Program 1(A):C program to display hello world.

AIM: Main objective of the C Program is to display Hello World message.

SOFTWARE USED: Visual Studio code using gcc compiler

CODE:

```
#include<stdio.h>
void main()
{
printf("1.Hello World!!!\n");
printf("2.Hello\nWorld!!!\n");
printf("3.Hello\n\tWorld!!!\n");
printf("4. World!!!\n");
printf("5.Helloworld!!!\n");
printf("6.\"Hello\nWorld!!!\"");
}
```

TEST CASES :

OUTPUT:

```
1.Hello World!!!
2.Hello World!!!
3.Hello World!!!
4.World!!!
5.HelloWorld!!!
6."Hello
   World!!!"
```

Program 1(B): C Program to scan all data type variables as input and print it as output.

AIM: The main objective of the C Program is to scan all data type variables as input and print it as output.

SOFTWARE USED: Visual studio code using gcc compiler

CODE:

```
#include<stdio.h>

void main()

{

char a;
 int b;

 float c;

 printf(“enter the char\n enter the integer \n

enter the float\n “,a,b,c);

 scanf(“%c%i%f”,&a,&b,&c);

 printf(“the char value read is %c \n the integer value read is %i
\n the float value read %f”,a,b,c);

}
```

TEST CASES AND OUTPUT:

S.No.	Input	Expected Output	Actual Output
1	A 10 3.14	The char value read is A The integer value read is 10 The float value read is 3.14	The char value read is A The integer value read is 10The float value read is 3.14
2	# 200 123.34546	The char value read is # The integer value read is 200 The float value read is 123.35	The char value read is # The integer value read is 200 The float value read is 123.35
3	2 123 56.32156	The char value read is 2 The integer value read is 123 The float value read is 56.32	The char value read is 2 The integer value read is 123 The float value read is 56.32

Program 1(C): C Program to perform arithmetic operators like +,-,*,/,% on two input variables

AIM: The main objective of the C Program is to perform arithmetic operators like, +, -, *, /, % on two input variables.

SOFTWARE USED: Visual studio code using gcc compiler

CODE:

```
#include<stdio.h>
void main()
 {
 int a,b,s1,s2,s3,s4,s5;
 printf("enter the two numbers \n");
 scanf("%d %d\n", &a ,&b);
  s1=a+b; s2=a-
  b; s3=a/b;
  s4=a*b; s5=a
  % b;
  printf("SUM=%i \n SUBTRACTION=%i \n
 DIVISION=%f\n MULTIPLICATION=%i \n  MODULUS
DIVISION=%i",s1,s2,s3,s4,s5);
}
```

TEST CASES AND OUTPUT:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1 .	**12 3**	The addition is 15 The difference is 9 The product is 36 The quotient is 4 The remainder is 0	SUM=15 SUBTRACTION=9 DIVISION=4.000000 MULTIPLICATION=36 MODULUS DIVISION=0
2 .	**1.2** **1.3**	The addition is 2.500000 The difference is -0.100000 The product is 1.560000. quotient is 0.923077. Modulo operator cannot be applied to float.	SUM=2.5 SUBTRACTION=-0.1 DIVISION=0.923077 MULTIPLICATION =1.56 MODULUS DIVISION=Cannot applied to float numbers
3 .	**a b**	The addition is 195 The difference is -1 The product is 9506 The quotient is 0 The remainder is 97	SUM=195 SUBTRACTION=-1 DIVISION=0 MULTIPLICATION=9506 MODULUS DIVISION=97

Program 1(D): C Program to perform temperature conversions from centigrade to Fahrenheit and vice versa

AIM: The main objective of the C Program is to perform temperature conversions from centigrade to Fahrenheit and vice versa.

SOFTWARE USED: Visual studio code using gcc compiler

CODE:

```
#include<stdio.h>

void main()

{

float F,C,X,Y;

printf(“enter the temperature in centigrade\n”,); scanf(“%f”, &C);

F=(C*9/5)+32;

printf(“ Converted Fahrenheit %f \n”,F);

printf(“enter the  temperature in fahrenheit \n”);

scanf(“%f”, &x);

Y=(5*(X-32)/9);

printf(“Converted Celsius %f”,Y);

}
```

TEST CASES AND OUTPUT:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	C=100 X=100	The temperature in Fahrenheit is 212.000000 F The temperature in Celsius is 37.777780 C	Converted Fahrenheit:212.000000 Converted Celsius:37.770000
2.	C=50 X=36	The temperature in Fahrenheit is 122.000000 F The temperature in Celsius is 2.222222 C	Converted Fahrenheit:122.000000 Converted Celsius:2.222222

PROGRAM 2.1

C program to scan an input and perform pre and post increment operations on it and display the result.

AIM:

The main objective of this program is to scan an input and perform pre and post increment operations

SOFTWARE USED:

Visual studio code

PROGRAM:

```
#include<studio.h>
Void main()
{
float a,b;
printf("enter any two values");
scanf("%f%f",&a&b);
printf("after post increment of 1st value  is %f",b++);
printf("after pre increment of 2nd value is%f",++a);
}
```

TEST CASES

S.NO	Output	Expected output	Actual output
1	1 2	After Post Increment of 1st Value 1 After Pre Increment of 2nd value 3	After Post Increment of 1st Value 1 After Pre Increment of 2nd value 3
2	1.1 2.1	After Post Increment of 1st Value 1.100000 After Pre Increment of 2nd value 3.100000	After Post Increment of 1st Value 1.100000 After Pre Increment of 2nd value 3.100000
3	2 3.4	After Post Increment of 1st Value 2 After Pre Increment of 2nd value 4.400000	After Post Increment of 1st Value 2 After Pre Increment of 2nd value 4.400000

2

2.2: C program to perform all bit wise operations.

Aim:

The main objective of this program is to make familiar with bitwise operators.

Software Used:

Visual Studio Code using Gcc compiler.

Code:

```
#include<stdio.h>
 Void main()
 {
      int a,b;
     printf("enter the values of a,b");
     scanf("%d%d",&a,&b);
     printf("using AND operator :%d%d\n",a&b);
    printf("using OR operator :%d%d\n",a|b);
    printf("using XOR operator :%d%d\n",a^b);
    printf("using COMPLEMENT operator :%d\n",~a);
    printf("using LEFT SHIFT operator :%d\n",a<<1);
    printf("using RIGHT SHIFT operator :%d\n",a>>1);
}
```

Test cases:

S.No	Input	Expected Output	Actual Output
1	5 6	The result of Bitwise AND is 4 The result of Bitwise OR is 7 The result of Bitwise XOR is 3 The result of Bitwise COMPLIMENT is -6	The result of Bitwise AND is 4 The result of Bitwise OR is 7 The result of Bitwise XOR is 3

		The result of LEFT SHIFT operator is 10 The result of RIGHT SHIFT operator is 2	The result of Bitwise COMPLIMENT is -6 The result of LEFT SHIFT operator is 10 The result of RIGHT SHIFT operator is 2
2	4 2	The result of Bitwise AND is 0 The result of Bitwise OR is 6 The result of Bitwise XOR is 6 The result of Bitwise COMPLIMENT is -5 The result of LEFT SHIFT operator is 8 The result of RIGHT SHIFT operator is 2	The result of Bitwise AND is 0 The result of Bitwise OR is 6 The result of Bitwise XOR is 6 The result of Bitwise COMPLIMENT is -5 The result of LEFT SHIFT operator is 8 The result of RIGHT SHIFT operator is 2
3	97 98	The result of Bitwise AND is 96 The result of Bitwise OR is 99 The result of Bitwise XOR is 3 The result of Bitwise COMPLIMENT is -98 The result of LEFT SHIFT operator is 194 The result of RIGHT SHIFT operator is 48	The result of Bitwise AND is 96 The result of Bitwise OR is 99 The result of Bitwise XOR is 3 The result of Bitwise COMPLIMENT is -98 The result of LEFT SHIFT operator is 194 The result of RIGHT SHIFT operator is 48

Output:

enter the values of a,b

5

6

using AND operator :4-938432031

using OR operator :7-938432608

using XOR operator :3-938432608

using COMPLEMENT operator :-6

using LEFT SHIFT operator :10

using RIGHT SHIFT operator :2

- **2.3:**

AIM : C program to extract last two digits of a given integer n ,where the last two digits of a given integer n , where the number of digits should be greater than 2.

SOFTWARE USED : Visual studio code using Gcc compiler

CODE :

```
#include<stdio.h>
Void main()
{
int n,r;
Printf("enter any number\n");
Scanf("%d",&n);
r=n%100;
if(r>9)
printf("The last two digits of given integer are %d",r);
else
Printf("The last two digits of given integer are 0 %d",r);
}
```

TEST CASES :

S.no	Input	Expected output	Actual output
1.	245	The last two digits of given integer are 45	The last two digits of given integer are 45
2.	001	The last two digits of given integer are 01	The last two digits of given integer are 01
3.	100	The last two digits of given integer are 00	The last two digits of given integer are 00

OUTPUT :

enter any number

245

The last two digits of given integer are 45

2

Program 2.4: C program to display greatest of three numbers using a conditional operator.

Aim:

The main objective of this program to make familiar with the usage of conditional operator.

Software used:

Visual studio code

Program:

```
#include<stdio.h>
void main()
{
    float a,b,c,big;
    printf("Enter the value of a:");
    scanf("%f",&a);
    printf("Enter the value of b:");
    scanf("%f",&b);
```

```
    scanf("%f",&c);
    big= ( (a>b) && (a>c) ) ? a : (b>c) ? b : c ;
    printf("The greatest of the three numbers is
    %f",big);
}
```

Test cases:

S.NO.	Input	Expected Output	Actual Output
1	6 4 5	The greatest of the three numbers is 6.000000	The greatest of the three numbers is 6.000000
2	12 1.5 13.5	The greatest of the three numbers is 13.500000	The greatest of the three numbers is 13.500000
3	11 11 11	The greatest of the three numbers is 11	The greatest of the three numbers is 11

Output:

```
Enter the value of a:6

Enter the value of b:4

Enter the value of c:5

The greatest of the three numbers is  6.000000
```

Program 2.5: C program to swap two numbers without using third variable.

Aim:

The main objective of this program to make familiar with interchanging values of two variables without using temporary variable.

Software used :

Visual studio code

Program:

```
#include<stdio.h>
void main()
{
    int a,b;
    printf("Enter the value of a:");
    scanf("%d",&a);
    printf("Enter the value of b:");
    scanf("%d",&b);
```

```
    printf("The values before swapping are %d
     %d\n",a,b);
    a=a+b;
    b=a-b;
    a=a-b;
    printf("The values after swapping are %d
    %d",a,b);
}
```

Test cases:

S.NO.	Input	Expected Output	Actual Output
1	10 15	The values before swapping are 10 15 The values after swaping are 15 10	The values before swapping are 10 15 The values after swaping are 15 10
2	16 14	The values before swapping are 16 14 The values after	The values before swapping are 16 14

		swaping are 14 16	The values after swaping are 14 16
3	A B	The values before swapping are A B The values after swaping are B A	The values before swapping are A B The values after swaping are B A

Output:

Enter the value of a:16

Enter the value of b: 14

The values before swapping are 16 14

The values after swapping are 14 16

Program 3(A): C Program to check whether a given integer is in between two values

AIM: To check whether a given input integer is in between two values.

SOFTWARE USED: Visual Studio code using gcc compiler.

CODE:

```
#include <stdio.h>
    void main()
   {
      int x,y,a;
     printf("enter the values\n");
      scanf("%d%d%d",&x,&y,&a);
     if(x<a && a<y)
    printf("%d is in between %d and %d\n",a,x,y);
    else
    printf("%d is not in between %d and %d\n",a,x,y);
  }
```

TEST CASES AND OUTPUT:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	-20 -10 -15	Key value is between the limits	-15 is in between -20 and -10
2	13 29 5	Key value is not in between the limits	5 is not in between 13 and 29

Program 3(B): C Program to check whether a given character is alphabet or a digit or a special symbol.

AIM: To check whether a given character is vowel or a constant or a digit or a special symbol

SOFTWARE REQUIRED: Visual Studio code using gcc compiler

CODE :

```
#include<stdio.h>

void main()
{
   int ASCII;

   char ch;

  printf(“enter any character\n”);

 scanf(“%c”,&ch);

   ASCII=ch;

  if ((ASCII>=65 && ASCII<=90) || (ASCII>=97 && ASCII<=122))

{
 printf(“entered character is an alphabet\n”,ASCII);
}
   else if(ASCII>=48 && ASCII<=57)
{
 printf(“entered character is an digit\n”,ASCII);
}
   else
{
   printf(“entered character is an special symbol\n”,ASCII);
}
}
```

TEST CASES AND OUTPUT:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	E	‘E’ is vowel	‘E’ is vowel
2.	P	‘P’ is constant	‘P’ is constant
3.	#	‘#’ is a special character	‘#’ is a special character

Program 3(C): C program to display the nature and roots of a quadratic equation.

AIM: The main objective of this program is to make the students use if else in solving the problem.

SOFTWARE USED: Visual Studio code using gcc.

CODE:

```
#include<stdio.h>
#include<math.h>
void main()
{
   int a,b,c;
   float root1,root2,realpart,imagpart;
   printf("Enter 'a' value :\nEnter 'b' value :\nEnter 'c' value :\n");
   scanf("%d %d %d",&a,&b,&c);
   if(b*b-4*a*c==0)
   printf("Roots are real and equal");
   else if(b*b-4*a*c>0)
   printf("Roots are real and distinct");
   else
   printf("Roots are imaginary");
   if(b*b-4*a*c>=0)
   {
   root1=(-b+sqrt(b*b-4*a*c))/2*a;
   root2=(-b-sqrt(b*b-4*a*c))/2*a;
```

```
printf("\n%f \t%f",root1,root2);
}
else
{
realpart = -b / (2 * a);
imagpart = sqrt((4*a*c)-b*b) / (2 * a);
printf("\nroot1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realpart, imagpart, realpart, imagpart);
}
```

TESTCASES AND OUTPUT :

s.no	Input	Expected output	Actual output
1.	1 4 4	Roots are real and equal. -2 -2	Roots are real and equal. -2 -2
2.	1 -5 6	Roots are real and distinct. 3 2	Roots are real and distinct. 3 2
3.	1 2 2	Roots are imaginary. root1=-1.00+1.00i and root2=-1.00-1.00i	Roots are imaginary. root1=-1.00+1.00i and root2=-1.00-1.00i

Program 3(D): C program to perform arithmetic operations using switch statement.

AIM: To perform arithmetic operations using switch statements.

SOFTWARE USED: Visual studio code using gcc compiler

CODE:

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int a,b,sum,sub,mul,div,rem;
  char ch;
  printf(“enter a, b values”);
  scanf(“%d%d”, &a, &b);
  printf(“enter the choice”);
  scanf(“%c”,&ch);
  switch(ch)
{
case ‘+’:{
  sum=a+b;
  printf(“sum=%d\n”,sum);
  break;
}
case ‘-’:{
  sub=a-b;
```

```
   printf(“sub=%d\n”,sub);
   break;
}
case ‘*’:{
   mul=a*b;
   printf(“mul =%d\n”, mul);
   break;
}
case ‘/’:{
  div=a/b;
   printf(“div =%d\n”, div);
   break;
}
case ‘%’:{
   rem=a%b;
   printf(“rem=%d\n”, rem);
   break;
}
}
default:
printf(“entered operator is not an arithmetic operator\n”);
}
```

TESTCASES:

S.no	Input	Expected output	Actual output
1.	10 20 *	Multiplication of 10 and 20 is : 200	Multiplication of 10 and 20 is : 200
2.	10 20 /	Division of 10 and 20 is : 0	Division of 10 and 20 is : 0
3.	10 20 %	Modulo division of 10 and 20 is : 10	Modulo division of 10 and 20 is : 10
4.	10 20 @	Please enter the correct operator	Entered operator is not an arithmetic operator

Program 3(E): C program to convert uppercase to lower case and lower case to upper case.

AIM: To convert uppercase to lower case characters and lower case to upper case characters.

SOFTWARE USED: Visual studio code using gcc compiler.

CODE:

```
#include <stdio.h>
int main()
{
int main()
char character;
printf(“enter a character=\t”);
scanf(“%c”,& character);
ASCII=character;
if(ASCII>65 && ASCII<90)
{
ASCII=character+32;
printf(“the converted lowercase character is %c\n”,ASCII);
}
else if(ASCII>97 && ASCII<122)
{
ASCII=character-32;
printf(“the converted uppercase character is %c\n”,ASCII);
}
```

```
else
printf(“entered character is not an alphabet”);
}
```

TESTCASES:

S.no	Input	Expected output	Actual output
1.	a	Uppercase is A	Uppercase is A
2.	@	Not an alphabet !!!	Not an alphabet !!!
3.	A	Lowercase is a	Lowercase is a

Program 4(A): C Program to print odd numbers between specified ranges.

AIM:

The main objective of this program is to print the odd numbers between the specified ranges

SOFTWARE USED: Visual Studio code using gcc compiler

CODE:

```
#include<stdio.h>

void main()

 {
    int a,b,c;
    printf("Enter the values of a and b: ");
    scanf("%d%d",&a,&b);
    for(c=a; c<b; c++)
    {
        if(c%2!=0)
        {
            printf("\n%d\t",c);
        }
    }
}
```

TESTCASES:

S. NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	10 30	11 13 15 17 19 21 23 25 27 29	11 13 15 17 19 21 23 25 27 29
2	-5 10	-3 -1 1 3 5 7 9	-3 -1 1 3 5 7 9
3	-30 -20	-29 -27 -25 -23 -21	-29 -27 -25 -23 -21

Program 4(B):C Program to display the factors of a given number and check whether it is a prime or not.

AIM:

The main objective of this program is to display the factors of a given program and check whether the number is prime or not.

SOFTWARE USED: Visual studio code using gcc compiler.

CODE:

```
#include<stdio.h>
void main()
{
    int x,i,count;
    printf("enter the values of x:");
    scanf("%d",&x);
    count=0;
    if(x>0)
    {
        printf("factors are :\n");
        for(i=1;i<=x;i++)
        {
            if(x%i==0)
            {
                printf("%d,",i);
                count++;
            }
        }
        if(count==2)
        {
            printf("\n %d is a prime number",x);
        }
        else
        {
            printf("\n%d is not a prime number",x);
        }
    }
    else
    {
```

```
        printf("invalid input");
    }
}
```

TEST CASES:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	8	Factors are: 1,2,4,8 8 is not a prime number	Factors are: 1,2,4,8 8 is not a prime number
2.	-9	Invalid input number	Invalid input number
3.	89	Factors are: 1,89 89 is a prime number	Factors are: 1,89 89 is a prime number

Program 4(C): C Program to display the sum of individual digits of a given integer raised to the power of n. Also check whether the given integer is Armstrong or not.

AIM:

The main objective of this program is to display the sum of individual digits of a given integer raised to the power of n. Also check whether the given integer is Armstrong or not.

SOFTWARE USED: Visual studio code using gcc compiler

CODE:

```
#include<stdio.h>
void main()
{
    int num,b,rem,sum=0,temp;
    printf("Enter the numbers: ");
    scanf("%d%d",&num,&b);
    temp=num;
    while(num!=0)
    {
        rem=rem%10;
        sum=sum+(rem*rem*rem);
        num=num/10;
    }
    printf("\nSum of individual digits raised to the power of %d is
%d\n",b,sum);
    if(temp==sum)
    printf("%d is an armstrong number\n",temp);
    else
    printf("%d is not an armstrong number\n",temp);
}
```

TEST CASES:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	371 3	Sum of individual digits raised to the power of 3 is 371. 371 is an Armstrong number	Sum of individual digits raised to the power of 3 is 371. 371 is an Armstrong number
2	9474 4	Sum of individual digits raised to the power of 4 is 9474. 9474 is an Armstrong number	Sum of individual digits raised to the power of 4 is 9474. 9474 is an Armstrong number.
3	150 4	Sum of individual digits raised to the power of 4 is 626. 150 is not an Armstrong number	Sum of individual digits raised to the power of 4 is 626. 150 is not an Armstrong number

Program 4(D):C Program to demonstrate the usage of unconditional control statements

AIM: The main objective of this program is to demonstrate the usage of unconditional control statements.

SOFTWARE USED: Visual studio code using gcc compiler.

CODE:

```
#include<stdio.h>
int main()
{
    int a,b,i,x;
    printf("enter the values of a and b:");
    scanf("%d%d",&a,&b);
    printf(“\nusage of break statement\n”);
    for(i=0; i<=a; i++)
    {
        if(i==b)
        break;
    }
    printf("%d\n",i);
    printf(“usage of continue statement\n”);
    for(i=0; i<a; i++)
    {
       if(i==b)
       continue;
       printf("%d\n",i);
    }
    printf(“usage of goto statement\n ”);
    for(i=1; i<a; i++)
    {
        goto label;
        printf("%d\n",i);
    }
    label:
    return 0;
}
```

TEST CASES:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1 .	10 4	Usage of break statement 0 1 2 3 Usage of continue statement 0 1 2 3 5 6 7 8 9 Usage of goto statement 0 1 2 3 loop break	Usage of break statement 0 1 2 3 Usage of continue statement 0 1 2 3 5 6 7 8 9 Usage of goto statement 0 1 2 3 loop break
2 .	15 6	Usage of break statement 0 1 2 3 4 5 Usage of continue statement 0 1 2 3 4 5 7 8 9 10 11 12 13 14 Usage of goto statement 0 1 2 3 4 5	Usage of break statement 0 1 2 3 4 5 Usage of continue statement 0 1 2 3 4 5 7 8 9 10 11 12 13 14 Usage of goto statement 0 1 2 3 4 5
3 .	5 2	Usage of break statement 0 1 Usage of continue statement 0 1 3 4 Usage of goto statement 0 1 .	Usage of break statement 0 1 Usage of continue statement 0 1 3 4 Usage of goto statement 0 1.

Program 4(E):C Program to display the following pattern

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

AIM: The main objective of this program is to display the following pattern.

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

SOFTWARE USED: Visual studio code using gcc compiler.

CODE:

```
#include <stdio.h>
int main()
{
    int r,c,i,j;
    printf("enter the number of rows:");
    scanf("%d",&r);
    printf(“\n”);
    for(i=r; i>=1; i--)
    {
        for(j=r; j>=i; j--)
        {
            printf(" ");
        }
        for(c=i; c>=1; c--)
        printf("%d ",c);
        printf("\n");
    }
    return 0;
}
```

TESTCASES:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	5	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1
2.	5	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1
3.	5	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1	5 4 3 2 1 4 3 2 1 3 2 1 2 1 1

Program 5(A): C Program to demonstrate the various categories of functions with respect to return type and number of arguments.

AIM: The main objective of this program is to make familiar with categories of functions.

SOFTWARE USED: Visual Studio code using gcc compiler

CODE:

```
 #include<stdio.h>
 void sum_1();
  int sum_2();
  void sum_3(int,int);
  int sum_4(int,int);
  void main()
   {
        int A,B,a,b,p=-5,q=6,r=-3,s=-7;
        sum_1();
        A=sum_2();
              printf("The sum of a and b with no arguments
                        and return type is %d\n",A);
        sum_3(p,q);
        B=sum_4(r,s);
       printf("The sum of a and b with arguments and return type
                                    is %d\n",B);
}
 void sum_1()
 {
        int add,a=2,b=3;
        add=a+b;
              printf("The sum of a and b with no arguements
                    and no return type is %d\n",add);

 }
   int sum_2()
 {
```

```
        int a=5,b=6,add;
        add=a+b;
        return(add);
}
 void sum_3(int x,int y)
{
        int add;
        add=x+y;
         printf("The sum of a and b with arguments and no return
                                    type is %d\n",add);
}
   int sum_4(int x,int y)
{
        int add;
        add=x+y;
        return(add);
}
```

TEST CASES AND OUTPUT:

S.NO.	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	2 3	The sum of num1 and num2 with no arguments and no return type is 5	The sum of num1 and num2 with no arguments and no return type is 5
2.	5 6	The sum of num1 and num2 with no arguments and return type is 11	The sum of num1 and num2 with no arguments and return type is 11
3.	-5 6	The sum of num1 and num2 with arguments and no return type is 1	The sum of num1 and num2 with arguments and no return type is 1
4.	-3 -7	The sum of num1 and num2 with arguments and return type is -10	The sum of num1 and num2 with arguments and return type is -10

PROGRAM 5(B): C PROGRAM TO FIND THE LCM OF TWO NUMBERS USING FUNCTIONS.

AIM: The main objective of this program is to make familiar with to find the LCM of two numbers.

SOFTWARE USED: Visual Studio code using gcc compiler.

CODE:

```
#include<stdio.h> int
lcm(int,int);
int main()
{
int a,b,n;
printf(“enter the two numbers \n”);
scanf(“%d %d “,&a,&b);
n=lcm(a,b);
printf(“The lcm of %d and %d is %d“,a,b,n);
}
int lcm(int a, int b)
{
int i;
for(i=2; i<(a*b); i++)
{
```

```
 if(i%a == 0 && i%b == 0) {
return i;
break;
}
else {
printf("LCM is undefined\n");
break;
}
 }
```

TEST CASES AND OUTPUT:

S.NO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1.	9 24	The LCM of 9 and 24 is 72.	The LCM of 9 and 24 is 72.
2.	5 7	The LCM of 5 and 7 is 35.	The LCM of 5 and 7 is 35.
3.	4 0	LCM is Undefined	LCM is Undefined

Program 5(C): Create header file which contains the following prototype:

i) int factorial(int); // non-recursive function

ii) int factorial_rec(int); //Recursive function

iii) int prime(int);

Use the above functions in a C program by including the above header file.

AIM:

The main objective of this program is to create a user-defined header file containing prototypes: i. int factorial(int) ; // non-recursive function ii.int factorial_rec(int); //Recursive function

iii. int prime(int);

To use them in a C program by including the header file.

SOFTWARE USED: Visual Studio Code using gcc compiler.

CODE:

functions.h //header file

```
int factorial(int n)
{   int f=1,i;
for(i=n;i>=1;i--
)   f=f*i;
return f;
}
int factorial_rec(int n)
```

```
{
   int fact=1;
if(n==1)    return 1;
else
fact=n*factorial_rec(
n-1);     return fact;
}
 int prime(int n)
{
   int i,flag=0;
for(i=2;i<=n/2;i+
+)    if(n%i==0)
flag=1;    return
flag;
 }
```

week5C.c //Main Program

```
#include<stdio.h>
#include"header.h”
 void main()
{
 int n,p;
printf("enter a number \n");    scanf("%d",&n);
printf("The factorial of %d without recursion is
%d\n",n,factorial(n));    printf("The factorial of %d
with recursion is %d\n",n,factorial_rec(n));
p=prime(n);
```

```
if(p==0)
printf("%d is a prime number\n",n);
else
printf("%d is not a prime
number\n",n);
}
```

TEST CASES AND OUTPUT:

S. No	Input	Expected Output	Actual Output
1	5	The factorial of 5 without recursion is 120 The factorial of 5 with recursion is 120 5 is a prime number.	The factorial of 5 without recursion is 120 The factorial of 5 with recursion is 120 5 is a prime number.
2	7	The factorial of 7 without recursion is 5040 The factorial of 7 with recursion is 5040 7 is a prime number.	The factorial of 7 without recursion is 5040 The factorial of 7 with recursion is 5040 7 is a prime number.
3	8	The factorial of 8 without recursion is 40320 The factorial of 8 with recursion is 40320 8 is not a prime number.	The factorial of 8 without recursion is 40320 The factorial of 8 with recursion is 40320 8 is not a prime number.

Program 5(D): C program to display Pascal’s triangle using functions.

AIM: The main objective of the C program is to display Pascal's triangle using functions.

SOFTWARE USED: Visual Studio using gcc compiler

CODE:

```
#include <stdio.h>
void pascal(int);
void main(){
int row;
printf(“enter the rows\n”);
scanf(“%d”, &row);
pascal(row);
}
void pascal(int x){
int sp,i,j,x=1;
for(i=0; i<n; i++){
for(sp=1; sp<n-i; sp--)
printf(“  ”);
for(j=0; j<=i; j++){
if(i==0 || j==0)
x=1;
else
x=x*(i-j+1)/j;
```

```
printf("%d",x);
}
}
```

TEST CASES:

S.No.	Input	Expected Output	Actual output
1	4	1 1 1 1 2 1 1 3 3 1	1 1 1 1 2 1 1 3 3 1
2	5	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1
3	7	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1	1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1

Program-6(A): C program to read n integer values into an array and display them.

AIM: The main objective of this program is to make familiar with accessing array elements.

SOFTWARE USED: Visual Studio code using GCC compiler.

CODE:

```
#include<stdio.h>
void main()
 {
     int arr[100],i,n;
     printf("enter number of values in array:");
     scanf("%d",&n);
     printf("enter values into array:");
     for(i=0;i<n;i++)
      {
       scanf("%d",&arr[i]);
       printf("values into array are:");
      }
     for(i=0;i<n;i++)
      {
       printf("%d\n",arr[i]);
      }
 }
```

TEST CASES:

SNo	Input	Expected Output	Actual Output
1	5 10 20 30 40 50	The elements in the array are 10 20 30 40 50(with single space)	The elements in the array are 10 20 30 40 50(with single space)
2	4 10 -20 30 -40	The elements in the array are 10 -20 30 -40 (with tab spaces)	The elements in the array are 10 -20 30 -40 (with tab spaces)

Program-6(B): C program to count and display the number of positive, negative, even and odd numbers in a given array of integers and display their sum.

AIM: The main objective of this program is to make familiar with accessing array elements.

SOFTWARE USED: Visual Studio code using GCC compiler.

CODE:

```
#include<stdio.h>
void main()
 {
     int n = 5,a[n],i,pos=0,neg=0,even=0,odd=0;
     printf("enter %d integer numbers\n",n);
     for(i=0;i<n;i++)
     {
         scanf("%d",&a[i]);
         if(a[i]>0)
           pos++;
         elseif(a[i]<0)
            neg++;
         elseif(a[i]%2==0)
            even++;
         else
            odd++;
     }
     printf("\n positive : %d\n",pos);
     printf("\n negative : %d\n",neg);
     printf("\n even : %d\n",even);
     printf("\n odd : %d\n",odd);

 }
```

TEST CASES:

SNo	Input	Expected Output	Actual Output
1	4 0 0 3 -2	The positive numbers are 1 The negative numbers are 1 The odd numbers are 1 The even numbers are 3 0 is neither positive nor negative The sum is 1	The positive numbers are 1 The negative numbers are 1 The odd numbers are 1 The even numbers are 3 0 is neither positive nor negative The sum is 1
2	5 1 2 3 4 5	The positive numbers are 5 The negative numbers are 0 The odd numbers are 3 The even numbers are 2 The sum is 15	The positive numbers are 5 The negative numbers are 0 The odd numbers are 3 The even numbers are 2 The sum is 15

Program-6(C): C program to find the smallest and largest numbers in an array of integers.

AIM: The main objective of this program is to make familiar with accessing array elements.

SOFTWARE USED: Visual Studio code using Gcc compiler

CODE:

```
#include<stdio.h>

 int main()
{

   int a[50], size, i, big, small;
   printf("\nEnter the size of the array: ");
   scanf("%d", &size);
   printf("Enter the %d elements of the array:",
size);
    for(i=0;i<size;i++)
    scanf("%d",&a[i]);
   big=a[0];
    for(i=1;i<size;i++)

    {
        if(big<a[i])           {
            big=a[i];

    }
    printf("The largest element is: %d", big);
   small=a[0];
    for(i=1;i<size;i++)

    {
        if(small>a[i])
            small=a[i];
    }
  printf("The smallest element is: %d", small);

  return 0;

}
```

TEST CASES:

S. No	input	Expected Output	Actual Output
1	5 10 5 20 13 4	The smallest element is 4 The biggest element is 20	The smallest element is 4 The biggest element is 20
2	6 -5 -4 -6 -8 -10 -20	The smallest element is -20The biggest element is -4	The smallest element is -20 The biggest element is -4
3	5 10 -2 -20 5 -3	The smallest element is -20The biggest element is 10	The smallest element is -20 The biggest element is 10

Program-6(D): C program to perform addition, multiplication, transpose of given matrices using functions.

AIM: The main objective of this program is to make familiar with accessing array elements.

SOFTWARE USED: Visual studio code using Gcc compiler

CODE:

```
#include <stdio.h>
int r,c,r1,c1, a[100][100], b[100][100],
sum[100][100],mul[100][100],i,j,k;
int add(){
   // adding two matrices
   if(r==r1&&c==c1){
   for (i = 0; i < r; ++i)
   for (j = 0; j < c; ++j) {
       sum[i][j] = a[i][j] + b[i][j];
     }
   printf("\nSum of two matrices: \n");
   for (i = 0; i < r; ++i)
   for (j = 0; j < c; ++j) {
   printf("%d ", sum[i][j]);
    }
   printf("\n\n");

}

else
```

```
printf("Addition not possible");
}
int prod(){
    //mutiplying two matrices
    if(c==r1){
    for (i=0;i<r;i++)
    for (j=0;j<=c1;j++){
    for(k=0;k<=c1;k++)
    mul[i][j]+=a[i][k]*b[k][j];
 }
   printf("\nmultiplication of two matrices: \n");
 for (i = 0; i < r; ++i){
   for (j = 0; j < c1; ++j) {
   printf("%d ", mul[i][j]);
  }
   printf("\n\n");
  }
}
else
printf("Multiplication not possible");
}
int tran(){
printf("Transpose of first matrix is\n");
  for (i=0;i<r;i++){
    for (j=0;j<c;j++){
 printf("%d  ",a[j][i]);
      }
        printf("\n\n");
```

```
    }
    printf("Transpose of second matrix is\n");
  for (i=0;i<r;i++){
    for (j=0;j<c;j++){
       printf("%d  ",b[j][i]);
      }
        printf("\n\n");
    }
int main() {
 printf("Enter the number of rows: ");
 scanf("%d", &r);
 printf("Enter the number of columns: ");
scanf("%d", &c);
 printf("Enter the number of rows in 2nd matrix: ");
 scanf("%d", &r1);
 printf("Enter the number of columns in 2nd matrix:
");
scanf("%d", &c1);
 printf("\nEnter elements of 1st matrix:\n");
 for (i = 0; i < r; ++i)
 for (j = 0; j < c; ++j) {
 printf("Enter element a%d%d: ", i + 1, j + 1);
scanf("%d", &a[i][j]);
 }
printf("Enter elements of 2nd matrix:\n");
for (i = 0; i < r1; ++i)
 for (j = 0; j < c1; ++j) {
 printf("Enter element b%d%d: ", i + 1, j + 1);
```

```
scanf("%d", &b[i][j]);
 }
 add();
prod();
tran();
return 0;
}
```

TEST CASES:

S NO	Input	Expected output	Actual Output
1	3 3 1 0 0 0 1 0 0 0 1 3 2 1 2 3 4 5 6	Addition is not possible The multiplication is 1 2 3 4 5 6 The Transpose of second matrix is 1 4 2 5 3 6	Addition is not possible The multiplication is 1 2 3 4 5 6 The Transpose of second matrix is 1 4 2 5 3 6
2	2 2 1 0 0 1 3 3 1 2 3 4 5 6 7 8 9	Addition is not possible Multiplication is not possible Transpose of first matrix is 1 0 0 1	Addition is not possible Multiplication is not possible Transpose of first matrix is 1 0 0 1

Program-6(E): C program to check whether a given integer exists in a list of numbers and print its index value if it is present, otherwise print “No”.

AIM: C Program to check whether a given integer exists is a list of numbers and print its index value if it is present otherwise print “NO”**.**

SOFTWARE USED: Visual studio code using Gcc compiler

CODE:

```
#include <stdio.h>
 int main()
  {
   int a[100],i,n,k;

    printf("Enter size of the  array : ");
    scanf("%d", &n);
    printf("Enter elements in array:\n");
    for(i=0; i<n; i++){
      scanf("%d",&a[i]);
    }
     printf("Enter element to search : ");
     scanf("%d", &k);
     for(i=0; i<n; i++)
    {
        if(a[i]==k)
         printf("Element found at %d position",i);
        else
         printf("Element  not  found");
    }
return 0;

 }
```

TEST CASES:

SNo	Input	Expected output	Actual Output
1	5 10 3 -4 15 6 15	Element found at 4 position	Element found at 4 position
2	5 10 3 -4 15 6 20	Element not found	Element not found
3	5 3 2 5 2 7 2	Element found at 2 position Element found at 4 position	Element found at 2 position Element found at 4 position

Program 7(A): C program to convert upper case character to lowercase and vice versa in a given string.

AIM: Convert upper case character to lowercase and vice versa in a given string.

SOFT WARE REQUIRED: Visual studio code

PROGRAM:

```
#include<stdio.h>
#include<string.h>
Void main()
{
      char name[20];
      Int i,l;
      Printf(“Enter string \n”)
      gets(name);
      l=strlen(name);
     {
          if(name[i]>=65 && name[ i ]<=90)
         name[i]=name[i]+32;
         else if(name[i]==97 && name[ i ]<=122)
         name[i]=name[i]-32;
       }
      Printf(“the string is %s “,name);
}
```

TEST CASES

INPUT	EXPECTED OUPUT	ACTUAL OUTPUT
Hello GVP	hELLO gvp	hELLO gvp
Welcome iNdiA	wELCOME InDIa	wELCOME InDIa
aDITYA 369	Aditya 369	Aditya 369
123	123	123

Program 7(B):C Program to delete all vowels in a given string and display the remaining string

AIM: Delete all vowels in a given string and display the

Remaining string

SOFTWARE REQUIRED : visual studio code

CODE :

```
#include<stdio.h>
 #include<string.h>
 void main()
  {
     char name[50];
      int l,j,l,t,n,c;
      printf(“please enter the name in lower case:\n”);
      scanf(“%s”,&name);
      l=strlen(name);
      printf(“%d”,l);
      for(i=0;i<l;i++)
     {
        n=name[i];
      if(n==65|| n==69|| n==73|| n==79|| n==85|| n==97|| n==101|| n==105|| n==111|| n==117)

{
for(j=i;j<l;j++)
```

```
{
name[I]=name[j+1];
c=name[l-1];
If(c==65|| c==69|| c==73|| c==79|| c==85|| c==97|| c==101||
c==105|| c==111|| c==117||)
name[l-1]=32;
   }
   i=i-1;
   l=l-1;
    }
 }
  printf("%s",name);
}
```

TEST CASES:

INPUT	**EXPECTED OUTPUT**	**ACTUAL OUTPUT**
Programming in c	Prgrmmng n c	Prgrmmng n c
algorithm	lgrthm	lgrthm
gvpce	gvpc	gvpc

Program 7(C): C program to check whether a given string is palindrome or not.

Aim: The main objective of this program is to make familiar with reversing agiven string, comparing string

Software Used: Visual Studio code using GCC compiler

Code :

```
#include<stdio.h>
#include<string.h>
void main()
{
 int I,l,a=0; char str[100];
 printf(“enter a string”); scanf(“%s”,str); l=strlen(str);
 for(i=0; i<1;i++);
 {

  if (str[i]!=str[l-i-1])
  {
    a=1;
    break;
  }
 }
if(a==0)
Printf(“%s is a palindrome”,str);
else
```

```
printf("%s is not a palindrome",str);
}
```

TEST CASES:

s.no	Input	Expected Output	Actual Output
1	MalayalM	The given string is palindrome	MalayalM is a palindrome
2	collect	The given string is not palindrome	Collect is not palindrome
3		The given string is palindrome	is a palindrome
4	\o	The given string is not a palindrome	\0 is not a palindrome

Program 7(D): Program that reads two integers as strings and display their sum.

Aim: C Program that reads two integers as strings and display their sum.

Software used: VISUAL STUDIOS (gcc complier)

Code:

```
 #include<stdio.h>
  #include<string.h>
  void main()
   {
      char name[50];
       int l,j,l,t,n,c;
       printf(“please enter the name in lower case:\n”);
       scanf(“%s”,&name);
       l=strlen(name);
       printf(“%d”,l);
       for(i=0;i<l;i++)
      {
          n=name[i];
       if(n==65|| n==69|| n==73|| n==79|| n==85||
n==97||n==101|| n==105|| n==111|| n==117||)
{
for(j=i;j<l;j++)
```

```
{
name[I]=name[j+1];
c=name[l-1];
If(c==65|| c==69|| c==73|| c==79|| c==85|| c==97|| c==101|| c==105|| c==111|| c==117||)
name[l-1]=32;
   }
   i=i-1;
   l=l-1;
    }
 }
  printf("%s",name);
}
```

TEST CASE:

S.NO.	INPUT	Expected output	Actual Output
1	123 37	The sum of two given strings 123 and 37 is 160	The sum of two given strings123 and 37 is 160
2	123 \0	Enter correct input	Enter correct input
3	154 $	Enter correct input	Enter correct input
4	123	The sum of two given string 123 and is 123	the sum of two strings 123 and is 123.

Program-8(A):- C Program to demonstrate the usage of at least 10 predefined string handling functions.

Aim:

Main objective of this program is to make familiar with predefined string handling functions.

Software used:

Visual studio code

Program:

```
#include<stdio.h>
#include<string.h>
void main()
{
   char s1[50],s2[50],s3[50],s4[50],s5[50],s6[50],s7[50];
   int l,k;
      printf("enter first string for string length,string comparision,to
            convert into string upper:\n");
   gets(s1);
     printf(“enter second string for string length,string
           comparision,to convert into string upper:\n”);
   gets(s2);
     printf(“enter string in upper case character to convert into lower
           case:\n”);
```

```
 gets(s3);
    printf("enter first string for string contanetion:\n");
  gets(s4);
    printf("enter second string for string contanation:\n");
  gets(s5);
    printf("enter first string for string copy:\n");
  gets(s6);
    printf("enter second string for string copy:\n");
  gets(s7);
  l=strlen(s1);
    printf( "string length:%d",l);
  k=strcmp(s1,s2);
    print("string comparision:%d",k);
  strupr(s1);
    printf("string upper case:%s",s1);
  strlwr(s3);
    printf("string lower case:%s",s3);
  strrev(s2);
    printf("string reverse:%s",s2);
  strcat(s4,s5);
    printf("string cantenation:%s",s4);
  strcpy(s6,s7);
    printf("string copy :%s",s6);
}
```

output:

enter first string for string length,stringcomparision,to convert into string upper :

program

enter second string for string length,stringcomparision,to convert into string upper :

computer

enter string in upper case character to convert into lower case :

COLLEGE

enter first string for string contanetion:

tony

enter second string for string contanation:

stark

enter first string for string copy:

engineering

enter first string for string copy:

gayatri

string length:7

string comparision:1

string upper case: PROGRAM

string lower case: college

string reverse:retupmoc

string cantenation:tonystark

string copy: gayatri

Program-8(B): C program that implements the following user defined string handling functions.

1. To find the length of the given string
2. To copy the contents of one string to another
3. To reverse the contents of a string
4. To compare two strings
5. To concatenate two strings

Aim:

The main objective of this program is to make familiar with user-defined string handling functions.

Software used:

Visual studio code.

1. To find the length of the given string

Program:

```
#include<stdio.h>
#include<string.h>
  Void  main()
{
      Char alpha[100];
       Int i=0;
       Printf("Enter the string:");
       Gets(alpha);
       While(alpha[i]!='\0')
       {
            i++;
       }
       Printf("The length of the string is :%d \n",i);
}
```

Output:

```
Enter the string: ARIES
The length of the string is: 5
```

2. To copy the contents of one string to another

Program:

```
#include<stdio.h>
#include<strig.h>
  void  main()
{
      char alpha[100],beta[100];
      int i=0;
      printf("Enter the string:  ");
      gets(alpha);
      while(alpha[i]!='\0')
      {
          beta[i]=alpha[i];
           i++;
      }
     beta[i]='\0';
    printf("The given string  is  '%s'",alpha);
   printf("The string copied is '%s'",beta);
}
```

Output:

```
Enter the string:Aries
The given string  is  'Aries'
The string copied is 'Aries'
```

3. To reverse the contents of a string

Program:

```
#include<stdio.h>
#include<string.h>
 void main()
{
      char alpha[100],temp;
      int i,l;
      printf(“Enter string : ”);
      scanf(“%s”,alpha);
      l=strlen(alpha)-1;
      for(i=0;i<strlen(alpha)/2;i++)
      {
          temp =  alpha[i];
          alpha[i] = alpha[l-i];
          alpha[l-i] = temp;
      }
      printf(“\n Reversed string is : %s”,alpha);
}
```

Output:

```
Enter string : ARIES
Reversed string is :  SEIRA
```

4. To compare two strings

Program:

```
#include<stdio.h>
#include<string.h>
  void  main()
{
      char  alpha[100],beta[100];
      int i=0,count=0,l1,l2;
      printf("Enter the string : ");
      gets(alpha);
      printf("Enter the string : ");
      gets(beta);
      l1=strlen(alpha);
      l2=strlen(beta);
      if(l1==l2)
        {
          for(i=0;i<l1;i++)
          {
              If(alpha[i]==beta[i])
              count++;
           }
          If(count>0)
             printf("The given strings are same.");
```

```
            else
               printf("The given strings are not same.");
            }
    else
      printf("The given strings are not same.");
}
```

Output:

Enter the string : ARIES

Enter the string : ZEUS

The given strings are not same

5. To concatenate two strings

Program:

```
#include<stdio.h>
#include<string.h>
void  main()
{
      char   alpha[100],beta[100];
      int l1,l2,i=0;
      printf(“enter the string : ”);
      puts(alpha);
      printf(“enter the string : ”);
      puts(beta);
      l1=strlen(alpha);
      l2=strlen(beta);
      for(i=0;i<=l2;i++)
        {
            Alpha[l1+i]=beta[i];
         }
       Printf(“%s”,alpha);
}
```

TESTCASES:

S. No	Input	Expected Output	Actual Output
1.	GVP College	String length is 11	The string length is: 11
2.	GVP College	String copied is GVP College	The string copied is ‘GVP College’
3.	GVP College	String reversed is egelloC PVG	Reversed string is egelloC GVP
4.	GVP College Gvp College	Strings are not same	The given strings are not same
5.	Gvp College	Gvp College	Gvp College

Program 9(A): C program to demonstrate the usage of pointers.

AIM :

The main objective of this program is to make familiar with the usage of pointers.

Software used :

Visual studio code.

Program :

```
#include<stdio.h>
void main()
{
   int a=5;
   int *p;
   p=&a;
   printf("value of a : %d\n",a);
   printf("value of a : %d\n",*p);
   printf("Address of a :%u\n",&a);
   printf("Address of a in hexadecimal form:%x\n",&a);
   printf("Address of a :%u\n",p);
   printf("Address of a in hexadecimal form:%x\n",p);
}
```

Output :

value of a : 5

value of a : 5

Address of a : 1780408940

Address of a in hexadecimal form : 6a1ee26c

Address of a : 1780408940

Address of a in hexadecimal form : 6a1ee26c

Program 9(B) : C program that uses dynamic memory allocation functions to add n elements and display their average.

Aim : The main objective of this program is to make familiar with the usage of dynamic memory allocating functions.

Software used : Visual studio code.

Program :

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
   int *p,i,n;
   float avg,sum=0;
   printf("Enter n value\n");
   scanf("%d",&n);
   p=malloc(n);
   if(p!=NULL)
  {
      printf("Enter the values\n");
      for(i=0;i<n;i++)
      {
         scanf("%d",&*(p+i));
         sum=sum+(*(p+i));
      }
```

```
        for(i=0;i<n;i++)
        {
           printf("The entered value = %d\n",*(p+i));
        }
        printf("The sum = %f\n",sum);
        avg=sum/n;
        printf("The average = %f",avg);
}
  else
  printf("The memory is not allocated");
  free(p);
}
```

Output :

```
Enter n value : 4
Enter the values : 1 2 3 4
The entered value =1
The entered value =2
The entered value =3
The entered value =4
The sum=10.000000
The average=2.500000
```

Program 9(C) : C program that performs pointer arithmetic.

Aim : The main objective of this program is to make familiar with the pointer arithmetic.

Software used : Visual studio code.

Program :

```
#include<stdio.h>
void main()
{
   int a[]={15,16,17,18,19};
   int *p,*q;
   p=a;
   q=&a[3];
   printf("p-q is %d\n",p-q);
   printf("q-p is %d\n",q-p);
   printf(" value at p+3 is %d\n",*(p+3));
   printf(" value at p-3 is %d\n",*(p-3));
   printf(" value at q-3 is %d\n",*(q+3));
   printf(" value at q-3 is %d\n",*(q-3));
   printf(" Address of p+3 is %x\n",(p+3));
   printf(" Address of p-3 is %x\n",(p-3));
   printf(" Address of q-3 is %x\n",(q+3));
   printf(" Address of q-3 is %x\n",(q-3));
   printf("*p++ = %d\n",*p++);
```

```
    printf("p++ = %x\n",p++);
    printf("*++p = %d\n",*++p);
    printf("++p = %x\n",++p);
    printf("*q-- = %d\n",*q--);
    printf("q-- = %x\n",q--);
    printf("*--q = %d\n",*--q);
    printf("--q = %x\n",--q);
}
```

Output :

p-q is -3

q-p is 3

value at p+3 is 18

value at p-3 is 32766

value at q-3 is -1145479936

value at q-3 is 15

Address of p+3 is cd75056c

Address of p-3 is cd750554

Address of q-3 is cd750578

Address of q-3 is cd750560

*p++ = 15

p++ = cd750564

*++p = 18

++p = cd750570

*q-- = 18

q-- = cd750568

*--q = 15

--q = cd75055c

Program 9(D): C program that implements call by reference.

Aim :

The main objective of this program is to make familiar with function call using call by reference.

Software used :

Visual studio code.

Program :

```
#include<stdio.h>
void add(int*,int*);
void main()
{
   int x,y;
   printf("Enter the values of x and y :\n");
   scanf("%d %d",&x,&y);
   add(&x,&y);
}
void add(int*a,int*b)
{
   int sum;
   sum=*a+*b;
   printf("Sum of given two numbers is %d",sum);
}
```

Output :

```
Enter the values of x and y :
5  8
Sum of given two numbers is 13
```

Program 10: C program to demonstrate the following

1. Pointers to pointers
2. Array of pointers
3. Pointers to array
4. Pointers to function

AIM:

The main objective of this program is to make familiar the usage of pointers, arrays and functions.

SOFTWARE USED: VISUAL STUDIO CODE (gcc complier).

Program:

1)*Pointers to pointers:*

CODE:

```
#include<stdio.h>
void main()
{
int a;
int *p;
int **q;
int ***r;
printf(“enter the value\n”);
scanf(“%d”,&a);
p=&a;
q=&p;
r=&q;
```

```
printf(“the value entered is %d\n”,a);
printf(“the value entered is %d\n”,*p);
printf(“the value entered is %d\n”**q);
printf(“the value entered is %d\n”***r); }
```

Input & Output:

Input:

a=5

Output:

the value entered is 5

the value entered is 5

the value entered is 5

the value entered is 5

2)*Array of pointers:*

CODE:

```
#include<stdio.h>
const int max=3;
void main()
{
int var[]={10,100,200};
int i,*ptr[max];
for(i=0 ; i<max ; i++)
{
ptr [i]=&var[i];
}
for(i=0 ; i<max ; i++)
{
  printf("value of var[%d]=%d\n",i,*ptr[i]);
 }
}
```

Input & Output:

No input required

Output:

value of var[0]=10

value of var[1]=100

value of var[2]=200

3)*pointer to array*

CODE:

```
#include<stdio.h>
void main()
{
 int n=5, i;
 int arr[5]={11,12,13,14,15};
 int *ptr;
 ptr=arr;
 for(i=0;i<n;i++){
 printf("%d\t",*ptr);
 ptr++;
 }
}
```

Input & Output:

INPUT: No input

OUTPUT:

11 12 13 14 15

4)*pointers to functions*

CODE:

```
#include<stdio.h>
void fun(int a, int b)
{
printf(“ the value of a & b are %d,%d ”,a,b);
}
void main()
{
void(*ptr)( int,int) &fun
ptr(10,20);
ptr(20,30);
}
```

Input & Output:

INPUT: No input

OUTPUT:

the value of a & b are 10,20

the value of a & b are 20,30

Program 11(A):C PROGRAM TO ACCESS AND DISPLAY THE MEMBERS OF THE STRING.

AIM: c program to access and display the members of the string.

SOFTWARE REQUIRED: visual studio code using gcc compiler.

CODE:

```
#include<stdio.h>
#include<string.h>
struct student
{
char name[30];
int marks;
} s1, s2;
void main()
{
s2={“kesav”,60};
strcpy (s1.name,” ramu”);
s1.marks = 90;
printf(“ name of the student 1 %s\n”,s1.name);
printf(“ marks of the student 1 %d\n”,s1.marks);
printf(“ name of the student 2 %s\n”,s2.name);
printf(“ marks of the student 2 %d\n”,s2.marks);
}
```

Output:

name of the student 1: ramu

marks of the student 1: 90

name of the student 2: kesav

marks of the student 2: 60

Program 11(B): C program that demonstrates different ways to access the structure elements using pointers.

AIM: The main objective of this program is to make familiar with accessing the structure members using pointers.

SOFTWARE REQUIRED: visual studio code using gcc compiler.

CODE:

```
#include<stdio.h>
#include<string.h>
struct student
{
  int sno;
  char name[30];
  float marks;
};

void main()
{
   struct student s;
   struct student *st;
   printf("enter the name\n");
   gets(s.name);
   printf("enter the serial no and marks of student \n");
   scanf("%d,%f",&s.sno,&s.marks);
   st=&s;
```

```
    printf("the s.no of student :%d\n",st->sno);
    printf("name of student :%s\n",st->name);
    printf("marks of student :%f\n",st->marks);
}
```

INPUT & OUTPUT:

Input:

1) “Rama”, 2,90.5
2) “Krishna”,3,45.89

Output:

1)the s.no of student :2

name of student: Rama

marks of student: 90.500000

2) the s.no of student :3

name of student: Krishna

marks of student: 45.889999

Program-13: C program to replace all the vowels in a given stringwith a given character.

Aim: To replace the vowels in a string with a given

character.

SOFTWARE REQUIRED: visual studio code using gcc compiler.

Program:

```
#include<stdio.h>
#include<string.h>
void main()
{
   char m,a[50];
   int i,l;
   printf(“enter the string\n”);
   gets(a);
   printf(“the vowel to be replaced\n”);
   scanf(“%c”,&m);
   l=strlen(a);
   for(i=0 ;i < l ;i++)
{
if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u'||a[i]=='A'||a[i]==
'E'||a[i]=='I'||a[i]=='O'||a[i]=='U')
a[i]=m;
puts(a);
} }
```

TESTCASES AND OUTPUT:

S. No.	Input	Expected Output	Actual Output
1	aeiou f	fffff	fffff
2	Rhythm g	Rhythm	Rhythm
3	C Programming x	C Prxgrxmmxng	C Prxgrxmmxng

Program 14:c program to perform arithmetic operations using command line arguments.

Aim:
c program to perform arithmetic operations using command line arguments.

Software used: visual studio code (gcc compiler).

Code:

```
#include<stdio.h>
#include<stdlib.h>
void main (int argc,char *argv[])
{
int a,b,sum=0,div=0,sub=0,mul=0,mod=0;
if(argc==3)
{
a=atoi(argv[1]);
b=atoi(argv[2]);
sum=a+b;
div=a/b;
sub=a-b;
mul=a*b;
mod=a%b;
printf("the argument supplied is %s %s %d %f %d %d %d
\n",argv[1],argv[2],sum,div,sub,mul,mod);
}
else if(argc>3)
printf("too many arguments supplied \n");
else
printf("atleast one argument is expected ");
}
```

Test cases:

S.NO	input	Expected output	output
1.	10 20 *	Multiplication of two numbers 10 20 is 200	10*20=200
2.	10 20 /	Division of two numbers 10 20 is 0.5	10/20=0.5
3.	10 20%	Modulus division of two numbers 10 20 is 10	10% 20=10
4.	10 20 +	Addition of two numbers 10 20 is 30	10+20=30
5.	10 20 -	Subtraction of numbers 10 20 is -10	10-20= -10