



List of programs

Week	Name Of the Program
1	<ul style="list-style-type: none">a) Write a C program to find sum and average of three numbers.b) Write a C program to find the sum of individual digits of a given positive integer.c) Write a C program to generate the first n terms of the Fibonacci sequence.
2	<ul style="list-style-type: none">a) Write a C program to generate prime numbers between 1 to n.b) Write a C program to Check whether given number is Armstrong Number or Not.c) Write a C program to evaluate algebraic expression $(ax+b)/(ax-b)$.
3	<ul style="list-style-type: none">a) Write a C program to check whether given number is perfect number or Not.b) Write a C program to check whether given number is strong number or not.
4	<ul style="list-style-type: none">a) Write a C program to find the roots of a quadratic equation.b) Write a C program perform arithmetic operations using switch statement.
5	<ul style="list-style-type: none">a) Write a C program to find factorial of a given integer using non-recursive function.b) Write a C program to find factorial of a given integer using recursive function.
6	<ul style="list-style-type: none">a) Write C program to find GCD of two integers by using recursive function.b) Write C program to find GCD of two integers using non-recursive function.
7	<ul style="list-style-type: none">a) Write a C program to find both the largest and smallest number in a list of integers.b) Write a C Program to Sort the Array in an Ascending Order.c) Write a C Program to find whether given matrix is symmetric or not.
8	<ul style="list-style-type: none">a) Write a C program to perform addition of two matrices.b) Write a C program that uses functions to perform Multiplication of Two Matrices.
9	<ul style="list-style-type: none">a) Write a C program to use function to insert a sub-string in to given main string from a given position.b) Write a C program that uses functions to delete n Characters from a given position in a given string.



List of programs

10	<ul style="list-style-type: none">a) Write a C program using user defined functions to determine whether the given string is palindrome or not.b) Write a C program that displays the position or index in the main string S where the sub string T begins, or - 1 if S doesn't contain T.
11	<ul style="list-style-type: none">a) Write C program to count the number of lines, words and characters in a given text.b) Write a C program to find the length of the string using Pointer.
12	<ul style="list-style-type: none">a) Write a C program to Display array elements using calloc() function.b) Write a C Program to Calculate Total and Percentage marks of a student using structure.
13	<ul style="list-style-type: none">a) Write a C program that uses functions and structures to perform the following operations:<ul style="list-style-type: none">i) Reading a complex numberii) Writing a complex numberiii) Addition of two complex numbersiv) Multiplication of two complex numbersb) Write a C program to display the contents of a file.
14	<ul style="list-style-type: none">a) Write a C program to copy the contents of one file to another.b) Write a C program to merge two files into a third file.c) Write a C program to reverse the first n characters in a file

UNIVERSITY SCHOOL OF AUTOMATION AND
ROBOTIC

END SEMESTER PRATICAL FILE

ICT 151

B.tech

SEMESTER 01

Submitted to:

Neetu Narang

Mahajan

Submitted by:

Name: **Avishisht Gupta**

Batch: B1-A

Enrollment No. :-

00419011721

INDEX

S.NO.	NAME OF PROGRAM	DATE	SIGNATURE
1-A	Write a C program to find sum and average of three numbers.	16-12-2021	
1-B	Write a C program to find the sum of individual digits of a given positive integer.	16-12-2021	
1-C	Write a C program to generate the first n terms of the Fibonacci sequence.	16-12-2021	
2-A	Write a C program to generate prime numbers between 1 to n.	23-12-2021	
2-B	Write a C program to Check whether given number is Armstrong Number or Not	23-12-2021	
2-C	Write a C program to evaluate algebraic expression $(ax+b)/[ax-b]$.	23-12-2021	
3-A	Write a C program to check whether given number is perfect number or Not.	30-12-2021	
3-B	Write a C program to check whether given number is strong number or not.	30-12-2021	
4-A	Write a C program to find the roots of a quadratic equation.	30-12-2021	
4-B	Write a C program using switch statement.	06-01-2022	
5-A	Write a C program to find factorial of a given integer using non-recursive function.	06-01-2022	
5-B	Write a C program to find factorial of even integer using recursive function.	06-01-2022	
6-A	Write C program to find GCD of two integers by using recursive function.	13-01-2022	
6-B	Write C program to find GCD of two integers using non-recursive function	13-01-2022	
7-A	Write a C program to find both the largest and smallest number in a list of integers.	13-01-2022	
7-B	Write a C Program to Sort the Array in an Ascending Order.	13-01-2022	
7-C	Write a C Program to find whether given matrix is symmetric or not.	20-01-2022	
8-A	Write a C program to perform addition of two matrices.	20-01-2022	
8-B	Write a C program that uses functions to perform Multiplication of Two Matrices	20-01-2022	
9-A	Write a C program to use function to insert a sub-string in to given main string from a given position.	20-01-2022	
9-B	Write a C program that uses functions to delete n Characters from a given position in a given string.	10-01-2022	
10-A	Write a C program using user defined functions to determine whether the given string is palindrome or not.	10-02-2022	
10-B	Write a C program that displays the position or index in the main string S where the sub string T begins, or - 1 if S doesn't contain T.	10-02-2022	
11-A	Write C program to count the number of lines, words and characters in a given text.	17-02-2022	
11-B	Write a C program to find the length of the string using Pointer.	17-02-2022	
12-A	Write a C program to Display array elements using calloc () function.	17-02-2022	
12-B	Write a C Program to Calculate Total and Percentage marks of a student using structure.	17-02-2022	

13-A	Write a C program that uses functions and structures to perform the following operations: i) Reading a complex number if) Writing a complex number iii) Addition of two complex numbers iv) Multiplication of two complex numbers	24-02-2022	
13-B	Write a C program to display the contents of a file.	24-02-2022	
14-A	Write a C program to copy the contents of one file to another.	03-03-2022	
14-B	Write a C program to merge two files into a third file.	03-03-2022	
14-C	Write a C program to reverse the first n characters in a file	03-03-2022	

PROGRAM 1 -A

Write a C program to find sum and average of three numbers

INPUT

```
#include <stdio.h>

int main()
{
    int num1, num2, num3, sum;
    float avg;

    printf("Enter the First Number = ");
    scanf("%d",&num1);

    printf("Enter the Second Number = ");
    scanf("%d",&num2);

    printf("Enter the Third Number = ");
    scanf("%d",&num3);

    sum = num1 + num2 + num3;

    avg = sum / 3;

    printf("\nThe Sum of Three Numbers    = %d", sum);
    printf("\nThe Average of Three Numbers = %.2f\n", avg);
}
```

OUTPUT

```
Enter the First Number = 45
Enter the Second Number = 14
Enter the Third Number = 45

The Sum of Three Numbers    = 104
The Average of Three Numbers = 34.00
```

PROGRAM 1-B

Write a C program to find the sum of individual digits of a given positive integer.

INPUT

```
#include<stdio.h>
int main()
{
    int n,sum=0,m;
    printf("Enter a number:");
    scanf("%d",&n);
    while(n>0)
    {
        m=n%10;
        sum=sum+m;
        n=n/10;
    }
    printf("Sum is=%d",sum);
    return 0;
}
```

OUTPUT

Enter a number:654

Sum is=15

Enter a number:654

Sum is=15

Enter a number:123

Sum is=6

PROGRAM 1-C

Write a C program to generate the first n terms of the Fibonacci sequence.

INPUT

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

    int i, n;

    int t1 = 0, t2 = 1;

    int nextTerm = t1 + t2;

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    printf("Fibonacci Series: %d, %d, ", t1, t2);

    for (i = 3; i <= n; ++i) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}
```

OUTPUT

```
Enter the number of terms: 10
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```


PROGRAM 2-A

Write a C program to generate prime numbers between 1 to n.

INPUT

```
#include<stdio.h>

int main(){

    int num,i,count,n;
    printf("Enter max range: ");
    scanf("%d",&n);

    for(num = 1;num<=n;num++){

        count = 0;

        for(i=2;i<=num/2;i++){
            if(num%i==0){
                count++;
                break;
            }
        }

        if(count==0 && num!= 1)
            printf("%d ",num);
    }

    return 0;
}
```

OUTPUT

```
Enter max range: 50
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
```

PROGRAM 2-B

Write a C program to Check whether given number is Armstrong Number or Not.

INPUT

```
#include <stdio.h>
int main()
{
    int n, n1, rem, num=0;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    n1=n;
    while(n1!=0)
    {
        rem=n1%10;
        num+=rem*rem*rem;
        n1/=10;
    }
    if(num==n)
        printf("%d is an Armstrong number.",n);
    else
        printf("%d is not an Armstrong number.",n);
    return 0;
}
```

Output

```
Enter a positive integer: 4566
4566 is not an Armstrong number.
```

```
Enter a positive integer: 371
371 is an Armstrong number.
```

PROGRAM 2-C

Write a C program to evaluate algebraic expression $(ax+b)/(ax-b)$.

INPUT

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b,x;
    float s;
    printf("Enter the Values of a,b,x...");
    scanf("%d%d%d",&a,&b,&x);
    s=(a*x+b)/(a*x-b);
    printf("The Resultant Value is : %f",s);
    return 0;
}
```

OUTPUT

```
Enter the Values of a,b,x...2 3 5
The Resultant Value is : 1.000000
```

PROGRAM 3- A

Write a C program to check whether given number is perfect number or Not.

INPUT

```
#include<stdio.h>
int main()
{
    int n,i=1,sum=0;

    printf("Enter a number: ");
    scanf("%d",&n);

    while(i<n)
    {
        if(n%i==0)
            sum=sum+i;
        i++;
    }
    if(sum==n)
        printf("%d is a perfect number",i);
    else
        printf("%d is not a perfect number",i);

    return 0;
}
```

Output

```
Enter a number: 6
6 is a perfect number
```

PROGRAM 3-B

Write a C program to check whether given number is strong number or not.

INPUT

```
#include<stdio.h>
int main(){
    int n,i;
    int fact,rem;
    printf("\nEnter a number : ");
    scanf("%d",&n);
    printf("\n");
    int sum = 0;
    int temp = n;
    while(n){
        i = 1,fact = 1;
        rem = n % 10;
        while(i <= rem){
            fact = fact * i;
            i++;
        }
        sum = sum + fact;
        n = n / 10;
    }
    if(sum == temp)
        printf("%d is a strong number\n",temp);
    else
        printf("%d is not a strong number\n",temp);
    return 0;
}
```

OUTPUT

```
Enter a number : 145
145 is a strong number
Enter a number : 25
25 is not a strong number
```

PROGRAM 4-A

Write a C program to find the roots of a quadratic equation.

INPUT

```
#include <stdio.h>
#include <math.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;
    printf("Enter coefficients a, b and c: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    discriminant = b * b - 4 * a * c;

    // condition for real and different roots
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", root1, root2);
    }

    // condition for real and equal roots
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", root1);
    }

    // if roots are not real
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi",
realPart, imagPart, realPart, imagPart);
    }

    return 0;
}
```

Output

```
Enter coefficients a, b and c: 2.3
4
5.6
root1 = -0.87+1.30i and root2 = -0.87-1.30i
```

PROGRAM 4-B

Write a C program perform arithmetic operations using switch statement.

INPUT

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b;
    int op;
    printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
    printf("Enter the values of a & b: ");
    scanf("%d %d",&a,&b);
    printf("Enter your Choice : ");
    scanf("%d",&op);
    switch(op)
    {
        case 1    :
            printf("Sum of %d and %d is : %d",a,b,a+b);
            break;
        case 2    :
            printf("Difference of %d and %d is : %d",a,b,a-b);
            break;
        case 3    :
            printf("Multiplication of %d and %d is : %d",a,b,a*b);
            break;
        case 4    :
            printf("Division of Two Numbers is %d : ",a/b);
            break;
        default   :
            printf(" Enter Your Correct Choice.");
            break;
    }
    return 0;
}
```

Output

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter the values of a & b:  20 15
Enter your Choice : 1
Sum of 20 and 15 is : 35
```


PROGRAM 5(A)

Write a C program to find factorial of a given integer using non-recursive function.

INPUT

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

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

int main(int argc, char **argv)
{
    int n;
    printf("Enter a positive number: ");
    scanf("%d", &n);
    printf("\n");
    printf("Non-recursive solution: %d! = %d", n, factorial(n));
    return 0;
}
```

OUTPUT

```
Enter a positive number: 5

Non-recursive solution: 5! = 120
```

PROGRAM 5(B)

Write a C program to find factorial of even integer using recursive function.

INPUT

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

int rfactorial(int n)
{
    if(n == 0 || n==1)
        return 1;
    else
        return n * rfactorial(n - 1);
}

int main(int argc, char **argv)
{
    int n;
    printf("Enter a positive number: ");
    scanf("%d", &n);
    printf("\n");
    printf("Recursive solution: %d! = %d", n, rfactorial(n));
    return 0;
}
```

OUTPUT

Enter a positive number: 8

Recursive solution: 8! = 40320

PROGRAM 6-A

Write C program to find GCD of two integers by using recursive function.

INPUT

```
#include <stdio.h>
int hcf(int n1, int n2);
int main() {
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
    return 0;
}

int hcf(int n1, int n2) {
    if (n2 != 0)
        return hcf(n2, n1 % n2);
    else
        return n1;
}
```

OUTPUT

```
Enter two positive integers: 366
60
G.C.D of 366 and 60 is 6.
```

PROGRAM 6-B

Write C program to find GCD of two integers using non-recursive function

INPUT

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int gcdnonR(int i,int j){
    int rem;
    rem=i-(i/j*j);
    if(rem==0)
        return j;
    else
        gcdnonR(j,rem);
}
int main(){
    int a,b;
    printf("enter the two numbers:");
    scanf("%d%d",&a,&b);
    printf("GCD of %d",gcdnonR(a,b));
    return 0;
}
```

OUTPUT

```
enter the two numbers:10
30
GCD of 10
```

PROGRAM 7-A

Write a C program to find both the largest and smallest number in a list of integers.

INPUT

```
#include<stdio.h>
int main()
{
    int i, n, lar,sm, elem;
    printf ("Enter total number of elements ");
    scanf ("%d", &elem);
    printf ("Enter first number ");
    scanf ("%d", &n);
    lar = n;
    sm=n;
    for (i=1; i<= elem -1 ; i++)
    {
        printf ("n Enter another number ");
        scanf ("%d",&n);
        if (n>lar)
            lar=n;
        if (n<sm)
            sm=n;
    }
    printf ("\n The largest number is %d", lar);
    printf ("\n The smallest number is %d", sm);
    return 0;
}
```

OUTPUT

Enter total number of elements 5

Enter first number 10

Enter another number 12

Enter another number 14

Enter another number 16

Enter another number 45

The largest number is 45

The smallest number is 10

PROGRAM 7-B

Write a C Program to Sort the Array in an Ascending Order.

INPUT

```
#include <stdio.h>
int main (){
    int num[20];
    int i, j, a, n;
    printf("enter number of elements in an array\n");
    scanf("%d", &n);
    printf("Enter the elements\n");
    for (i = 0; i < n; ++i)
        scanf("%d", &num[i]);
    for (i = 0; i < n; ++i){
        for (j = i + 1; j < n; ++j){
            if (num[i] > num[j]){
                a = num[i];
                num[i] = num[j];
                num[j] = a;
            }
        }
    }
    printf("The numbers in ascending order is:\n");
    for (i = 0; i < n; ++i){
        printf("%d\n", num[i]);
    }
    return 0;
}
```

OUTPUT

```
enter number of elements in an array
7
Enter the elements
15 1 5 13 17 10 9
The numbers in ascending order is:
1
5
9
10
13
15
17
```

PROGRAM 7-C

Write a C Program to find whether given matrix is symmetric or not.

INPUT

```
#include<stdio.h>

int main()
{
    int i, j, rows, columns, a[10][10], b[10][10], Count = 1;

    printf("\n Please Enter Number of rows and columns  :  ");
    scanf("%d %d", &i, &j);

    printf("\n Please Enter the Matrix Elements \n");
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            scanf("%d", &a[rows][columns]);
        }
    }
    //Transpose of matrix
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            b[columns][rows] = a[rows][columns];
        }
    }

    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            if(a[rows][columns] != b[rows][columns])
            {
                Count++;
                break;
            }
        }
    }
    if(Count == 1)
    {
```



```
        printf("\n The Matrix that you entered is a Symmetric
Matrix ");
    }
    else
    {
        printf("\n The Matrix that you entered is Not a Symmetric
Matrix ");
    }

    return 0;
}
```

OUTPUT

Please Enter Number of rows and columns : 3 3

Please Enter the Matrix Elements

2 3 4

2 6 8

3 5 8

The Matrix that you entered is Not a Symmetric Matrix

PROGRAM 8-A

Write a C program to perform addition of two matrices.

INPUT

```
include<stdio.h>

int main()
{
    int i, j, rows, columns, a[10][10], b[10][10];
    int arr[10][10];

    printf("\n Please Enter Number of rows and columns : ");
    scanf("%d %d", &i, &j);

    printf("\n Please Enter the First Elements\n");
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            scanf("%d", &a[rows][columns]);
        }
    }

    printf("\n Please Enter the Second Elements\n");
    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            scanf("%d", &b[rows][columns]);
        }
    }

    for(rows = 0; rows < i; rows++)
    {
        for(columns = 0; columns < j; columns++)
        {
            arr[rows][columns] = a[rows][columns] +
b[rows][columns];
        }
    }

    printf("\n The Sum of Two a and b = a + b \n");
    for(rows = 0; rows < i; rows++)
```

```
{
    for(columns = 0; columns < j; columns++)
    {
        printf("%d \t ", arr[rows][columns]);
    }
    printf("\n");
}
return 0;
}
```

OUTPUT

Please Enter Number of rows and columns : 2 3

Please Enter the First Elements

2 3 4

4 6 7

Please Enter the Second Elements

3 6 8

4 5 7

The Sum of Two a and b = a + b

5 9 12

8 11 1

PROGRAM 8-B

Write a C program that uses functions to perform Multiplication of Two Matrices.

INPUT

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int mat1[3][3], mat2[3][3], mat3[3][3], sum=0, i, j, k;
    printf("Enter first 3*3 matrix element: ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            scanf("%d", &mat1[i][j]);
    }
    printf("Enter second 3*3 matrix element: ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            scanf("%d", &mat2[i][j]);
    }
    printf("\nMultiplying two matrices...");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            sum=0;
            for(k=0; k<3; k++)
                sum = sum + mat1[i][k] * mat2[k][j];
            mat3[i][j] = sum;
        }
    }
    printf("\nMultiplication result of the two given Matrix is: \n");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            printf("%d\t", mat3[i][j]);
        printf("\n");
    }
}
```

```
    return 0;  
}
```

OUTPUT:

```
Enter first 3*3 matrix element: 1 2 3 4 5 6 7 8 9  
Enter second 3*3 matrix element: 10 11 12 13 14 15 16 17 18  
  
Multiplying two matrices...  
Multiplication result of the two given Matrix is:  
84      90      96  
201     216     231  
318     342     366
```

PROGRAM 9-A

Write a C program to use function to insert a sub-string in to given main string from a given position.

INPUT

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    char str1[20], str2[20];
    int l1, l2, n, i;
    puts("Enter the string 1\n");
    gets(str1);
    l1 = strlen(str1);
    puts("Enter the string 2\n");
    gets(str2);
    l2 = strlen(str2);
    printf("Enter the position where the string is to be
inserted\n");
    scanf("%d", &n);
    for(i = n; i < l1; i++)
    {
        str1[i + l2] = str1[i];
    }
    for(i = 0; i < l2; i++)
    {
        str1[n + i] = str2[i];
    }
    str2[l2 + 1] = '\0';
    printf("After inserting the string is %s", str1);
    return 0;
}
```

OUTPUT

```
Enter First String:
program
Enter Second String:
ming
Enter the position where the item has to be inserted: 7
programming
```

PROGRAM 9-B

Write a C program that uses functions to delete n Characters from a given position in a given string.

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

int delchar(char *x,int a, int b);

int main()
{
    char string[10];
    int n,pos,p;
    puts("Enter a string :");
    gets(string);
    printf("Enter the position from where you want to delete:");
    scanf("%d",&pos);
    printf("Enter the number of characters to be deleted :");
    scanf("%d",&n);
    delchar(string, n,pos);
    return 0;
}

// Function to delete n characters
int delchar(char *x,int a, int b);
{
    if ((a+b-1) <= strlen(x))
    {
        strcpy(&x[b-1],&x[a+b-1]);
        puts(x);
    }
}
```

OUTPUT

```
Enter a string :
programming
Enter the position from where you want to delete:6
Enter the number of characters to be deleted :5
progrg
```

PROGRAM 10-A

Write a C program using user defined functions to determine whether the given string is palindrome or not.

INPUT

```
#include <stdio.h>
#include <string.h>
int main(){
    char string1[20];
    int i, length;
    int flag = 0;

    printf("Enter a string:");
    scanf("%s", string1);

    length = strlen(string1);
    for(i=0;i < length ;i++){
        if(string1[i] != string1[length-i-1]){
            flag = 1;
            break;
        }
    }
    if (flag) {
        printf("%s is not a palindrome", string1);
    }
    else {
        printf("%s is a palindrome", string1);
    }
    return 0;
}
```

OUTPUT:

```
Enter a string:COMPUTER
COMPUTER is not a palindrome
```


PROGRAM 10-B

Write a C program that displays the position or index in the main string S where the sub string T begins , or -1 if S doesn't contain T

INPUT

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
int main()
{
    char s[30], t[20];
    char *found;
    puts("Enter the first string: ");
    gets(s);
    puts("Enter the string to be searched: ");
    gets(t);
    found = strstr(s, t);
    if(found)
    {
        printf("Second String is found in the First String at %d position.\n", found - s);
    }
    else
    {
        printf("-1");
    }
    return 0;
```

OUTPUT

```
Enter the first string:
computer
Enter the string to be searched:
pu
Second String is found in the First String at 3 position.
```

PROGRAM 11-A

Write C program to count the number of lines, words and characters in a given text.

INPUT

```
#include<stdio.h>
int main()
{
    // declare variables
    char str[200];
    int line, word, ch;
    // initialize count variables with zero line = word = ch = 0;
    // read multiline string
    printf("Enter string terminated with ~ :\n");
    scanf("%[^~]", str);
    // check every character
    for(int i=0; str[i]!='\0'; i++)
    {
        // if it is new line then
        // one line and one word completed
        if(str[i]=='\n')
        {
            line++;
            word++;
        }
        // else it is a character
        else
        {
            // if character is space or tab
            // then one word is also completed
            if(str[i]==' ' || str[i]=='\t')
            {
                word++;
                ch++;
            }
            // it was not '\n', space or tab
            // it is a normal character
            else {
                ch++;
            }
        }
    }
}
```

```
}  
// display count values  
printf("\nCharacter counts = %d\n", ch);  
printf("Word counts = %d\n", word);  
printf("Line counts = %d\n", line);  
return 0;  
}
```

OUTPUT

```
Enter string terminated with ~ :  
Hello, how are you?  
Welcome to the programming world.  
Programming is fun.  
~  
Character counts = 71  
Word counts = 12  
Line counts = 3
```

PROGRAM 11-B

Write a C program to find the length of the string using Pointer.

INPUT

```
#include<stdio.h>
#include<conio.h>
int string_ln(char*);
void main() {
    char str[20];
    int length;
    printf("\nEnter any string : ");
    gets(str);
    length = string_ln(str);
    printf("The length of the given string %s is : %d", str, length);
    getch();
}
int string_ln(char*p) /* p=&str[0] */
{
    int count = 0;
    while (*p != '\0') {
        count++;
        p++;
    }
    return count;
}
```

OUTPUT

```
Enter any string : computer
The length of the given string computer is : 8
```

PROGRAM 12-A

Write a C program to Display array elements using calloc() function.

INPUT

```
#include<stdlib.h>
int main()
{
    int *p, i, n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);
    p = (int*)calloc(n, sizeof(int));
    if(p==NULL)
    {
        printf("Memory allocation failed");
        exit(1); // exit the program
    }
    for(i = 0; i < n; i++)
    {
        printf("Enter %d element: ", i);
        scanf("%d", p+i);
    }
    printf("\nprinting array of %d integers\n\n", n);
    // calculate sum
    for(i = 0; i < n; i++)
    {
        printf("%d ", *(p+i));
    }
    // signal to operating system program ran fine
    return 0;
}
```

OUTPUT

```
Enter any string : 3
The length of the given string 3 is : 1
```

PROGRAM 12-B

Write a C Program to Calculate Total and Percentage marks of a student using structure.

INPUT

```
#include <stdio.h>
struct student
{
    char name [30];
    int marks[ 5];
    int total;
    float percentage;
};
int main()
{
    struct student std;
    int i;
    printf("Enter name: ");
    gets(std.name);
    printf("Enter marks:\n");
    std.total=0;
    for(i=0;i< 5;i++){
        printf("Marks in subject %d: ",i+1);
        scanf("%d",&std.marks[i]);
        std.total+=std.marks[i];
    }
    std.percentage=(float)((float)std.total/(float)500)*100;
    printf("\nName: %s \nTotal Marks: %d \nPercentage:
%.2f",std.name,std.total,std.percentage);
    return 0;
}
```

OUTPUT

Enter name: ABC

Enter marks:

Marks in subject 1: 98

Marks in subject 2: 85

Marks in subject 3: 90

Marks in subject 4: 80

Marks in subject 5: 75

Name: ABC

Total Marks: 428

Percentage: 85.60

PROGRAM 13-A

Write a C program that uses functions and structures to perform the following operations:

- i) Reading a complex number**
- ii) Writing a complex number**
- iii) Addition of two complex numbers**
- iv) Multiplication of two complex numbers**

INPUT

```
#include <stdio.h>
#include <conio.h>
struct complex
{
    float real, imag;
}a, b, c;
struct complex read(void);
void write(struct complex);
struct complex add(struct complex, struct complex);
struct complex sub(struct complex, struct complex);
struct complex mul(struct complex, struct complex);
struct complex div(struct complex, struct complex);
void main ()
{
    printf("Enter the 1st complex number\n");
    a = read();
    write(a);
    printf("Enter the 2nd complex number\n");
    b = read();
    write(b);
    printf("Addition ");
    c = add(a, b);
    write(c);
    printf("Substraction");
    c = sub(a, b);
    write(c);
    printf("Multiplication");
    c = mul(a, b);
    write(c);
    printf("Division");
```



```

    c = div(a, b);
    write(c);
    getch();
}
struct complex read(void)
{
    struct complex t;
    printf("Enter the real part");
    scanf("%f", &t.real);
    printf("Enter the imaginary part");
    scanf("%f", &t.imag);
    return t;
}
void write(struct complex a)
{
    printf("Complex number is");
    printf(" %.1f + i %.1f", a.real, a.imag);
    printf("\n");
}
struct complex add(struct complex p, struct complex q)
{
    struct complex t;
    t.real = (p.real + q.real);
    t.imag = (p.imag + q.imag);
    return t;
}
struct complex sub(struct complex p, struct complex q)
{
    struct complex t;
    t.real = (p.real - q.real);
    t.imag = (p.imag - q.imag);
    return t;
}
struct complex mul(struct complex p, struct complex q)
{
    struct complex t;
    t.real=(p.real * q.real) - (p.imag * q.imag);
    t.imag=(p.real * q.imag) + (p.imag * q.real);
    return t;
}
struct complex div(struct complex p, struct complex q)
{
    struct complex t;
    t.real = ((p.imag * q.real) - (p.real * q.imag)) / ((q.real * q.real) + (q.imag * q.imag));
    t.imag = ((p.real * q.real) + (p.imag * q.imag)) / ((q.real * q.real) + (q.imag * q.imag));
}

```

```
    return(t);  
}
```

OUTPUT

```
Enter the 1st complex number  
Enter the real part 2  
Enter the imaginary part 4  
Complex number is 2.0 + i 4.0  
Enter the 2nd complex number  
Enter the real part 4  
Enter the imaginary part 2  
Complex number is 4.0 + i 2.0  
Addition Complex number is 6.0 + i 6.0  
SubstractionComplex number is -2.0 + i 2.0  
MultiplicationComplex number is 0.0 + i 20.0  
DivisionComplex number is 0.6 + i 0.8
```

PROGRAM 13-B

Write a C program to display the contents of a file.

INPUT:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void copy(FILE *source, FILE *target)
{
    char ch;
    system("cd D:");
    if(source != NULL)
    {
        while((ch=fgetc(source))!=EOF)
        {
            fputc(ch,target);
        }
        fclose(source);
        fclose(target);
        printf("Copied successfully...\n");
    }
    else
    {
        fclose(source);
        fclose(target);
        printf("Press any key to exit...\n");
    }
}

int main()
{
    char s[80], t[80];
    printf("Enter the source file: ");
    scanf("%s",s);
    printf("\nEnter the target file: ");
    scanf("%s",t);
    FILE *source = fopen(s,"r+");
    FILE *target = fopen(t,"a+");
    copy(source, target);
    getch();
    return 0;
```

}

OUTPUT

```
Enter the source file: s.txt  
Enter the target file: t.txt  
D:\  
Copied successfully...
```

PROGRAM 14-A

Write a C program to copy the contents of one file to another.

INPUT

```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
    FILE *fptr1, *fptr2;
    char filename[100], c;

    printf("Enter the filename to open for reading \n");
    scanf("%s", filename);

    // Open one file for reading
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL)
    {
        printf("Cannot open file %s \n", filename);
        exit(0);
    }

    printf("Enter the filename to open for writing \n");
    scanf("%s", filename);

    // Open another file for writing
    fptr2 = fopen(filename, "w");
    if (fptr2 == NULL)
    {
        printf("Cannot open file %s \n", filename);
        exit(0);
    }

    // Read contents from file
    c = fgetc(fptr1);
    while (c != EOF)
    {
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }
    printf("\nContents copied to %s", filename);
}
```

```
fclose(fp1);  
fclose(fp2);  
return 0;  
}
```

OUTPUT

```
Enter the filename to open for reading  
a.txt  
Enter the filename to open for writing  
b.txt  
Contents copied to b.txt
```

PROGRAM 14-B

Write a C program to merge two files into a third file.

INPUT

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

int main()
{
    FILE *sourceFile1;
    FILE *sourceFile2;
    FILE *destFile;
    char sourcePath1[100];
    char sourcePath2[100];
    char destPath[100];

    char ch;

    /* Input path of files to merge to third file */
    printf("Enter first source file path: ");
    scanf("%s", sourcePath1);
    printf("Enter second source file path: ");
    scanf("%s", sourcePath2);
    printf("Enter destination file path: ");
    scanf("%s", destPath);

    /*
     * Open source files in 'r' and
     * destination file in 'w' mode
     */
    sourceFile1 = fopen(sourcePath1, "r");
    sourceFile2 = fopen(sourcePath2, "r");
    destFile = fopen(destPath, "w");

    /* fopen() return NULL if unable to open file in given mode. */
    if (sourceFile1 == NULL || sourceFile2 == NULL || destFile == NULL)
    {
        /* Unable to open file hence exit */
        printf("\nUnable to open file.\n");
        printf("Please check if file exists and you have read/write privilege.\n");

        exit(EXIT_FAILURE);
    }
}
```

```
/* Copy contents of first file to destination */
while ((ch = fgetc(sourceFile1)) != EOF)
    fputc(ch, destFile);

/* Copy contents of second file to destination */
while ((ch = fgetc(sourceFile2)) != EOF)
    fputc(ch, destFile);

printf("\nFiles merged successfully to '%s'.\n", destPath);

/* Close files to release resources */
fclose(sourceFile1);
fclose(sourceFile2);
fclose(destFile);

return 0;
}
```

OUTPUT

```
Enter first source file path: data\file1.txt
Enter second source file path: data\file2.txt
Enter destination file path: data\merged-file.txt

Files merged successfully to 'data\merged-file.txt'.
Enter second source file path: data\file2.txt
```


PROGRAM 14-C

Write a C program to reverse the first n characters in a file

INPUT:

```
#include <stdio.h>
int main()
{
    char str[1000], rev[1000];
    int i, j, count = 0;
    scanf("%s", str);
    printf("\nString Before Reverse: %s", str);
    //finding the length of the string
    while (str[count] != '\0')
    {
        count++;
    }
    j = count - 1;
    //reversing the string by swapping
    for (i = 0; i < count; i++)
    {
        rev[i] = str[j];
        j--;
    }
    printf("\nString After Reverse: %s", rev);
}
```

OUTPUT:

```
Hello
String Before Reverse: Hello
String After Reverse: olleH
```