Government Girls' Polytechnic, Bilaspur

Name of the Lab: **Programming Lab**

Practical : **Programming in C Lab**

Class: **2nd Semester (CSE, IT)**

Teachers Assessment: 40 End Semester Examination:100

EXPERIMENT NO:-1

1. OBJECTIVE:-Assignment to prepare general algorithms and flow chart.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY:-

The flowchart

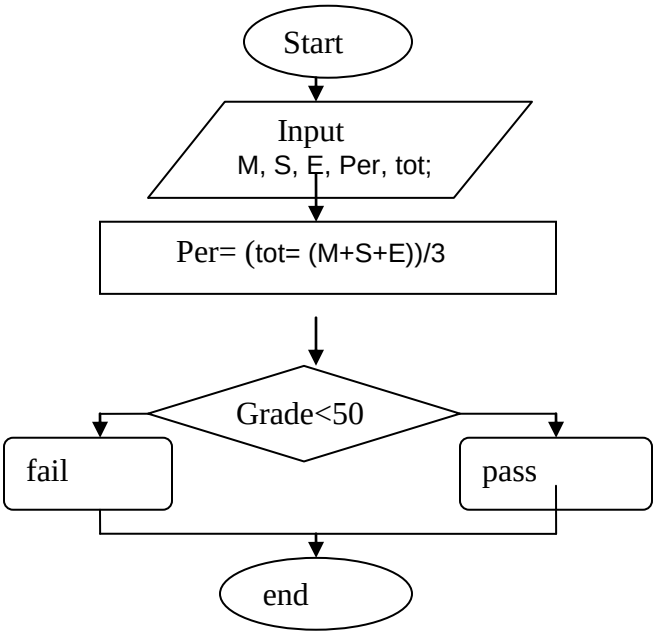
A graphical representation of the sequence of operations in an information system or program .information system flowcharts show how data flows from source documents through the computer to final destination to users.

Algorithm

1>produce an ordered sequence of steps that describe solution of problem.
2>this sequence of step is called an algorithm.

```
Step1:-input M1, M2, M3, M4
Step2: grade-(M1+M2+M3+M4)/4
Step3: if (grade<50) then
        Print "fail"
        Else
        Print "pass"
        End if
```

5. FLOW CHART:-



6. OBSERVATIONS:- studied all the blocks of flowcharts.

EXPERIMENT NO:-2

1. OBJECTIVE:-Assignment to write character, operators symbols of C Language.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY:-In C Language character values are stored in 1 byte, and are encoded as numbers using the ASCII.

Special characters:

character	usage
{ }	Used to delimit blocks of code
()	Used to delimit the parameter, list in a function declaration definition or call
[ ]	Used to delimit the index of an array variable.
< >	Used to delimit the name of standard header files.
>	Greater –than comparison operator.
<	Less-than comparison operator.
==	Is equal to comparison operator.
=	Assignment operator.
>=	Greater than or equal to comparison operator.
<=	Less than or equal to comparison operator.
!=	Is not equal to comparison operator.
!	Logical not operator.
~	Bit wise complement operator.
/ *	Used to begin a comment.
*/	Used to end a comment.
‘ ‘	Used to delimit a single character.
“ “	Used to delimit a string of character.
*	Multiplication operator and pointer dereference operator and pointer in declaration.
%	Modulus operator.
/	Division operator.
+	Addition operator.
-	Subtraction operator or negation operator.
++	Increment operator.

- -	Decrement operator.
^	Bitwise exclusive or operator.
!	Bitwise or operator.
||	Logical operator.
&	Bitwise and operator.
>>	Bit shift right operator.
<<	Bit shift left operator.
?:	Conditional expression operator.
#	Used to include a preprocessor.
\	Used to introduce a special character.
;	Used to end each statement.
+=	Used to perform an addition and assign the result.
-=	Used to perform a subtraction and assign the result.
*=	Used to perform a multiplication and assign the result.
/=	Used to perform a division and assign the result.
%=	Used to perform a modulus and assign the result.
^=	Used to perform an exclusive OR and assign the result.
&=	Used to perform and AND and assign the result.

Operators:--

Arithmetic operators	
Operator name	syntax
Unary plus	+a
Addition(sum)	A+b
Prefix increment	++A
Postfix increment	A++
Assignment by addition	A+=b
Unary minus(negation)	-A
Subtraction(difference)	A-B
Prefix decrement	--A
Postfix decrement	A—
Assignment by subtraction	A-=B
Multiplication (product)	A*B
Assignment by multiplication	A*=B
Division	A/B
Assignment of division	A/=B

Modula (remainder)	A%B
Assignment of Modula	A%=B

5. OBSERVATIONS:-All the character, operator of c –language have been studied.

EXPERIMENT NO:-3

1. OBJECTIVE:-Assignment to identify valid and invalid variables, constants, and expressions.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:- c++ software.

THEORY:-

Variable:- 1.variable is a tool to reserve space in computer memory.
2. The reserved space is given a name which we called a variable name.
3. Variable store value during a program execution.
Example: - I=10
I is a variable

Rules for constructing variable name:-

1. A variable name is any combination of 1 to 31 alphabets, digits or underscore.
2. The first character in the variable name must be an alphabet or underscore.
3. No commas or blank are allowed within a variable name.
4. No special symbol other than an underscore (as in gross_sal) can be used in a variable name.
5. ex. Si_int ,m_hra,pop_e_98.

Example of valid and invalid variable names:-

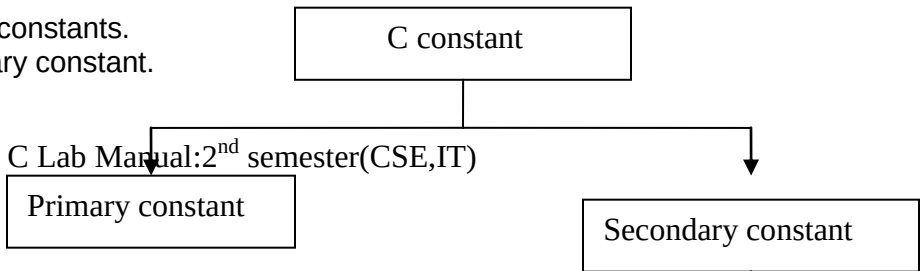
Valid variable	Invalid variable	Reason
Myvariable	My variable	Contains a space
Room4	4room	Start with number
Lastname	Lastname?	Has illegal character

Note: - characters are case sensitive in c, therefore, my variable .and my variable are all different variable.

Constants:- Some variable don't change value during the program execution are called constant.
Ex. i=7;
7 is a constant which is being stored in a location which has been given the name i.

Types of constant:- Constant can be divided into two major categories.

A. Primary constants.
B. Secondary constant.



Rules for constructing integer constant:-

1. Integer constants must have at least one digit.
2. It must not have a decimal point.
3. it can be either positive or negative.
4. if no sign precedes an integer constant , it is assumed to be positive.
5. No commas or blanks are allowed within an integer constant.
6. The allowable range for integer constant is -32768 to 32767.
7. ex.426

Example of valid and invalid integer constant:-

Valid integer constant	Invalid integer constant	Reason
426	426.0	Contains a decimal point
456	45 6	Contains space
789	78,9	Contains commas

Rules for constructing real constant:-

1. A real constant must have at least one digit.
2. It must have a decimal point.
3. It could be either positive or negative.
4. Default sign is positive.
5. No commas or blank are allowed within a real constant.

Ex. +324.34

Example of valid and invalid real constant:-

Valid real constant	Invalid real constant	Reason
426.34	42634	Not Contains decimal point
456.89	45 6.89	Contains space
789.79	78,9.79	Contains commas

Rules for constructing character constant:-

1. A character constant is a single alphabet .a single digit or a single special symbol enclosed within single inverted commas. Both the inverted commas should point to the left

Ex. ‘A’ is a valid character constant whereas ‘a’ is invalid.

2. The maximum length of a character constant can be 1 character.

Ex. ‘A’
‘I’
‘5’

Arithmetic expression:-

1. An expression is a combination of variable ,constant and operators written according to the syntax of c language.
2. In c every expression evaluates to a value

Ex. every expression result some values of a certain type that can be assign to a variable.
1. Some example of c expression is shown in the table given below.

Algebraic expression	C expression
A*B-C	A*B-C
(M+N)(X+Y)	(M+N)*(X+Y)
(AB/C)	A*B/C

4. OBSERVATIONS:- variables, constants, and expressions are studied properly.

EXPERIMENT NO:-4

1 .OBJECTIVE:-Program based on input/output statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: - source code

//Marks -Total, Grade

```
#include<stdio.h>
#include<conio.h>
Void main ( )
{
Float M, S, E, Per, tot;
clrscr( );
printf("\n Enter Maths Marks = ");
scanf("%f",&M);
printf("\n Enter Science Marks = ");
scanf("%f",&S);
printf("\n Enter English Marks = ");
scanf("%f",&E);
tot=M+S+E;
printf("\n Total = %f ",tot);
Per=tot/3;
if(Per>90)
printf("\n GRADE =  A ");
else
if(Per>=70)
printf("\n GRADE = B ");
else
if(Per>=50)
printf("\n PASSED ");
else
printf("\n FAILED ");
getch( );
}
```

5.PROGRAM INPUTS & OUTPUT:-

C-OUT/PUT	
Enter Maths Marks 23 Enter Science Marks. 33 Enter English Marks. 34 Total -90 Grade C	

6.OBSERVATIONS:-source codes are running successfully.

EXPERIMENT NO:-5

1. OBJECTIVE:-Program based on arithmetic expression.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-C++ software.

4. THEORY: - source code

```
// Calculate Simple interest
#include<stdio.h>
#include<conio.h>
void main( )
{
Int  p,r,t;
float  si;
Printf (“Enter no for p,r,t”);
Scanf(“%d%d%d”,&p,&r,&t);
Si=(p*r*t)/100;
Printf(“si=”,si);
getch( );
}
```

5.PROGRAM INPUTS & OUTPUT:-

C-OUT/PUT	
Enter Principal P=2000 Enter Rate R=12 Enter Time T=6 Simple Interest SI=1440.000000	

6.OBSERVATIONS:-source codes are running successfully.

EXPERIMENT NO:-7

1. OBJECTIVE:-Program based on goto statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:- c++ software.

4. THEORY: - source code

```
#include<stdio.h>
#include<conio.h>
Void main ( )
{
  Clrscr ();
Printf ("Hope is hoping against hope---");
   Goto hope I;
Printf ("even if it seems hopeless");
Hope:
Getch ();

}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
Hope is hoping against hope--		

6.OBSERVATIONS:-source codes are running successfully

EXPERIMENT NO:-8

1. OBJECTIVE:-Two programs based on 'if' and 'nested if'.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: - 8A> source code based on 'if'

```
// Check Number Is Even or Odd
#include<stdio.h>
#include<conio.h>
void main( )
{
int n;
clrscr( );
printf("\n Enter Number = ");
scanf("%d",&n);
if(n%2==0)
{
printf("\n Number Is Even  ",n);
}
else
printf("\n Number Is Odd  ",n);
getch( );
}
```

8B > source code based on 'if'

```
// Greater Three Numbers
#include<stdio.h>
#include<conio.h>
void main( )
{
int a,b,c;
clrscr( );
printf("\n Enter Number A = ");
scanf("%d",&a);
printf("\n Enter Number B = ");
scanf("%d",&b);
printf("\n Enter Number C = ");
scanf("%d",&c);
if(a>b&&a>c)
printf("\n A = %d is Greater",a);
else
if(b>c)
printf("\n B = %d is Greater",b);
```

```
else
printf("\n C = %d is Greater",c);
getch( );
}
```

8C> source code based on 'nested if

```
// Largest of Three Numbers.
#include<stdio.h>
#include<conio.h>
Void main ()
{
  Float A, B, C:
Clrscr ();
Printf ("enter three values\n=");
Scanf ("%f%f%f",&a&b&c);
Printf ("\n largest value is=");
   If (a>b)
{
  If (a>c)
Printf("%f\n", A);
else
Printf ("%f\n", c);
}
else
{
 If(C>B)
Printf ("%f\n, c);
else
Printf ("%f\n, B);
}
Getch ();
}
```

8D> source code based on 'nested if

```
//Marks -Total, Grade
#include<stdio.h>
#include<conio.h>
void main( )
{
float M,S,E,Per,tot;
clrscr( );
printf("\n Enter Maths Marks = ");
scanf("%f",&M);
printf("\n Enter Science Marks = ");
scanf("%f",&S);
printf("\n Enter English Marks = ");
scanf("%f",&E);
tot=M+S+E;
printf("\n Total = %f ",tot);
Per=tot/3;
if(Per>90)
```

```
printf("\n GRADE =  A ");
else
if(Per>=70)
printf("\n GRADE = B ");
else
if(Per>=50)
printf("\n PASSED ");
else
printf("\n FAILED ");
getch( );
}
```

5. **PROGRAM INPUTS & OUTPUT:-**

8A>// Check Number Is Even or Odd

C-OUT/PUT	
Enter number =12 Number is Even	

8B > // Greater Three Numbers

C-OUT/PUT	
Enter Number A=123 Enter Number B=100 Enter Number C=222 C=222 is grater	

8C> // Largest of Three Numbers

C-OUT/PUT	
Enter the Value =50 25 40 Largest Value is =50.000000	

8D>MarksTotal,Grade

C-OUT/PUT
Enter Math Marks=80 Enter Science Marks=75 Enter English Marks=86 Total =241.000000 GRADE =B

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-9

1. OBJECTIVE:-Program based on “switch case” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: - The control statement that allows us to make a decision from the number of choices is called switch

source code

```
//switch statement

#include<stdio.h>
#include<conio.h>
Void main ( )
{
  int i=22
Clrscr ();
Switch(i)
{
      Case 121:
       Printf(“I am in case 121\n”);
    Break;
    Case 7:
    Printf(“ I am in case7\n”);
   Break;
  Case 22:
  Printf(“ I am in case 22\n”);
  Break;
  Default:
  Printf(“I am in default\n”);
}
}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
I am in case 22		

6.

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-10A

1. OBJECTIVE:-Program based on “for” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3.SOFTWARE REQUIRED:-c++ software.

THEORY: - source code

```
// Sum N Numbers
#include<stdio.h>
#include<conio.h>
Void main ( )
{
Int i,n=0,y=0,z=0;
Clrscr ( );
Printf ("\n Enter Numbers for N = ");
scanf ("%d",&n);
for(i=1;i<=n;i++)
{
printf("\n Enter value = ");
scanf("%d",&y);
z = z + y;
}
printf("\n SUM N Numbers= %d ",z);
getch( );
}
```

4.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
Enter Numbers for N=2 Enter value =12 Enter value =45 SUM N Numbers =57		

5.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-10B

1. OBJECTIVE:-Program based on “while” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: - source code

```
// Factorial Using While Loop
#include<conio.h>
#include<stdio.h>
void main( )
{
float no,f=1;
clrscr( );
printf("\nEnter no = ");
scanf("%f",&no);
while(no>=1)
{
f=no*f;
no--;
}
printf("\nFactorial = %f ",f);
getch( );
}

}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
Enter no=4 Factorial =24.000000		

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-10C

1. OBJECTIVE:-Program based on “do-while” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: -Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. **SOFTWARE REQUIRED:-**c++ software.

4.THEORY: - source code for number generation

```
//do-while statement

#include<stdio.h>
#include<conio.h>
Void main ()
{
 Int i ;
Clrscr( ) ;

do

{
  Printf(“%d\n”, i);
i  = i+1;
}
While(i<=11);
getch( );

}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
1 2 3 4 5 6 7 8 9 10		

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-10D

1. OBJECTIVE:-Program based on “Break” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED: - Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: - We often come across situation where we want to jump out of a loop instantly, without waiting to get back to the conditional test. The keyword break allow us to do that .A break is usually associated with an if.

Source code for break statement

```
#include<stdio.h>
#include<conio.h>
Void main ()
{
Int i;
Clrscr ();
For(i=1;i<=10;i++)
{
If(i==5)
Break
Else
Printf(“%d\n”,i);
}
Printf(“hello”);
Getch();
}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
1 2 3 4 HELLO		

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-10E

1. OBJECTIVE:-Program based on “continue” statement.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY: In some programming situation, we want to take the control to the beginning of the loop, bypassing the statement inside the loop, which have not yet been executed. The keyword continues allows us to do this .When **continue** is encountered inside any loop, control automatically passes to the beginning of the loop.

Note- A **continue** is usually associated with an **if**.

Source code for continue statement

```
#include<stdio.h>
#include<conio.h>
Void main( )
{

   Int i ;
Clrscr() ;

For(i=1;i<=10;i++)
{
  If(i=5)
Continue;
Printf(“%d\n”,i);
}
Getch();
}
```

5.PROGRAM INPUTS & OUTPUT:-

	C-OUT/PUT	
1 2 3 4 5 6 7 8		

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-11

1. OBJECTIVE: - One program based on one dimensional array

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY:-.An array is a collection of similar elements. These similar elements could be all ints, or all floats, or all chars. etc

Program for sum and avg of numbers

```
/* Example Of Array
Size Of Array  - n[0] n[1] n[2] n[3] n[4]
Value Of Array -  10   20   30   40   50
Add Of Array   - 4000 4002 4004 4006 4008 */

#include<stdio.h>
#include<conio.h>
void main( )
{
float Avg,Sum=0;
int i,n[5];
clrscr( );
printf("\n Enter 5 Numbers = ");
for(i=0;i<=4;i++)
scanf("%d",&n[i]);
for(i=0;i<=4;i++)
Sum=Sum+n[i];
Avg=Sum/5;
printf("\n Sum = %f ",Sum);
printf("\n\n Avg = %f ",Avg);
getch( );
}
```

5. PROGRAM INPUTS & OUTPUT:-

C-OUT/PUT

```
Enter 5 Number =2
 3
4
5
7

Sum = 22.000000
Avg =4.400000
```

6. OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-12

1. OBJECTIVE: - One program based on two dimensional array

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY:-.An array is a collection of similar elements. These similar elements could be all ints, or all floats, or all chars. Etc

The two dimensional array is also called a matrix.

Source code:-

```
#include<stdio.h>
#include<conio.h>
Void main ( )
{
        Int s[4][3] ;
        Int I, j;
        Clrscr();
      For(i=0;i<= 3;i++)
{
      Printf (“\nenter roll number and marks=”);
      Scanf(“%d%d” ,&[i][0] , & s[i][1]);
}
      For(i=0;i<=3,i++)
      Printf(“\n %d%d” , s[i][0] ,s[i][1] );

}
```

5.PROGRAM INPUTS & OUTPUT:-

C-OUT/PUT

```
Enter Roll Number And Marks =1
420
Enter Roll Number And Marks =2
380
Enter Roll Number And Marks =3
250
Enter Roll Number And Marks =4
490
```

6.OBSERVATION: - source codes are running successfully.

EXPERIMENT NO:-13

1. OBJECTIVE: - Three programs based on string operation.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:- Intel P-III Processor and above
64 MB RAM and above
HDD 4.3 GB Hard Disc

3. SOFTWARE REQUIRED:-c++ software.

4. THEORY:-The way a group of integers can be stored in an integer array .
Similarly a group of characters can be stored in character array. Called string.

(A) String operations

```
#include<stdio.h>
#include<conio.h>
Void main( )
{
   Char name[25];
   Clrscr();
   Printf("enter name");
   Scanf("%s",name);
   Printf("%s" ,name);
   Getch( );
}
```

(B)String operations

```
#include<stdio.h>
#include<conio.h>
Void main( )

{
   Static char name[ ] = "Nagpur" ;
    Int i= 0;
  While (name[ i ] !=\0)
{
  Printf("%c" , name[ i ] );
  I++;
}
Getch( );


}
```

13c string operations

```
#include<stdio.h>
#include<conio.h>
Void main ( )
{
  Static char*name [ ] = {
                          "Nagpur"
                          "Agra"
                          "Delhi"
                          "culcutta"
                        }

Int i ;
Clrscr ( ) ;
For( i=0 ; i<=3 ;i++)
Printf("%s " , name[ i] );
  Getch( );

}
```

5.PROGRAM INPUTS & OUTPUT:-

(A)

C-OUT/PUT	
Enter Name: Mohan Mohan	

(B)

C-OUT/PUT	
Nagpur	

C-OUT/PUT	
NagpurAgraDelhi	

6. OBSERVATION: -All source codes are running successfully

EXPERIMENT NO:-14

1. OBJECTIVE: - Two programs based on functions.

2. HARDWARE & SYSTEM SOFTWARE REQUIRED:-

3. SOFTWARE REQUIRED:-

4. THEORY:-A function is a self-contained block of statement s that performs a coherent task of some kind. Every c program can be thought of as a collection of these functions.

14A Based on function

```
/* Call By Reference
   x = 20 y = 10
   a = 20 b = 10 */
#include<stdio.h>
#include<conio.h>
void main( )
{
int a=10,b=20;
clrscr( );
swapr(&a,&b);
printf("\n  a = %d b = %d",a,b);
getch( );
}
swapr(x,y)
int *x,*y;
{
int t;
t=*x;
*x=*y;
*y=t;
printf("\n *x = %d *y =% d",*x,*y);
getch( );
}
```

14B Based on function

```
/* Call By Value
   x=20 y=10
   a=10 b=20 */
#include<stdio.h>
#include<conio.h>
void main( )
{
int a=10,b=20;
clrscr( );
swapv(a,b);
printf("\n a = %d b = %d",a,b);
getch( );
}
swapv(x,y)
int x,y;
```

```
{
int t;
t=x;
x=y;
y=t;
printf("\n x = %d y =% d",x,y);
getch( );
}
```

5.PROGRAM INPUTS & OUTPUT:-

14A

c-out/put

```
X=20 Y =10
A=10 B= 20
```

14B

C-OUT/PUT

```
X=20 Y =10
A=10 B= 20
```

6. **OBSERVATION: -**All source codes are running successfully