)  
    even++;  
else  
    odd++;  
}  
printf("\n even number=%d\n",even);  
printf("\n odd number = %d\n",odd);  
getch();  
}
```

Q-7 (D) Answer in Detail.

(5)

(1) Explain Array with Matrix by suitable example.

- ❖ Array of Array is known as 2 – D array.
- ❖ 2-D array in C is also known as Matrix.
- ❖ Two dimensional array or multi-dimensional array are used to represent data in matrix form.



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

❖ Suppose , We declare array like `a[3][3]` then Matrix can be store as array like following:

	← i → row →			
j ↓ column	<code>a [0][0]</code>	<code>a [0][1]</code>	<code>a [0][2]</code>	<code>a [i][j]</code>
	<code>a [1][0]</code>	<code>a [1][1]</code>	<code>a [1][2]</code>	element
	<code>a [2][0]</code>	<code>a [2][1]</code>	<code>a [2][2]</code>	

Syntax:

`<data-type> <array_nm> [row_subscript][column-subscript];`

Example:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[3][3], i, j;
    clrscr();
    printf("\n\t Enter matrix of 3*3 : ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            scanf("%d",&a[i][j]); //read 3*3 array
        }
    }
}
```



```
printf("\n\t Matrix is : \n");
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
    {
        printf("\t %d",a[i][j]); //print 3*3 array
    }
    printf("\n");
}
getch();
}
```

Q-8(A) Objective Type Questions

(4)

(1) List out types of array.

One dimension array

Two dimension array

Multi dimension array

(2) _____ keyword is used to create structure.

Struct

(3) Write down syntax for one dimension array.

data_type array_name[size];

(4) Array is _____ datatype.

Derived

Q-8 (B) Answer in Brief.

(2)

(1) Give difference between structure and union.

<u>Structure</u>	<u>Union</u>
1) In structure, each member have its own Storage Space.	1) In Union, All the members can share Same Memory Space.



2)Structure can be declared using “struct” keyword.	2)Union can be declared using “union” keyword.
3)The size of structure depends on length of its elements.	3)The size of union depends on maximum length of integer elements.

Q-8 (C) Answer in brief.

(3)

(1) Explain Structure with UDF by Example.

- ❖ UDF stands for User Defined Function.
- ❖ UDF means the functions that are created by the user itself.
- ❖ It is also possible to pass structure as arguments to a function.
- ❖ When you create UDF with the help of structure then it is called UDF with structure.
- ❖ The following Example will display how to pass structure as argument in UDF.

Example:

```
#include<stdio.h>
#include<conio.h>
struct data
{
    float amount;
    char fname[30];
    char lname[30];
}rec;

void print_rec(struct data x)
{
    printf("\n %s %s %f",x.fname,x.lname,x.amount);
}
```



```
void main()
{
    clrscr();
    printf("\n Enter the donor's first and last names");
    scanf("%s %s", rec.fname, rec.lname);
    printf("\n Enter the donation amount");
    scanf("%f",&rec.amount);
    print_rec(rec);
    getch();
}
```

Q-8 (D) Answer in Detail.

(5)

(1) Write a program to get detail of 5 students and print it on the screen using array of structure.

```
#include <stdio.h>
struct student
{
    char firstName[50];
    int roll;
    float marks;
} s[10];

int main() {
    int i;
    printf("Enter information of students:\n");

    // storing information
    for (i = 0; i < 5; ++i) {
        s[i].roll = i + 1;
    }
}
```



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

```
printf("\nFor roll number%d,\n", s[i].roll);
printf("Enter first name: ");
scanf("%s", s[i].firstName);
printf("Enter marks: ");
scanf("%f", &s[i].marks);
}
printf("Displaying Information:\n\n");

// displaying information
for (i = 0; i < 5; ++i) {
    printf("\nRoll number: %d\n", i + 1);
    printf("First name: ");
    puts(s[i].firstName);
    printf("Marks: %.1f", s[i].marks);
    printf("\n");
}
return 0;
}
```

Q-9(A) Objective Type Questions

(4)

(1) Pointer can be declare using _____ operator.

***(asterisk operator)**

(2) Pointer value means _____.

Address of other variable

(3) What is datafile?

Data files allow us to store necessary informations permanently.

(4) Pointer is _____ datatype.

Derived



Q-9 (B) Answer in Brief.

(2)

(1) Give difference between Text file and Binary file.

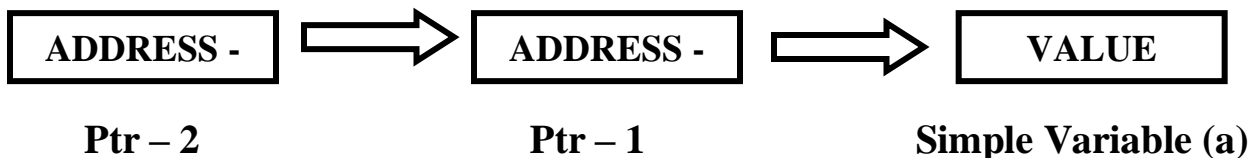
<u>Text File</u>	<u>Binary File</u>
<ol style="list-style-type: none">1) Text file is the file created by user.2) The extension of text file may be .txt, .c etc.3) In text file, there is special character ('\0') which indicates end of file.4) In text file, data is stored in text format, so if we have integer value 1234 then it will require 5 bytes (1234\0)	<ol style="list-style-type: none">1) Binary file is the file created by the computer.2) The extension of binary file may be .exe, .obj etc.3) In binary file there is no such special character that indicates end of file.4) In binary file, data is stored in the format of main memory. So if we have integer value 1234 then it will require only 2 bytes.

Q-9 (C) Answer in brief.

(3)

(1) Explain pointer to pointer with example.

- ❖ In C Programming it is also possible that pointer variable is pointed by another pointer variable.
- ❖ This process is called pointer to pointer.



- ❖ Pointer variable ptr-2 having the address of ptr-1, same way ptr-1 having the address of simple variable(a).
- ❖ This process is known as multiple indirection.
- ❖ Pointer to pointer variable must be declared using additional indirection operation(**) in front of variable name like ;
Int **p;



Example:

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

void main()
{
    int a 10;
    int *p;
    int **a;
    clrscr();
    p=&a;
    a=&p;
    clrscr();
    printf("\n value of p = %d", *p); //will print address of a
    printf("\n address of p=%d=u",p); // will print value of a
    printf("\n value of a = %d", *p);
    printf("\n value of **a=%d", **a);
    getch();
}
```

Q-9 (D) Answer in Detail.

(5)

(1) Write down program code to copy content of one file into another file.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fp1,*fp2;
    char ch;
    clrscr();
    fp1 = fopen("snehal.txt","r");
    fp2 = fopen("copy.txt","w");
    while((    ch=fgetc(fp1))!=EOF)
    {
```



```
        fputc(ch,fp2);
    }
    printf("file copied successfully");
    fclose(fp1);
    fclose(fp2);
    getch();
}
```

Q-10(A) Objective Type Questions

(4)

(1) To indicate end of the file _____ character can be used.

Null ('\0')

(2) Pointer to pointer variable must be declare using _____.

**** (double pointer)**

(3) _____ indicate number of arguments to be passed at command line argument.

Argc

(4) To Read data to the particular file , the file must be open in ____ mode.

r

Q-10 (B) Answer in Brief.

(2)

(1) Explain Pointer to Array with suitable example.

❖ When array is declared ,compiler decide the memory to store array elements.

❖ Pointer to array is used to store whole array into single pointer variable.

❖ We can declare pointer to array like following :

```
Int a[5];
```

```
Int *p;
```

```
P=a;
```

```
P=&a[0];
```



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- ❖ For Example , you can store each elements of array which can be point out by pointer.
- ❖ The array elements using pointer can be store like following ;

Elements	10	20	30	40	50
Array Index	a[0]	a[1]	a[2]	a[3]	a[4]
Pointer	p	p+1	p+2	p+3	p+4
Address	1000	1002	1004	1006	1008

Example :-

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

```
int main()
{
    int i;
    int a[5] = {1, 2, 3, 4, 5};
    int *p = a; // same as int*p = &a[0]
    for (i = 0; i < 5; i++)
    {
        printf("%d", *p);
        p++;
    }

    return 0;
}
```

Q-10 (C) Answer in brief.

(3)

(1) Explain fseek(), ftell() and rewind().

(1)fseek() :-



SHREE H.N.SHUKLA COLLEGE OF I.T. & MGMT.

- It is used to set file position means that it is used to move file position to given location in the file.

Syntax :- `fseek(<file pointer> , <off set> , <position>)`

Example :- `fseek(fp , s,seek_set)`

0- SEEK_SET -> beginning of the file

1- SEET_CUR -> Current position

2- SEEK_END -> end of the file

(2)Ftell() :-

- It is used to return current file position of given file.

Syntax :- `ftell(<file pointer>)`

Example :- `ftell(fp)`

(3)Rewind() :-

- It is used to set file position to beginning of the file or starting of the file.

Syntax :- `rewind(<file pointer>)`

Example :- `rewind(fp)`

Q-10 (D) Answer in Detail.

(5)

(1) Explain Command Line Argument with Example.

- ❖ Command line arguments are the argument that are pass when the program is execute.
 - ❖ In command line argument , `main()` also take argument from the user.
 - ❖ `Main()` take two arguments
- (i) `Argc` – It indicate number of arguments to be passed
- (ii) `Argv[]` – It is pointer array which point to each argument that is passed with the `main()`.



Command Line Arguments (Cont...)

- It should be noted that **argv[0]** holds the name of the program itself and **argv[1]** is a pointer to the first command line argument supplied, and ***argv[n]** is the last argument.
- If no arguments are supplied, **argc** will be one, and if you pass one argument then **argc** is set at 2.
- E.g. F:\C\L04>CommandLineArguments.exe test

```
argv  CommandLineArguments.exe  test
argc is 2 here
```

(iii) main() takes two arguments:

- i. **argc** → Number of arguments passed
- ii. **argv[]** → pointer array which points to each argument which is passed to main().

Note:-

- argv[0] → name of the program that is executed.
- argv[1] → First argument
- argv[2] → Second argument

The screenshot shows the Turbo C++ IDE window titled 'TC - exit - exit - exit - exit - exit'. The menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Window, and Help. The 'File' menu is open, showing options like New, Open..., Save, Save as..., Save all, Change dir..., Print, DOS shell (highlighted with a red box), and Quit. The main editor area shows C code for a program named 'ADD.C'. The code includes a loop that iterates through command-line arguments and calculates their sum. The status bar at the bottom indicates 'F1 Help | Temporarily exit to DOS'.