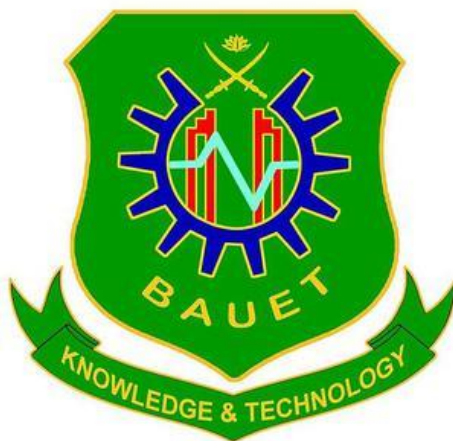


Bangladesh Army University of Engineering & Technology (BAUET)

Qadirabad, Natore 6431, Bangladesh



Department of Information and Communication Engineering (ICE)

Structured Programming Language Lab Manual

Course Code	ICE-1212
Course Title	Structured Programming Language Sessional
Credit Hours	1.5

Prepared by	Verified By
1. A.B.M. Kabir Hossain Lecturer, Dept. of ICE	1. Md. Zihadur Rahman Lecturer, Dept. of ICE

Knowledge and Technology

Table of Content

Si.	Content	Page
1	General Guidelines & Safety Instructions	3
2	Course Description	4
3	Course Objectives	4
4	Statement of Course Outcomes	4
5	Assessment of Course Outcomes	5
6	Mark Distribution and Assessment Criteria	5
7	Format of Lab Report	5
8	Instruction for Lab Report Writing	6
9	Lab Project	6

General Guideline and Safety Instructions

1. Strictly follow the written and verbal instructions given by the teacher /Lab Instructor. If you do not understand the instructions, the handouts and the procedures, ask the instructor or teacher.
2. Students are required to attend all labs with official dress code and wearing ID card.
3. Mobile phones should be switched off in the lab. Keep bags in the bag shelf.
4. Keep the labs clean at all times, no food and drinks allowed inside the lab.
5. Students should work individually/team in the hardware and software task.
6. Students have to bring the lab manual cum lab report file along with them whenever they come for lab work.
7. Should take only the lab manual, calculator (if needed) and a pen or pencil to the work area.
8. Should utilize 3 hours' time properly to perform the experiment and to record the readings. Do the calculations, draw the graphs and take signature from the instructor.
9. If the experiment is not completed in the stipulated time, the pending work has to be carried out in the leisure hours or extended hours.
10. Intentional misconduct will lead to expulsion from the lab.
11. Do not handle any equipment without reading the safety instructions. Read the handout and procedures in the Lab Manual before starting the experiments.
12. Do your wiring, setup, and a careful circuit checkout before applying power. Do not make circuit changes or perform any wiring when power is on.
13. Avoid contact with energized electrical circuits.
14. Do not insert connectors forcefully into the sockets.
15. **NEVER** try to experiment with the power from the wall plug.

Course Description

To introduce the fundamental principles, mechanism of programming skills and develop basic programming skills to program design and development.

Course Objective:

At the end of the course students should

1. To learn basic idea of programming languages.
2. To learn how to write a program with C.
3. To learn how to think about the problems, their solutions and translating it to programming language.
4. To learn how to write program with user defined function and built-in function.
5. To learn how write program with file, structure, union, array etc.

Statement of Course Outcomes (CO):

Upon completion of all sessional, the students will be able to:

1. Discuss algorithm and solve problems using computers.
2. Practically analyze the fundamental principles, typical characteristics, and mechanisms of a structured programming language.
3. Apply practical knowledge to develop basic programming skills with respect to program design and development.

Assessment of Course Outcomes (CO):

CO	PO	Bloom's Taxonomy Level	KP	CP	CA	Delivery methods and activities	Assessment Tools
CO1	PO1	C2, C3, C4	KP1			Lecture, Lab Manual, and Demonstration	Lab Quiz, Lab Viva
CO2	PO3	C3		CP1 CP3 CP7		Lab Manual, Demonstration	Lab Performance/Lab Test
CO3	PO4	C4				Lab Manual, Demonstration	Lab Report, Lab Test

Assessment Criteria and Marks Distribution

Si. No.	Particulars	Marks
1	Lab Performance	10
2	Lab Report	20
3	Quiz Test	20
4	Lab Test	40
5	Lab Viva	10
Total		100

Format of Lab Report

All lab reports should have to follow the common format as below

- Experiment No
- Experiment Title
- Objectives
- Theory Overview (with formula/equations and/or figure if required)
- Results
- Discussion
- Conclusion

Instruction for Lab Report Writing

- Lab report must be hand written without copying from other works.
- Writing should be neat and clean with proper caption and labeling in figure and table.
- The title page of report should contain all the basic information such as experiment no & title, course code & title, student's information, teacher's information, experiment date, submission date.
- Result should include calculated and/or simulated and/or measured data with proper unit.
- Table and/or graph of result should be neat and clear with axis label and units where applicable.
- The discussion should present your findings from the experiment. Evaluate the outcome objectively, taking a candid and unbiased point of view. Suppose that the outcome is not close to what you expected. Even then, after checking your results, give reasons why you believe that outcome is not consistent with the expected.
- In discussion, state the discrepancies between the experimental results and the model (theory), and discuss the sources of the differences in terms of the errors by offering logical inferences and suggest improvements.
- Conclusion should present, a brief summary of what was done, how it was done, show the results and conclusions of the experiment.
- Report should be submitted timely, late submission will cause reduction of marking.
- All lab reports have to be maintained in a single file which has to bring in every laboratory class.

CONTENTS

Week	Name of the program	Page no
1	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.	1-6
2	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 $(ax+b)/(ax-b)$.	7-12
3	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.	13-15
4	a) Write a C program to find the roots of a quadratic equation. b)Write a C program perform arithmetic operations using switch statement.	16-21
5	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.	22-25
6	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.	26-29
7	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.	30-34
8	Revision of Programs	
9	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.	35-40
10	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.	41-45
11	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.	46-49
12	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.	50-53
13	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.	54-55
14	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 b) Write a C program to display the contents of a file.	56-60

15	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.	61-67
16	Revision of Programs	

INSTRUCTIONS TO STUDENTS

- Before entering the lab the student should carry the following things (MANDATORY)
 1. Identity card issued by the college.
 2. Class notes
 3. Lab observation book
 4. Lab Manual
 5. Lab Record
- Student must sign in and sign out in the register provided when attending the lab session without fail.
- Come to the laboratory in time. Students, who are late more than 15 min., will not be allowed to attend the lab.
- Students need to maintain 100% attendance in lab if not a strict action will be taken.
- All students must follow a Dress Code while in the laboratory
- Foods, drinks are NOT allowed.
- All bags must be left at the indicated place.
- Refer to the lab staff if you need any help in using the lab.
- Respect the laboratory and its other users.
- Workspace must be kept clean and tidy after experiment is completed.
- Read the Manual carefully before coming to the laboratory and be sure about what you are supposed to do.
- Do the experiments as per the instructions given in the manual.
- Copy all the programs to observation which are taught in class before attending the lab session.
- Students are not supposed to use floppy disks, pen drives without permission of lab- in charge.
- Lab records need to be submitted on or before the date of submission.

Week 1:

1. a) Write a C program to find the sum and average of three numbers.

Algorithm:

Step 1: Start

Step 2: Declare variables num1, num2, num3 and sum, average.

Step 3: Read values num1, num2, num3

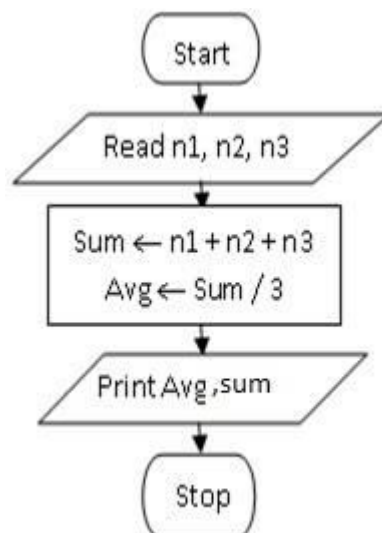
Step 4: Add num1, num2, num3 and assign the result to sum.

$\text{sum} \leftarrow \text{num1} + \text{num2} + \text{num3}$

$\text{average} \leftarrow \text{sum} / 3$

Step 5: Display sum and average

Step 6: Stop

Flow Chart:

Program:

```
#include<stdio.h>
int main( )
{
    int a,b,c;
    int sum,average;
    printf("Enter any three integers: ");
    scanf("%d%d %d",&a,&b,&c);
    sum = a+b+c;
    average=sum/3
    printf("Sum and average  of three  integers: %d %d",sum,average);
    return 0;
}
```

SAMPLE INPUT:

Enter any three integers:2 4 5

EXPECTED OUTPUT:

Sum and average of three integers: 11 3

Record at least 3 results

Signature of faculty with date

1. b) Write a C program to find the sum of individual digits of positive integer.**AIM:**

To find the sum of individual digits of positive integer.

Description:

Summation of digits of a number

Ex: 1234

Summation = $1+2+3+4=10$

ALGORITHM:

Step 1: Start

Step 2: Read n

Step 3: Initialize sum $\leftarrow 0$

Step 4: while(n!=0)

Begin

Step 5: $r \leftarrow n \% 10$

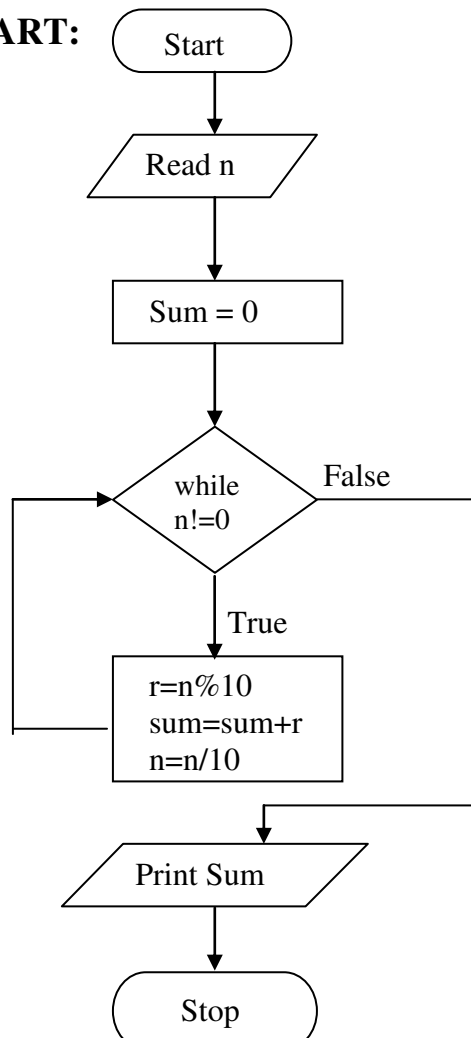
Step 6: $\text{sum} \leftarrow \text{sum} + r$

Step 7: $n \leftarrow n / 10$

End

Step 8: Print "sum"

Step 9: Stop

FLOWCHART:

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,r,sum=0;
    clrscr();
    printf("ENTER A POSITIVE INTEGER \n");
    scanf("%d",&n);
    while(n!=0)
    {
        r=n%10;
        sum=sum+r;
        n=n/10;
    }
    printf("THE SUM OF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..%d",sum);
    getch();
}
```

SAMPLE INPUT:

ENTER A POSITIVE INTEGER
5 3 2 1

EXPECTED OUTPUT:

THE SUM OF INDIVIDUAL DIGITS OF A POSITIVE INTEGER IS..11

Record at least 3 results

Signature of faculty with date

1).c) **Fibonacci Sequence** is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write a C program to generate the first n terms of the sequence.

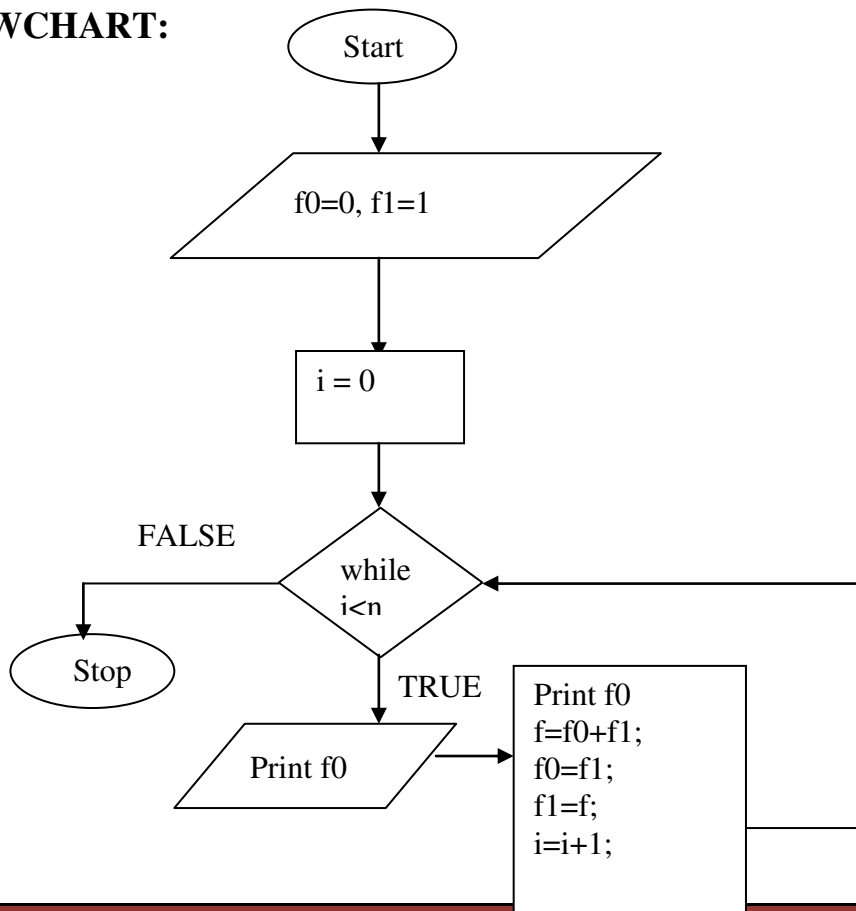
AIM: To generate the first n terms of the Fibonacci sequence..

Description: Initial Fibonacci numbers are 0 and 1. Next number can be generated by adding two numbers. So $0+1=1$. Therefore next number can be generated by adding two previous. so Fibonacci series is 0 1 1 2 3 5

ALGORITHM:

Step 1 : Start
 Step 2 : Read n
 Step 3 : Initialize $f0 \leftarrow 0$, $f1 \leftarrow 1$, $f \leftarrow 0$
 Step 4 : $i=0$
 Step 5 : while($i \leq n$) do as follows
 printf("%d\t",f0);
 $f=f0+f1$;
 $f0=f1$;
 $f1=f$;
 $i=i+1$;
 If not goto step 7
 Step 6 : Stop

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int f0,f1,f,n,i;
    clrscr();
    printf("ENTER THE VALUE FOR n \n");
    scanf("%d",&n);
    f0=0;
    f1=1;
    printf("FIBONACCI SEQUENCE FOR THE FIRST %d TERMS:\n",n);
    i=0;
    while(i<n)
    {
        printf("%d\t",f0);
        f=f0+f1;
        f0=f1;
        f1=f;
        i=i+1;
    }
}
```

INPUT:

ENTER THE VALUE FOR n
10

OUTPUT:

FIBONACCI SEQUENCE FOR THE FIRST 10 TERMS:

0 1 1 2 3 5 8 13 21 34

Record at least 3 results

Signature of faculty with date

Week: 2

2) a) Write a C program to generate all prime numbers between 1 and n. Where n is the value supplied by the user.

Aim: To print a prime numbers up to 1 to n

Description:

Prime number is a number which is exactly divisible by one and itself only

Ex: 2, 3, 5, 7,

Algorithm:

Step 1: start

Step 2: read n

Step 3: initialize i=1, c=0

Step 4: if i<=n goto step 5

If not goto step 10

Step 5: initialize j=1

Step 6: if j<=i do the following. If no goto step 7

i) if i%j==0 increment c

ii) increment j

iii) goto Step 6

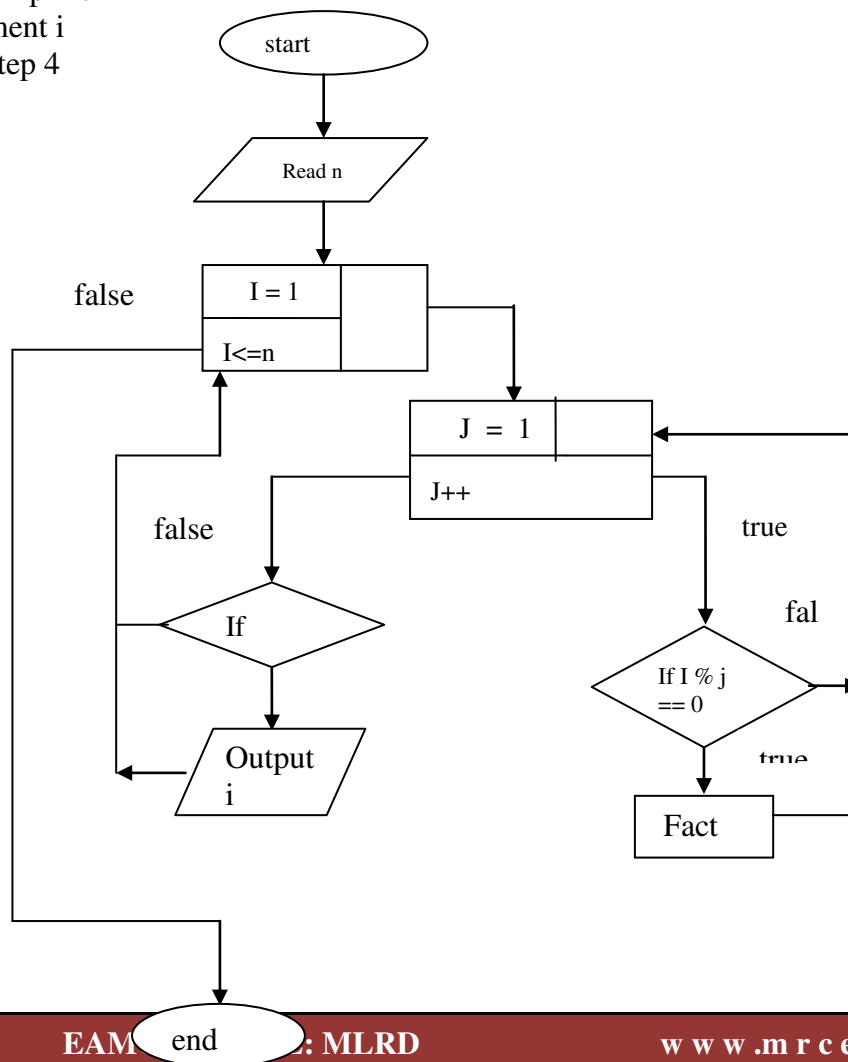
Step 7: if c== 2 print i

Step 8: increment i

Step 9: goto step 4

Step 10: stop

FLOWCHART:



Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,fact,j;
    clrscr();
    printf("enter the number:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        fact=0;
        //THIS LOOP WILL CHECK A NO TO BE PRIME NO. OR NOT.
        for(j=1;j<=i;j++)
        {
            if(i%j==0)
                fact++;
        }
        if(fact==2)
            printf("\n %d",i);
    }
    getch( );
}
```

Output:

Enter the number : 5
2 3 5

Record at least 3 results

Signature of faculty with date

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

AIM: To Check whether given number is Armstrong Number or Not

Algorithm:

Armstrong number

Step 1: start

Step 2: read n

Step 3: assign $\text{sum} \leftarrow 0, I \leftarrow m \leftarrow n, \text{count} = 0$

Step 4: if $m > 0$ repeat

Step 4.1: $m \leftarrow m/10$

Step 4.2: $\text{count}++$

Step 4.3: until the condition fail

Step 5: if $I > 0$ repeat step 4 until condition fail

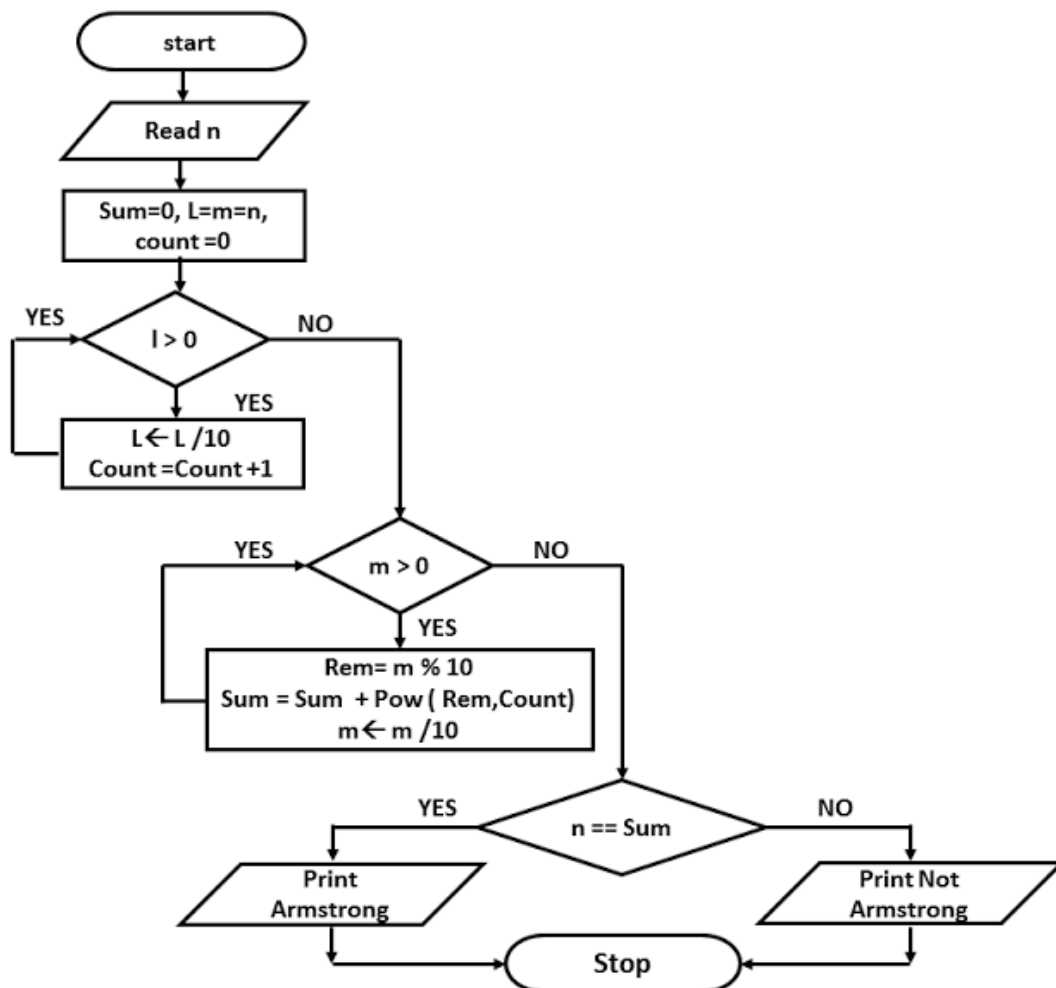
Step 5.1: $\text{rem} \leftarrow I \% 10$

Step 5.2: $\text{sum} \leftarrow \text{sum} + \text{pow}(\text{rem}, \text{count})$

Step 5.3: $I \leftarrow I/10$

Step 6: if $n = \text{sum}$ print Armstrong otherwise print not armstrong

Step 7: stop



Program:

```
#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);
}
```

Input:

Enter a positive integer: 371

Output:

371 is an Armstrong number.

Record at least 3 results

Signature of faculty with date

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

Algorithm:

Step 1: start

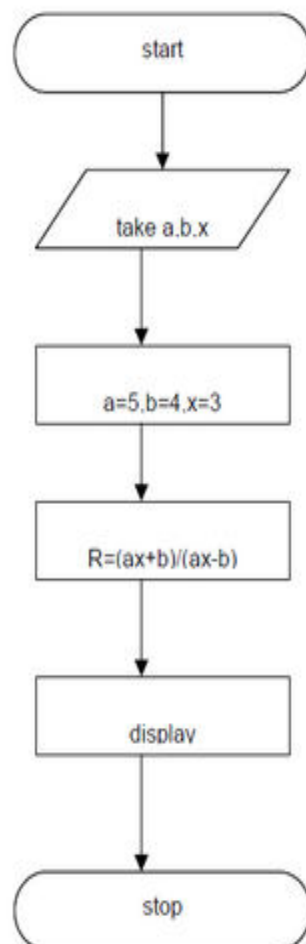
Step 2: input a,b,x,s

Step 3: $s=(a*x+b)/(a*x-b)$

Step 4: Result s

Step 5: stop

Flow Chart:



Program:

```
#include<stdio.h>
#include<conio.h>
int main( )
{
    int a,b,x;
    float s;
    clrscr();
    printf("enter the values of a,b,x");
    scanf("%d %d %d",&a,&b,&x);
    s=(a*x+b)/(a*x-b);
    printf("The value of s=%f",s);
    getch();
}
```

Input:enter the values of a,b,x

1 3 2

Output:

The value of s= 5

Record at least 3 results

Signature of faculty with date

Week: 3

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

AIM: To Check whether given number is perfect number or not

Algorithm:**Perfect number**

Step 1: read n

Step 2: assign i=1,sum=0

Step 3: while(i<n) goto step 4

Step 4: if(n%i==0)

 sum=sum+i

 i++

step 5: if(sum==n) print given number is perfect number otherwise not a perfect number.

Program:

```
#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;
}
```

Input:

Enter a number:6

Output:

6 is a perfect number

3) b) Write a C program to check whether a number is strong number or not.

AIM: To check whether given number is strong number or not

Algorithm:

Strong number

Step 1: read num,i,f,r,sum=0,temp

Step 2: assign num to temp

Step 3: while(num) goto step 4

Step 4: i=1,f=1

 r=num%10

 while(i<=r) goto step 5

Step 5: f=f*i

 i=i+1

Step 6: sum=sum+f;

Step 7: num=num/10;

Step 8: if sum and temp are equal got step 9

Step 9: print strong number otherwise not a strong number

Program:

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

Input:

Enter a number: 145

Output:

145 is a strong number

Record at least 3 results

Signature of faculty with date

Week: 4

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

AIM: To find the roots of a quadratic equation.

Description: roots of quadratic equation are $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

ALGORITHM:

Step 1: Start

Step 2: Read a,b,c

Step 3: calculate disc = $b*b-4*a*c$

Step 4: if(disc>0)

Begin

Step 5: root1= $(-b+\sqrt{\text{disc}})/(2*a)$

Step 6: root2= $(-b-\sqrt{\text{disc}})/(2*a)$

Step 7: Print "Root1", "Root2"

End

Step 8: else if(disc=0)

Begin

Step 9: root1= $-b/(2*a)$

Step 10: root2=root1;

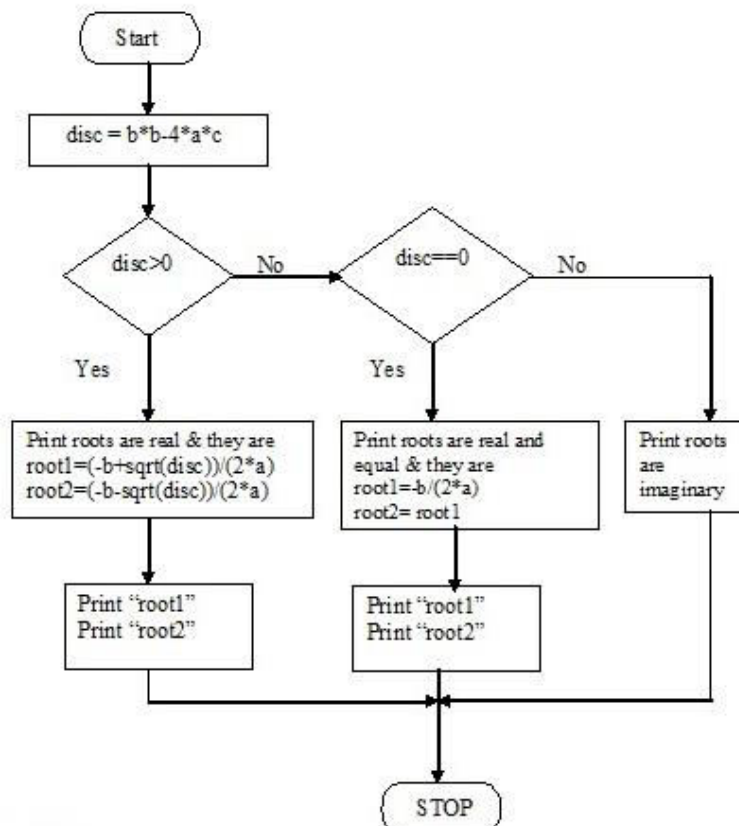
Step 11: Print "Root1", "Root2"

End

Step 12: else

Step 13: Print Roots are imaginary

Step 14: Stop

Flow Chart

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
    int a,b,c;
    float disc,root1,root2;
    float img,real;
    printf("ENTER VALUES FOR a,b,c:\n");
    scanf("%d%d%d",&a,&b,&c);
    disc=(float)b*b-4*a*c;
    if(disc>0)
    {
        printf("THE ROOTS ARE REAL & UNEQUAL:\n");
        root1=(-b+sqrt(disc))/(2*a);
        root2=(-b-sqrt(disc))/(2*a);
        printf("Root1=%f\n",root1);
        printf("Root2=%f\n",root2);
    }
    else if(disc==0)
    {
        printf("THE ROOTS ARE REAL AND EQUAL:\n");
        root1=-b/(2*a);
        root2=root1;
        printf("Root1=%f\n",root1);
        printf("Root2=%f\n",root2);
    }
    else
    {
        printf("THE ROOTS ARE IMAGINARY:\n");
        disc=-disc;
        img=(float)disc/2*a;
        real=(float)-b/2*a;
        if (img>0)
        {
            printf("Root1=%f + i%f\n",real,img);
            printf("Root2=%f - i%f\n",real,img);
        }
        else
        {
            img=-img;
            printf("Root1=%f + i%f\n",real,img);
            printf("Root2=%f - i%f\n",real,img);
        }
    }
    return 0;
}
```


INPUT:

ENTER VALUES FOR a,b,c

1 4 4

OUTPUT:

THE ROOTS ARE EQUAL AND THEY ARE.. Root1=-2 Root2=-2

Record at least 3 results**Signature of faculty with date**

4) b). Write a C program which takes two integer operands and one operator from the user, performs the operation and then prints the result.(Consider the operators +,-,*,/,% and use Switch Statement.)

AIM:

To perform arithmetic operations using switch statement.

Algorithm:

Step 1: Read a,b

Step 2: Print "Menu Options"

Step 3: do

Begin Step 4: Read ch

Step 5: switch(ch)

Begin Step 6:

case 1:

Begin

Calculate $c = a + b$

Print "c"

break;

End

case 2:

Begin

Calculate $c = a - b$

Print "c"

break;

End

case 3:

Begin

Calculate $c = a * b$

Print "c"

break;

End

case 4:

Begin

Calculate $c = a / b$

Print "c"

break;

End

case 5:

Begin

Calculate $c = a \% b$

Print "c"

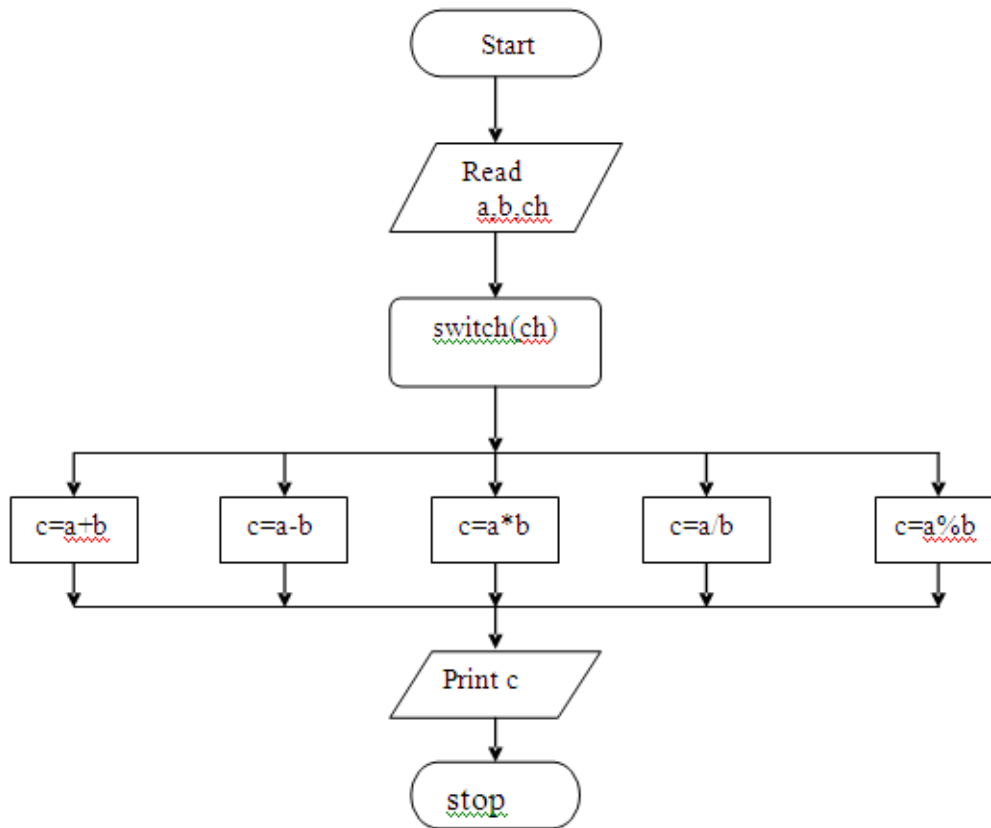
break;

End

default:

Print "Invalid choice"

End

Flowchart**Program:**

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c,ch; clrscr();
    printf("ENTER TWO VALUES FOR a & b\n"); scanf("%d %d",&a,&b);
    while(1) {
        printf("MENU OPTIONS \n");
        printf("*****\n");
        printf("1.Addition\n");
        printf("2.Subtraction\n");
        printf("3.Multiplication\n");
        printf("4.Division\n");
        printf("5.Modulus\n");
        printf("6.Exit\n");
        printf("\n");
        printf("ENTER UR CHOICE\n");
        scanf("%d",&ch);
        switch(ch) {

```

```
case 1: c=a+b;
printf("The addition of %d and %d is..%d\n",a,b,c); break;
case 2: c=a-b;
printf("The subtraction of %d and %d is..%d\n",a,b,c); break;
case 3: c=a*b;
printf("The multiplication of %d and %d is..%d\n",a,b,c); break;
case 4: c=a/b;
printf("The division of %d and %d is..%d\n",a,b,c); break;
case 5: c=a%b;
printf("The modulus of %d and %d is..%d\n",a,b,c); break;
case 6:exit(0); default:printf("INVALID CHOICE\n"); }
}
getch();
}
```

INPUT:

ENTER TWO VALUES FOR a & b: 20 16

OUTPUT:

MENU OPTIONS

1.Addition 2.Subtraction 3.Multiplication 4.Division 5.Modulus
6.Exit

ENTER UR CHOICE 1

The addition of 20 and 16 is..36

Record at least 3 results

Signature of faculty with date

Week: 5

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

AIM:

To find the factorial of a given number using non-recursive function.

ALGORITHM:

Step 1: Start

Step 2: Read n

Step 3: Call fact(n) goto step 6

Step 4: Store result in “f”

Step 5: Print “f” goto step 10

Step 6: Begin //sub program

Initialize $f \leftarrow 1$

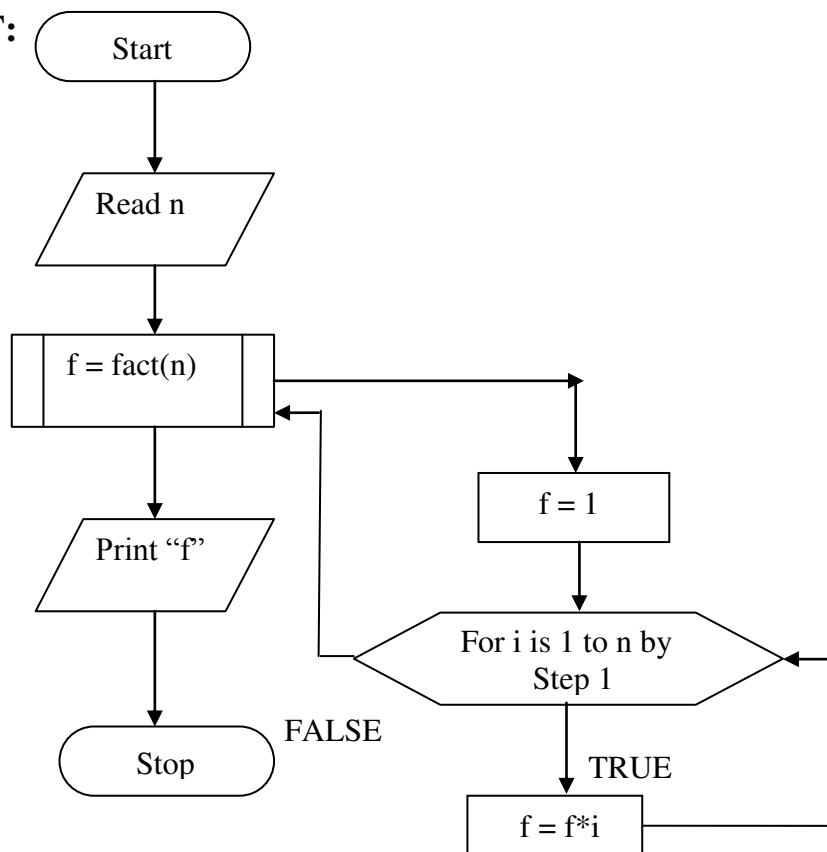
Step 7: for i is 1 to n by step 2

Step 8: Calculate $f = f * i$

Step 9: return “f”

End

Step 10: Stop

FLOWCHART:

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int fact(int);
void main()
{
    int n,i,f;
    clrscr();
    printf("ENTER A VALUE FOR n:\n");
    scanf("%d",&n);
    f=fact(n);
    printf("THE FACTORIAL OF A GIVEN NO IS..%d",f);
    getch();
}
int fact(int n)
{
    int i,f=1;
    for(i=1;i<=n;i++)
        f=f*i;
    return(f);
}
```

INPUT:

ENTER A VALUE FOR n

5

OUTPUT:

THE FACTORIAL OF A GIVEN NUMBER IS..120

Record at least 3 results

Signature of faculty with date

5) b) Write a C program to find the factorial of a given integer using recursive function.

AIM:

To find the factorial of a given number using recursive function.

ALGORITHM:

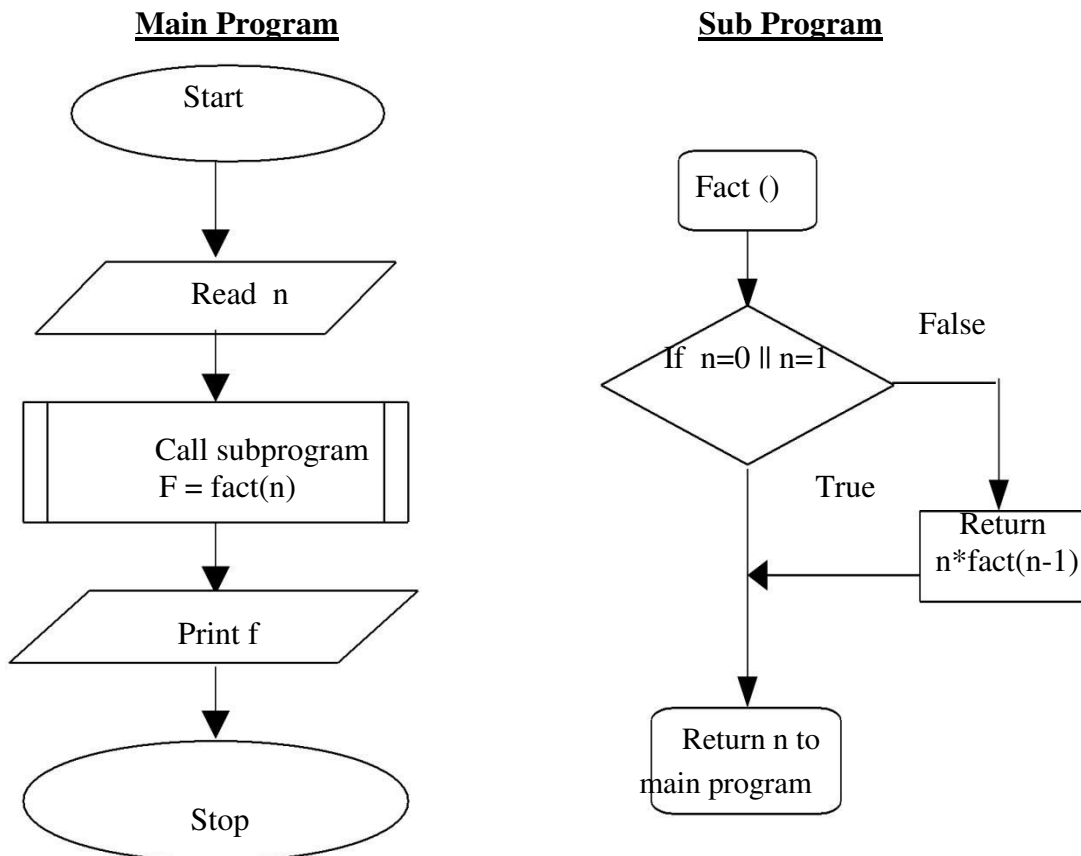
main program

- Step 1: start
- Step 2: read n
- Step 3: call sub program as $f = \text{fact}(n)$
- Step 4: print f value
- Step 5: stop

Sub program:

- Step 1: initialize the f
- Step 2: if $n = 0$ or $n == 1$ return 1 to main program if not goto step 3
- Step 3: return $n * \text{fact}(n-1)$ to main program

FLOW CHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int fact(int);
void main()
{
    int n,res;
    clrscr();
    printf("ENTER A NUMBER:\n");
    scanf("%d",&n);
    res=fact(n);
    printf("THE FACTORIAL OF A GIVEN NUMBER IS..%d",res);
    getch();
}
int fact(int n)
{
    int r;
    if(n==0)
        return(1);
    else
    {
        r=n*fact(n-1);
        return(r);
    }
}
```

INPUT:

ENTER A VALUE FOR n
5

OUTPUT:

THE FACTORIAL OF A GIVEN NUMBER IS..120

Record at least 3 results

Signature of faculty with date

Week: 6

6) a) Write a C program to find the GCD of two given integers by using the recursive function

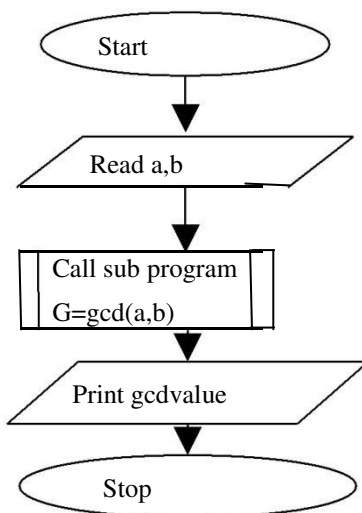
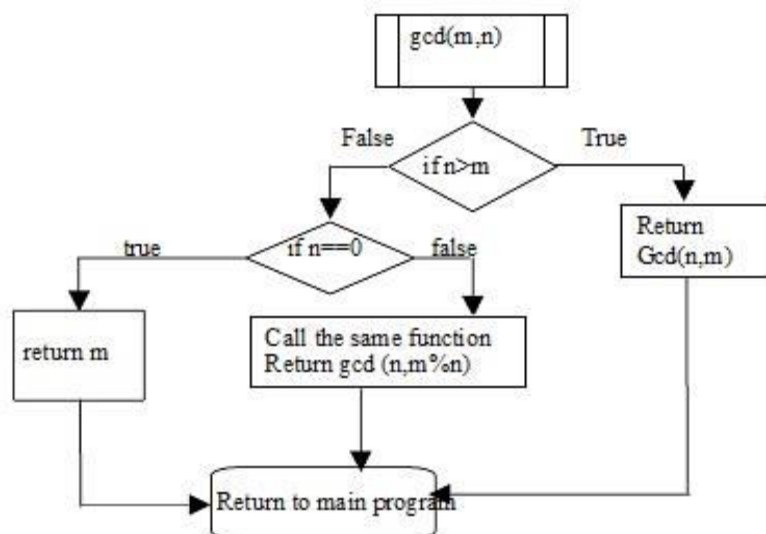
Aim: To find the Gcd of two given integers by using the recursive function

Algorithm:**Main program:**

Step 1: start
 Step 2: read a,b
 Step 3: call the sub program GCD(a,b) for print the value
 Step 4: stop

Sub program: GCD(n,m)

Step 1: if $n > m$ return GCD(n,m)
 Step 2: if $n == 0$ return m else goto step 3
 Step 3: return GCD (n,m%n)
 Step 4: return to main program

Flow CHART:**Main Program:****Sub Program:**

Program:

```
#include<stdio.h>
#include<conio.h>
int gcdrecursive(int m,int n)
{
    if(n>m)
        return gcdrecursive(n,m);
    if(n==0)
        return m;
    else
        return gcdrecursive(n,m%n);    // return to the main program
}
void main()
{
    int a,b,igcd; clrscr();
    printf("enter the two numbers whose gcd is to be found:");
    scanf("%d%d",&a,&b);
    printf("GCD of a,b is %d",gcdrecursive(a,b)); // return to the sub program getch();
}
```

Input:

Enter the two numbers whose gcd is to be found: 5 25

Output:

GCD of a,b is : 5

Record at least 3 results

Signature of faculty with date

6) b) Write a C program to find the GCD of two given integers using non-recursive function.

To find the GCD of two given integers by using the non recursive function

Description:

GCD means Greatest Common Divisor. i.e the highest number which divides the given number

Ex: GCD(12,24) is 12

Formula: $\text{GCD} = \text{product of numbers} / \text{LCM of numbers}$

Algorithm:

Step 1: start

Step 2: read a,b

Step 3: call sub program $g = \text{GCD}(a,b)$

Step 4: print the g value

Step 5: stop

Sub program:

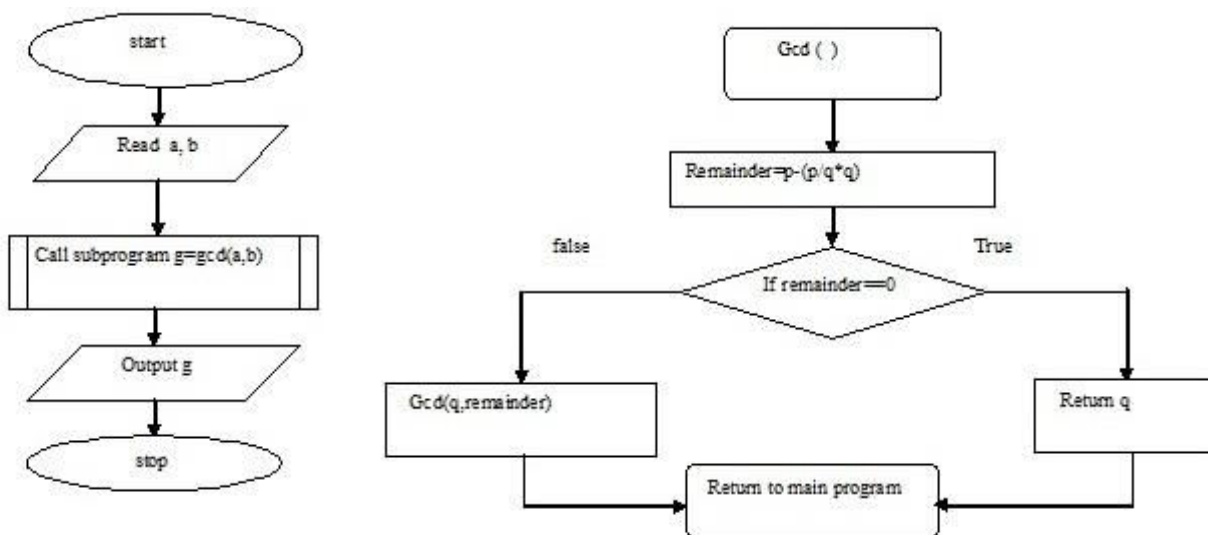
Step 1: initialize the $p=1, q, \text{remainder}$

Step 2: $\text{remainder} = p - (p/q * q)$

Step 3: $\text{remainder} = 0$ return q else goto step 4

Step 4: $\text{GCD}(q, \text{remainder})$ return to main program

Flowchart:



Program:

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int gcdnonrecursive(int m,int n)
{
    int remainder;
  
```

```
    remainder=m-(m/n*n);
    if(remainder==0)
    return n;
    else
    gcdnonrecursive(n,remainder);
}

void main()
{
    int a,b,igcd;
    clrscr();
    printf("enter the two numbers whose gcd is to be found:");
    scanf("%d%d",&a,&b);
    printf("GCD of %d",gcdnonrecursive(a,b));
    getch();
}
```

Output:

1. enter the two numbers whose gcd is to be found:5,25
GCD of a,b is : 5

Record at least 3 results

Signature of faculty with date

Week: 7

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

AIM:

To find the largest and smallest number in a list of integers.

ALGORITHM:

Step 1: start

Step 2: read n

Step 3: initialize i=0

Step 4: if i<n do as follows. If not goto step 5

 Read a[i]

 Increment i

 Goto step 4

Step 5: small=a[0], large=a[0]

Step 6: initialize i=0

Step 7: if i<n do as follows. If

 not goto step 8

 If a[i]<small

 Assign small=a[i]

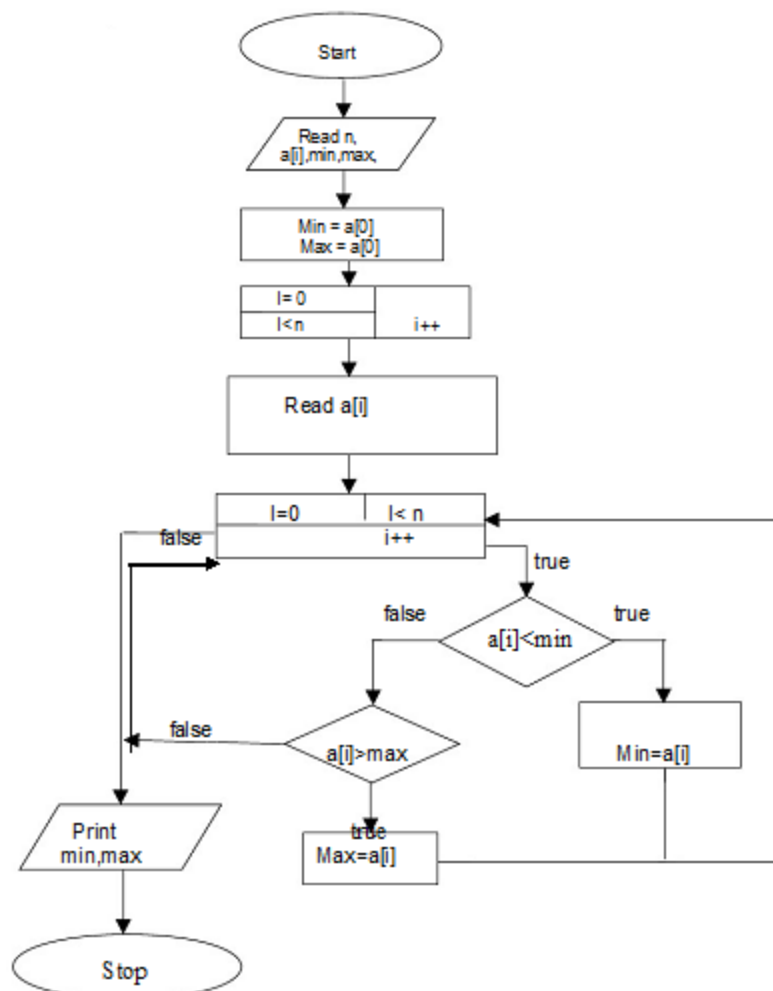
 If a[i]>large

 Assign large=a[i]

 Increment i goto Step 7

Step 8: print small, large

Step 9: stop



Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{ int a[10],i,n,small,large;
  clrscr();
  printf("Enter The Array Size:");
  scanf("%d",&n);
  printf("ENTER ELEMENTS OF ARRAY");
  for(i=0;i<n;i++) // read the elements of an array
  scanf("%d",&a[i]);
  small=a[0];
  large=a[0];
  for(i=0;i<n;i++)// read the elements of an array
  {      if(a[i]<small)// check the condition for minimum value
        small=a[i];
        if(a[i]>large)//check the condition for maximum value
        large=a[i];
  }
  printf("largest value is:%d\n",large);
  printf("smallest value is:%d\n",small);
  getch();
}
```

INPUT:

Enter The Array Size:10

ENTER THE ELEMENTS OF ARRAY

7 10 9 8 6 5 2 3 4 1

OUTPUT:

largest value is : 10

smallest value is : 1

Record at least 3 results

Signature of faculty with date

7) b) Write a C Program to Sort the Array in an Ascending Order.**Program:**

C Program to Sort the Array in an Ascending Order

#include <stdio.h>

void main()

```
{  
  
    int i, j, a, n, number[30];  
  
  
    printf("Enter the value of N \n");  
  
    scanf("%d", &n);  
  
    printf("Enter the numbers \n");  
  
    for (i = 0; i < n; ++i)  
        scanf("%d", &number[i]);  
  
    for (i = 0; i < n; ++i)  
    {  
        for (j = i + 1; j < n; ++j)  
        {  
            if (number[i] > number[j])  
            {  
                a = number[i];  
                number[i] = number[j];  
                number[j] = a;  
            }  
        }  
    }  
}
```

```
printf("The numbers arranged in ascending order are given below \n");
```

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

```
    printf("%d\n", number[i]);
```

```
}
```

Output:

Enter the value of N

6

Enter the numbers

3

78

90

456

780

200

The numbers arranged in ascending order are given below

3

78

90

200

456

780

Record at least 3 results

Signature of faculty with date

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

Program:

```
#include<conio.h>
#include<stdio.h>
void main()
{
    int a[10][10],i,j,m;
    clrscr();
    printf("Enter order of square matrix: ");
    scanf("%d",&m);
    for(i=0;i<m;i++)
    {
        for(j=0;j<m;j++)
        {
            printf("Enter value of a[%d][%d]: ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<m;j++)
        {
            if(a[i][j]!=a[j][i])
            {
                printf("\n\nMatrix is not symmetric");
                getch();
                exit(0);
            }
        }
    }
    printf("\n\nMatrix is symmetric");
    getch();
}
```

Output:

```
Enter order of square matrix:2
Enter value of a[0][0]:12
Enter value of a[0][1]:34
Enter value of a[1][0]:34
Enter value of a[1][1]:54
Matrix is symmetric
```

Week : 9**9) a) Write a C program to perform addition of two matrices.****AIM:**

To perform addition of two matrices.

ALGORITHM:

Step 1: Start

Step 2: for i is 0 to 2 by step 1
for j is 0 to 2 by step 1

Step 3: Read $a[i][j], b[i][j]$

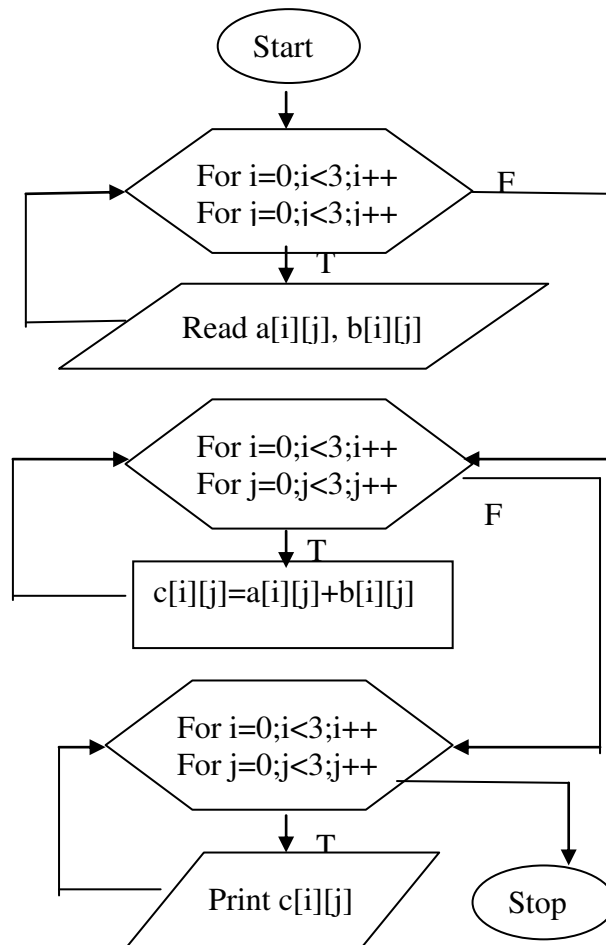
Step 4: goto step 2

Step 5: calculate $c[i][j] = a[i][j] + b[i][j]$

Step 6: goto step 2

Step 7: Print $c[i][j]$

Step 8: Stop

Flow Chart:

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3],b[3][3],c[3][3];
int i,j;

clrscr();
printf("ENTER A MATRIX\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
        scanf("%d",&a[i][j]);
}
printf("ENTER B MATRIX\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
        scanf("%d",&b[i][j]);
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
        c[i][j]=a[i][j]+b[i][j];
}
printf(" After addition of two matrices :\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d\t",c[i][j]);
    }
    printf("\n");
}
getch();
}
```

INPUT:

ENTER a MATRIX

1 2 3

4 5 6

7 8 9

ENTER b MATRIX

1 1 1

1 1 1

1 1 1

OUTPUT:

After addition of two matrices is..

2 3 4

5 6 7

8 9 10

Record at least 3 results**Signature of faculty with date**

9) b) Write a C program that uses functions to perform Multiplication of Two Matrices.**AIM:**

To perform multiplication of two matrices.

ALGORITHM:

Step 1: Start

Step 2: for i is 0 to 2 by step 1

for j is 0 to 2 by step 1

Step 3: Read a[i][j], b[i][j]

Step 4: goto step 2

Step 5: calculate $c[i][j] = c[i][j] + a[i][k] * b[k][j]$

Step 6: goto step 2

Step 7: Print c[i][j]

Step 8: Stop

Program:

```
#include<stdio.h >
#include<conio.h>
int i,j,k;
void main()
{
    int a[10][10],b[10][10],c[10][10],m,n,p,q;
    void mul(int x[10][10],int y[10][10],int z[10][10],int m,int n,int p,int q);
    void read(int x[10][10],int m,int n);
    void display(int x[10][10], int m,int n);
    clrscr();
    printf("Enter the size of A Mtrix (Row and Col): \n");
    scanf("%d%d",&m,&n);
    printf("Enter the size of B Mtrix (Row and Col): \n");
    scanf("%d%d",&p,&q);
    if(n!=p)
    {
        printf("Multiplication Not Possible\n Please re-enter\n");
        printf("correct size and try again ..... \n");
    }
    else
    {
        read(a,m,n);
        read(b,m,n);
        mul(a,b,c,m,n,p,q);
        printf("A Matrix is : \n");
        display(a,m,n);
    }
}
```

```
printf("B Matrix is :\n");

display(b,m,n);
printf("C Matrix is :\n");
display(c,m,n);
}
getch();
}

void mul(int x[10][10],int y[10][10],int z[10][10],int m,int n,int p,int q)
{
    for (i=0;i<m;i++)
        for(j=0;j<q;j++)
        {
            z[i][j]=0;
            for(k=0;k<n;k++)
                z[i][j]+= x[i][k]*y[k][j];
        }
}

void read(int x[10][10], int m,int n)
{
    printf("Enter Matrix Value Row by Row\n");
    for (i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&x[i][j]);
}

void display(int x[10][10], int m,int n)
{
    for (i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
            printf("%5d",x[i][j]);
        printf("\n");
    }
    printf("\n");
}
```

Input:

Enter the size of A Mtrix (Row and Col): 2 2

Enter the size of B Mtrix (Row and Col): 2 2

Enter Matrix Value Row by Row

1 0

2 6

Enter Matrix Value Row by Row

3 4

4 2

Output:

A matrix is:

1 0

2 6

B Matrix is:

3 4

4 2

C matrix is:

3 4

24 20

Record at least 3 results

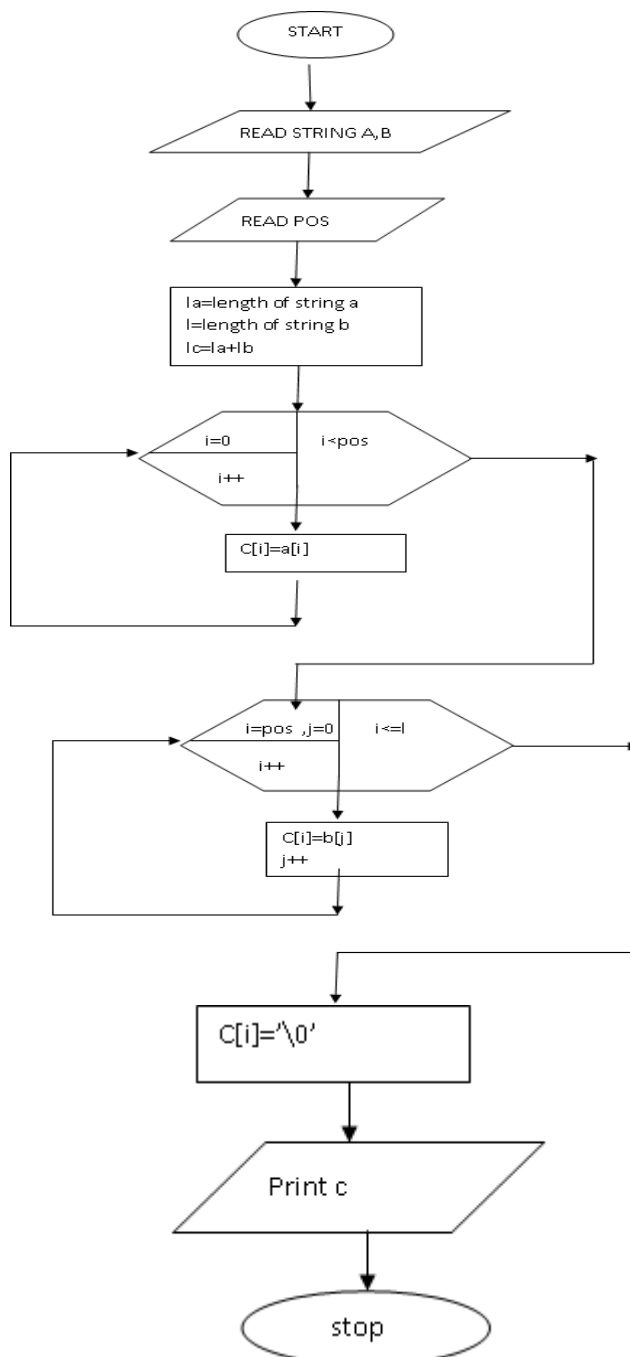
Signature of faculty with date

Week: 10

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

Aim:

To insert a string into another string from a specified position.

Flow Chart :

Algorithm:

- Step 1: start
- Step 2: read main string and sub string
- Step 3: find the length of main string(r)
- Step 4: find length of sub string(n)
- Step 5: copy main string into sub string
- Step 6: read the position to insert the sub string(p)
- Step 7: copy sub string into main string from position p-1
- Step 8: copy temporary string into main string from position p+n-1
- Step 9: print the strings
- Step 10: stop

Program:

```
#include<stdio.h>
#include<string.h>
main()
{
char a[300],b[30],c[30];
int pos=0,i=0,l,la,lb,lc,j;
    puts("Enter a string");
    gets(a);
    puts("Enter sub string");
    gets(b);
    puts("enter position for insertion");
    scanf("%d",&pos);
    la=strlen(a);
    lb=strlen(b);
    l=pos+lb;
    lc=la+lb;
    for(i=0;i<pos;i++)
    {
        c[i]=a[i];
    }
    j=0;
    for(i=pos;i<=l;i++)
    {
        c[i]=b[j];
        j++;
    }
    j=pos;
    for(i=l;i<lc;i++)
    {
        c[i]=a[j];
        j++;
    }
    c[i]='\0';
    puts("String after Insertion is:");
    printf("%s",c);
}
```

```
}
```

Input:

Enter First String:

Comer

Enter Second String:

put

Output:

Enter the position where the item has to be inserted:3

Computer

Record at least 3 results

Signature of faculty with date

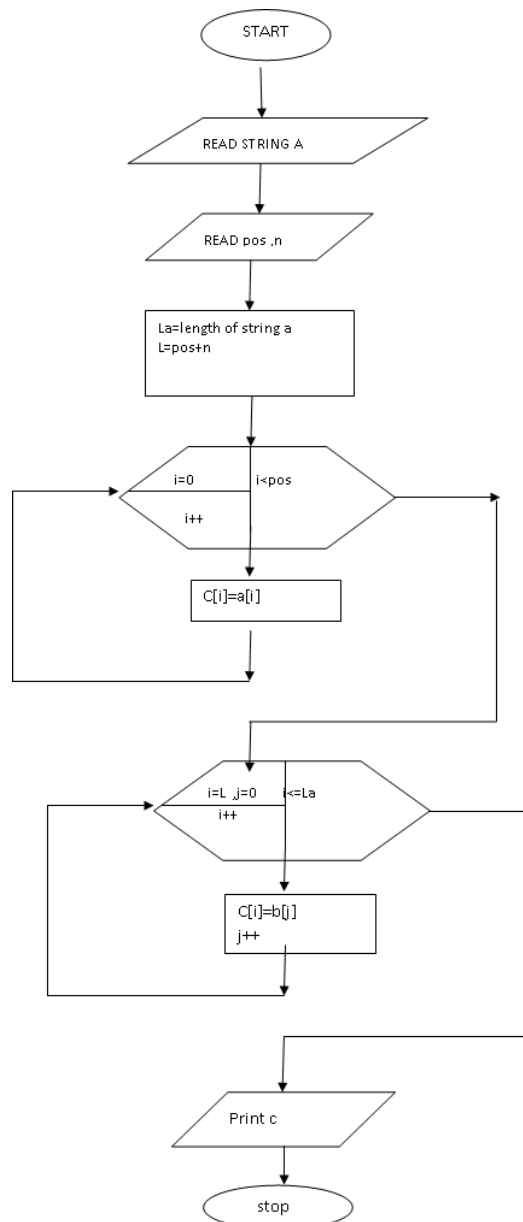
10) b) To delete n Characters from a given position in a given string.

Aim: To delete n Characters from a given position in a given string.

Algorithm:

- Step 1: start
- Step 2: read string
- Step 3: find the length of the string
- Step 4: read the value of number of characters to be deleted and positioned
- Step 5: string copy part of string from position to end, and
(position + number of characters to end)
- Step 6: stop

Flow Chart:



Program:

```
#include<stdio.h>
#include<string.h>
main()
{
char a[30],c[30];
int pos=0,i=0,L,La,j,n;
puts("Enter a string");
gets(a);
puts("enter position for deletion");
scanf("%d",&pos);
puts("Enter number of characters to be deleted");
scanf("%d",&n);
La=strlen(a);
L=pos+n;
for(i=0;i<pos;i++)
{
c[i]=a[i];
}
j=pos;
for(i=L;i<=La;i++)
{
c[j]=a[i];
j++;
}
puts("String after Deletion is:");
printf("%s",c);
}
```

Input:

Enter the string
jayapal

Enter the position from where to delete:4
Enter the number of characters to be deleted 2

Output:

jayal

Record at least 3 results

Signature of faculty with date

Week: 11

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

Aim: To determine if the given string is palindrome or not.

Description :

Palindrome means string on reversal should be same as original

Ex: madam on reversal is also madam

Algorithm:

Step 1: start

Step 2: read string A

Step 3: copy string A into B

Step 4: reverse string B

Step 5: compare A & B

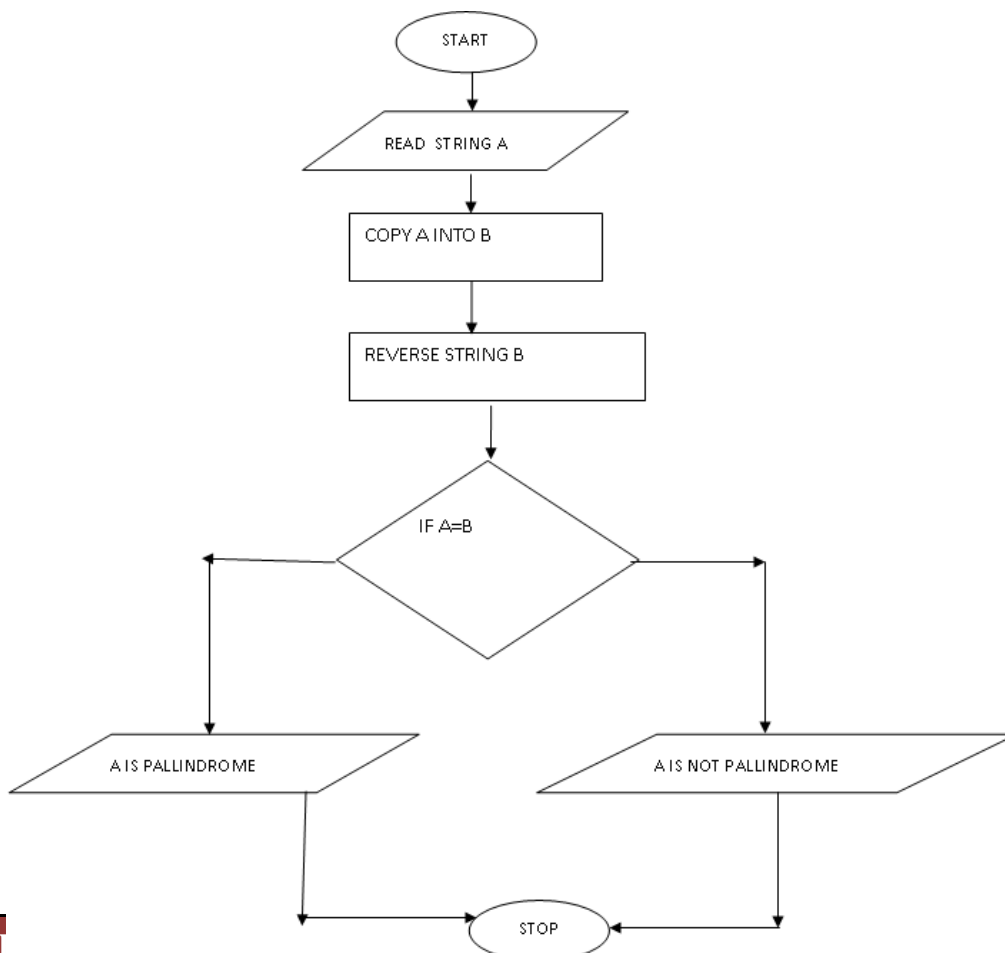
If A equals B to go to step 6

Else goto step 7

Step 6: print given string A is pallindrom

Step 7: print given string is not pallindroma

Step 8: stop

Flow Chart:

Program:

```
#include <stdio.h>
#include <string.h>
void main()
{
    char string[25], reverse_string[25] = {'\0'};
    int i, length = 0, flag = 0;

    printf("Enter a string \n");
    gets(string);
    for (i = 0; string[i] != '\0'; i++)
    {
        length++;
    }
    printf("The length of the string '%s' = %d\n", string, length);
    for (i = length - 1; i >= 0 ; i--)
    {
        reverse_string[length - i - 1] = string[i];
    }

    for (flag = 1, i = 0; i < length ; i++)
    {
        if (reverse_string[i] != string[i])
            flag = 0;
    }
    if (flag == 1)
        printf ("%s is a palindrome \n", string);
    else
        printf ("%s is not a palindrome \n", string);
}
```

Input:

Enter a string
madam

Output:

The length of the string 'madam' = 5
madam is a palindrome

Record at least 3 results

Signature of faculty with date

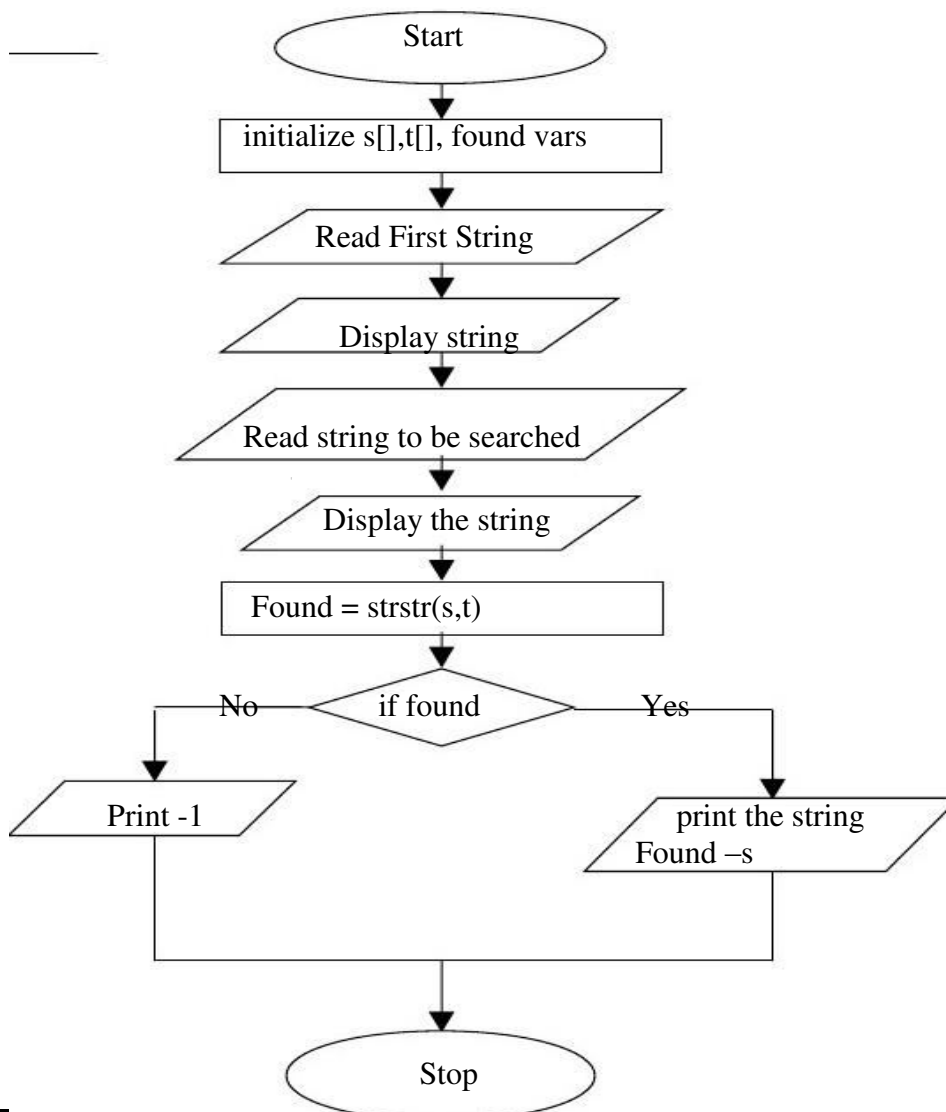
11. b) Write a C program that displays the position or index in the string S where the string T begins, or - 1 if S doesn't contain T.

Aim: To display the position or index in the string S where the string T begins, or - 1 if S doesn't contain T

Algorithm:

- Step 1: start
 Step 2: read the string and then displayed
 Step 3: read the string to be searched and then displayed
 Step 4: searching the string T in string S and then perform the following steps
 i. found=strstr(S,T)
 ii. if found print the second string is found in the first string at the position. If not goto step 5
 Step 5: print the -1
 Step 6: stop

Flow Chart:



Program:

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

Input:

Enter the first string:
computer
Enter the string to be seareched:
mp

Output:

Second string is found in the first string at 2 position

Record at least 3 results

Signature of faculty with date

Week: 12

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

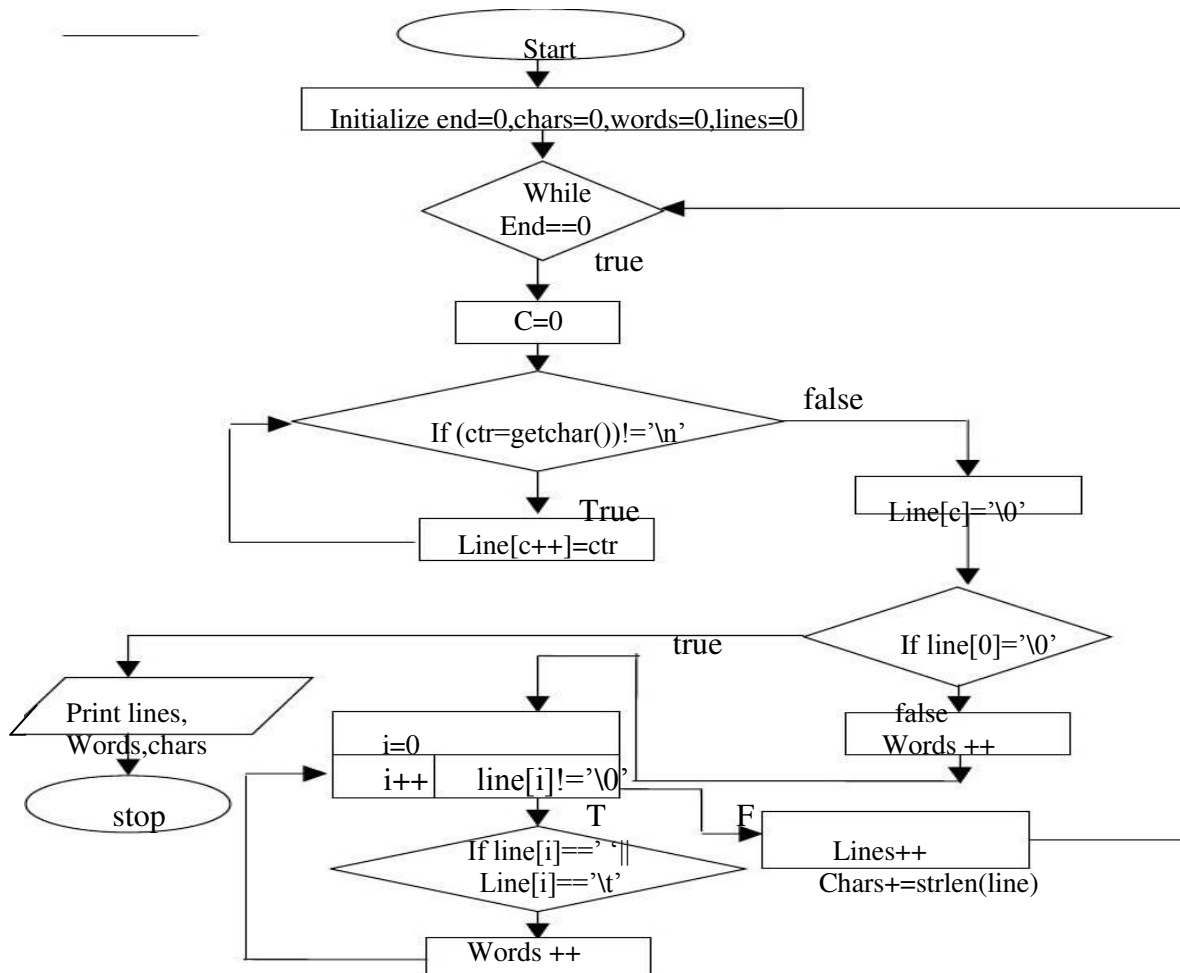
AIM:

To count the number of lines, words and characters in a given list.

ALGORITHM:

- Step 1: Start
- Step 2: Read the text until an empty line
- Step 3: Compare each character with newline char '\n' to count no of lines
- Step 4: Compare each character with tab char '\t' or space char ' ' to count no of words
- Step 5: Compare first character with NULL char '\0' to find the end of text
- Step 6: No of characters = length of each line of text
- Step 7: Print no of lines, no of words, no of chars
- Step 8: Stop.

Flow Chart:



PROGRAM:

```
#include <stdio.h>
void main()
{
    char line[81], ctr;
    int i,c,
        end = 0,
        characters = 0,
        words = 0,
        lines = 0;
    printf("TYPE ANY TEXT.\n");
    printf("GIVE ONE SPACE AFTER EACH WORD.\n");
    while( end == 0)
    {
        /* Reading a line of text */
        c = 0;
        while((ctr=getchar()) != '\n')
            line[c++] = ctr;
        line[c] = '\0';
        /* counting the words in a line */
        if(line[0] == '\0')
            break ;
        else
        {
            words++;
            for(i=0; line[i] != '\0';i++)
                if(line[i] == ' ' || line[i] == '\t')
                    words++;
        }
        /* counting lines and characters */
        lines = lines +1;
        characters = characters + strlen(line);
    }
    printf ("\n");
    printf("Number of lines = %d\n", lines);
    printf("Number of words = %d\n", words);
    printf("Number of characters = %d\n", characters);
}
```

INPUT:

TYPE ANY TEXT

GIVE ONE SPACE AFTER EACH WORD.

Ramu is a good boy.

OUTPUT:

THE NUMBER OF CHARACTERS IN A GIVEN TEXT IS..18

THE NUMBER OF WORDS IN A GIVEN TEXT IS..5

THE NUMBER OF LINES IN A GIVEN TEXT IS..1

Record at least 3 results

Signature of faculty with date

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

Program:

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

int string_ln(char*);

void main() {
    char str[20];
    int length;
    clrscr();

    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;
}
```

Input:

Enter the String : pritesh

Output:

Length of the given string pritesh is : 7

Record at least 3 results

Signature of faculty with Date

Week: 13

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

AIM:

To write a C program to find sum of n elements entered by user. To perform this program, allocate memory dynamically using calloc() function.

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

int main()
{
    int i, n;
    int *a;

    printf("Number of elements to be entered:");
    scanf("%d",&n);

    a = (int*)calloc(n, sizeof(int));
    printf("Enter %d numbers:\n",n);
    for( i=0 ; i < n ; i++ )
    {
        scanf("%d",&a[i]);
    }

    printf("The numbers entered are: ");
    for( i=0 ; i < n ; i++ )
    {
        printf("%d ",a[i]);
    }

    return(0);
}
```

Output:

```
Number of elements to be entered:3
Enter 3 numbers:
22
55
14
The numbers entered are: 22 55 14
```

Record at least 3 results

Signature of faculty with Date

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

Program:

```
#include<stdio.h>
#include<conio.h>
struct student
{
    int rl;
    char nm[20];
    int m1;
    int m2;
    int m3;
    int t;
    float per;
};
void main()
{
    struct student a;
    clrscr();
    printf(" Enter RollNo, Name amd three sub marks\n");
    scanf("%d%s%d%d%d",&a.rl,&a.nm,&a.m1,&a.m2,&a.m3);
    a.t=a.m1+a.m2+a.m3;
    a.per=a.t/3.0;
    printf("rollno=%d\n",a.rl);
    printf("Name=%sk\n",a.nm);
    printf("m1=%d\n",a.m1);
    printf("m2=%d\n",a.m2);
    printf("m3=%d\n",a.m3);
    printf("total=%d\n",a.t);
    printf("per=%f\n",a.per);
    getch();
}
```

Input:

Enter RollNo, Name and three sub marks

12 rama 30 40 50

Output:

rollno=12

Name=rama

m1=30

m2=40

m3=50

total=120

per=40.000000

Week: 14

14) a) Write a C program that uses functions 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**

AIM: To perform arithmetic operations on complex numbers

Complex numbers of type a+ib

Addition: $(a+ib)+(x+iy)=a+x+i(b+y)$

Subtraction: $(a+ib)-(x+iy)=a-x+i(b-y)$

Multiplication: $(a+ib)*(x+iy)=ax-by+i(ay+bx)$

Division

$$(a+ib)/(x-iy) = \frac{a+ib}{x+iy} * \frac{x-iy}{x-iy} = \frac{(a+ib)*(x-iy)}{x^2+y^2} = \frac{(ax+by)+i(bx-ay)}{x^2+y^2} = \frac{ax+by}{x^2+y^2} + i \frac{bx-ay}{x^2+y^2}$$

ALGORITHM:

Step 1: start
 Step 2: Read Two complex numbers c1 ,c2
 Step 3: c3=c1+c2
 Step 4: print c3
 Step 5: c3=c1-c2
 Step 6: print c3
 Step 7: c3=c1*c2
 Step 8: print c3
 Step 9: c3=c1/c2
 Step 10: print c3
 Step 11: print c
 Step 12: stop

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
struct complex
{
float real,img;
};
```

```
/*code for reading complex number*/
struct complex read_complex()
{
```

```
struct complex c;
    printf("enter real part of complex number");
    scanf("%f",&c.real);
    printf("enter Imaginary part of complex number");
    scanf("%f",&c.img);
    return c;
}

/*code for adding complex numbers*/
struct complex add_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real+c2.real;
    c3.img=c1.img+c2.img;
    return c3;
}

/*code for subtraction of complex numbers*/
struct complex sub_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real-c2.real;
    c3.img=c1.img-c2.img;
    return c3;
}

/*code for multiplication of complex numbers*/
struct complex mul_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real=c1.real*c2.real-c1.img*c2.img;
    c3.img= c1.img*c2.real+c2.img*c1.real;
    return c3;
}

/*code for division of complex numbers*/
struct complex div_complex(struct complex c1,struct complex c2)
{
    struct complex c3;
    c3.real= (c1.real*c2.real+c1.img*c2.img)/(c2.real*c2.real+c2.img*c2.img);
    c3.img= (c1.img*c2.real-c1.real*c2.img)/(c2.real*c2.real+c2.img*c2.img);
    return c3;
}

/*code for display of complex number*/
void display_complex(struct complex c)
{
    char sign;
    printf("The result is:");
    if(c.img<0)
```



```
{
    sign='-';
    c.img=-c.img;
}
else
    sign='+';
printf("%5f%ci%5f",c.real,sign,c.img);

}

int main()
{
int choice;
struct complex a,b,c;
while(1)
{
printf("\n-----\n");
printf("Menu for operation complex numbers\n ");
printf("-----\n");
printf("1.Addition \n ");
printf("2.Subtraction \n ");
printf("3.Multiplication \n ");
printf("4.Division \n ");
printf("5.Clear Screen \n ");
printf("6.Exit Menu  \n ");
printf("Enter Your Choice: ");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("You Have Selected Addition operation on complex NUmbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=add_complex(a,b);
        display_complex(c);
        break;
case 2:printf("You Have Selected Subtraction operation on complex NUmbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=sub_complex(a,b);
        display_complex(c);
        break;
case 3:printf("You Have Selected Multiplication operation on complex Numbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
```

```
        c=mul_complex(a,b);
        display_complex(c);
        break;
case 4:printf("You Have Selected Division operation on complex Numbers\n");
        printf("Enter First complex number\n");
        a=read_complex();
        printf("Enter Second complex Number\n");
        b=read_complex();
        c=div_complex(a,b);
        display_complex(c);
        break;
case 5: clrscr();
        break;
case 6: exit(0);
        default:printf("Invalid choice");
        }
    }
}
```

Record at least 3 results

Signature of faculty with date

14)b) write a c program to display the contents of a file.

Aim: To display the contents of a file.

Program:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <process.h>
int main()
{
    FILE *fs;
    char ch;
    char *fname;

    printf("Enter the file name :");
    gets(fname);
    fs = fopen(fname,"r");
    if(fs==NULL)
    {
        puts("Source file cannot be opened.");
        getch();
    }
    else
    {
        while((ch=fgetc(fs))!=EOF)
        {
            putchar(ch);
        }
    }
    getch();
    return 0;
}
```

Input:

Enter the file name :sample.txt

Output:

this is my first program

Record at least 3 results

Signature of faculty with date

Week: 15

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

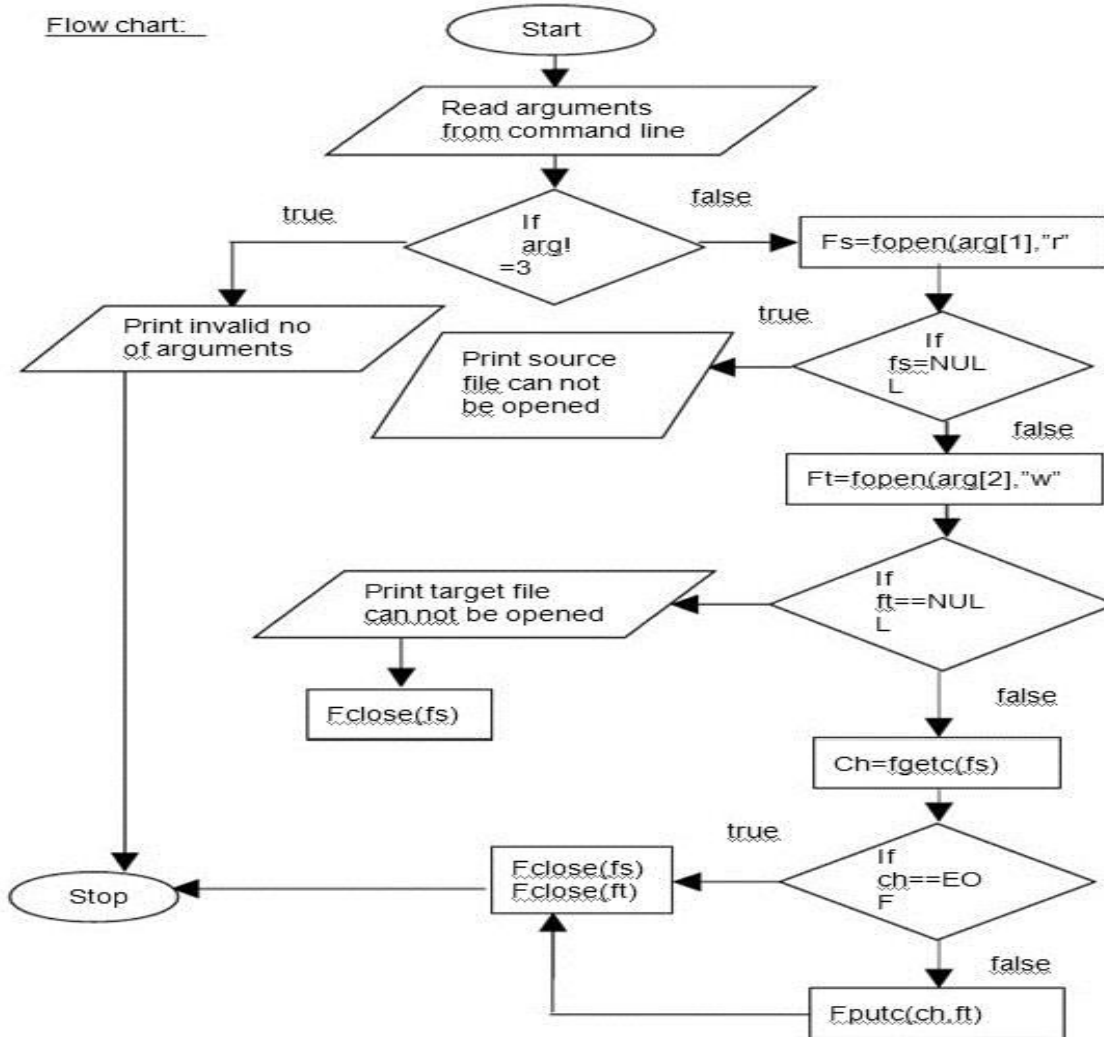
Aim:

Program which copies one file to another

Algorithm:

- Step 1: Start
- Step 2: read command line arguments
- Step 3: check if no of arguments =3 or not. If not print invalid no of arguments
- Step 4: open source file in read mode
- Step 5: if NULL pointer, then print source file can not be open
- Step 6: open destination file in write mode
- Step 7: if NULL pointer, then print destination file can not be open
- Step 8 : read a character from source file and write to destination file until EOF
- Step 9: Close source file and destination file
- Step 10: Stop

Flow chart:

**Program:**

```

#include<stdio.h>
#include<process.h>
#include<conio.h>
void main()
{
FILE *ft,*fs;
int c=0;
clrscr();
fs=fopen("a.txt","r");
ft=fopen("b.txt","w");
if(fs==NULL)
{
printf("Source file opening error\n");
exit(1);
}
else
if(ft==NULL)
{

```

```
        printf("Target file opening error\n");
        exit(1);
    }
    while(!feof(fs))
    {
        fputc(fgetc(fs),ft);
        c++;
    }
    printf("%d bytes copied from 'a.txt' to 'b.txt'",c);
    c=fcloseall();
    printf("%d files closed",c);
}
```

INPUT:

a.txt

An array is a collection of elements of similar datatypes

OUTPUT:

57 bytes copied from 'a.txt' to 'b.txt'

2 files closed

Record at least 3 results

Signature of faculty with date

15) b) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third.

Program :

```
#include<stdio.h>
#include<conio.h>
int main()
{
    FILE *fp1,*fp2,*fp3;
    char file1[20],file2[20],file3[20],ch;
    puts("Program to merge two files....\n");
    puts("Enter first file name:");
    gets(file1);
    puts("Enter Second file name:");
    gets(file2);
    puts("Enter Destination file name:");
    gets(file3);
    fp1=fopen(file1,"r");
    fp2=fopen(file2,"r");
    fp3=fopen(file3,"w");
    if(fp1==NULL&&fp2==NULL)
        printf("Error opening file1 and file2.....\n");
    else
    {
        if(fp3==NULL)
            printf("Error in creating destination file....\n");
        else
        {
            while((ch=fgetc(fp1))!=EOF)
                putc(ch,fp3);
            while((ch=fgetc(fp2))!=EOF)
                putc(ch,fp3);
        }
        printf("File Merging Sucessfull....");
        fcloseall();
        getch();
    }
}
```

Record at least 3 results

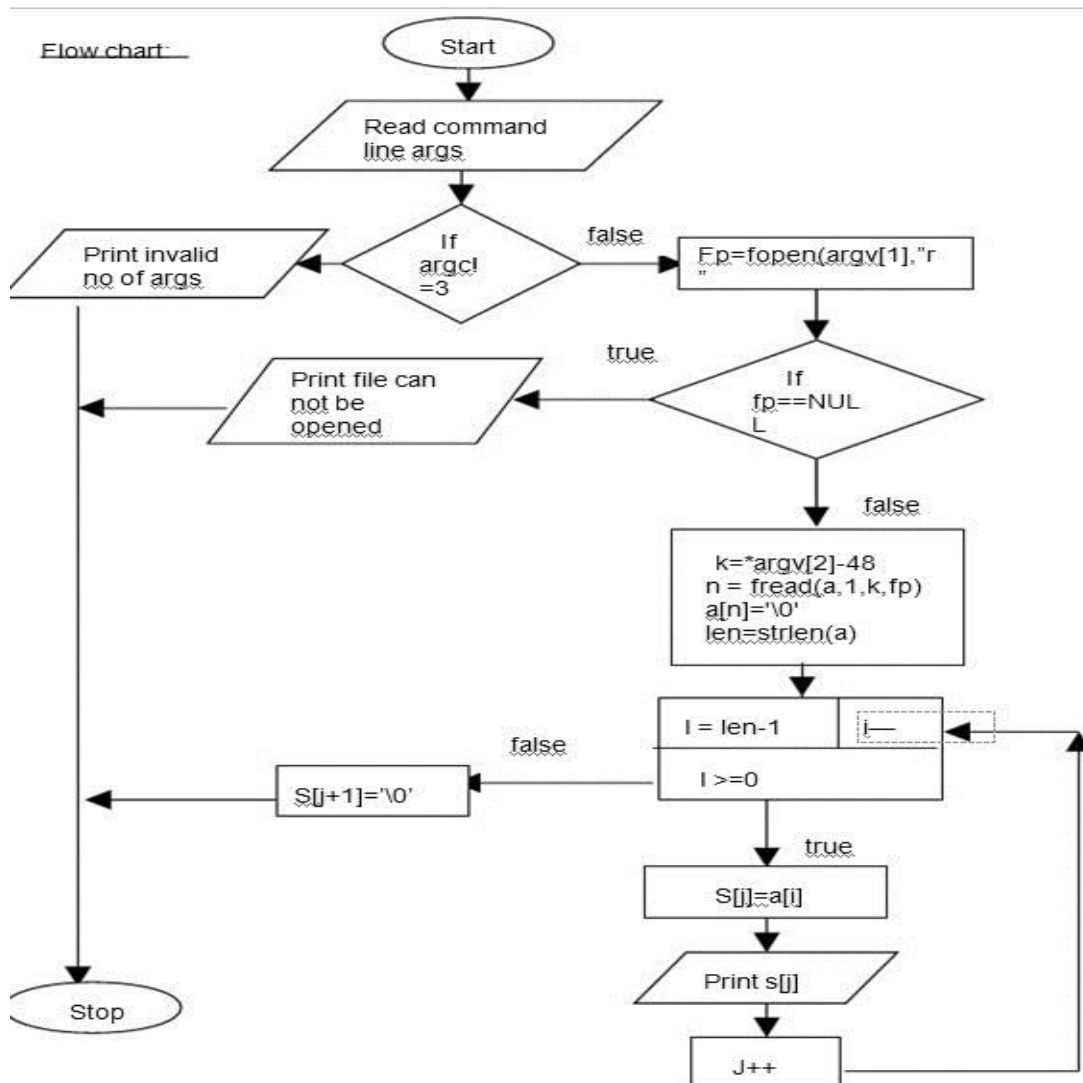
Signature of faculty with date

15) c) Write a C program to reverse the first n characters in a file.
(Note: The file name and n are specified on the command line.)

Aim: To reverse the first n characters in a file

Algorithm:

- Step 1: Start
- Step 2: read the command line arguments
- Step 3: check if arguments=3 or not
If not print invalid no of arguments
- Step 4: open source file in read mode
- Step 5: if NULL pointer, then print file can not be open
- Step 6: Store no of chars to reverse in k
K= *argv[2]-48
- Step 7: read the item from file stream using fread
- Step 8: Store chars from last position to initial position in another string(temp)
- Step 9: print the temp string
- Step 10: Stop



Program:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <process.h>
void main(int argc, char *argv[])
{
    char a[15];
    char s[20];
    char n;
    int k;
    int j=0;
    int i;
    int len;
    FILE *fp;
    if(argc!=3)
    {
        puts("Improper number of arguments.");
        exit(0);
    }
    fp = fopen(argv[1], "r");
    if(fp == NULL)
    {
        puts("File cannot be opened.");
        exit(0);
    }
    k=*argv[2]-48;
    n = fread(a,1,k,fp);
    a[n]='\0';
    len=strlen(a);
    for(i=len-1;i>=0;i--)
    {
        s[j]=a[i];
        printf("%c",s[j]);
        j=j+1;
    }
    s[j+1]='\0';
    getch();
}
```

Input:

source.c

 this is source

ouput.c

 this is source

Output: Command line arguments

source.c ouput.c

source.c

 this is source

ecruos si siht

Record at least 3 results

Signature of faculty with date