MARRI EDUCATIONAL SOCIETY’S GROUP OF INSTITUTIONS

MARRI LAXMAN REDDY

INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Approved by AICTE, New Delhi & Affiliated JNTU, Hyderabad)

Dundigal, Quthbullapur, Hyderabad – 500 043, R.R.Dist., A.P., INDIA

www.mlritm.ac.in, 08418-25554/55

Computer Programming Lab Manual

I B.TECH II-SEMISTER

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

WEEK-1

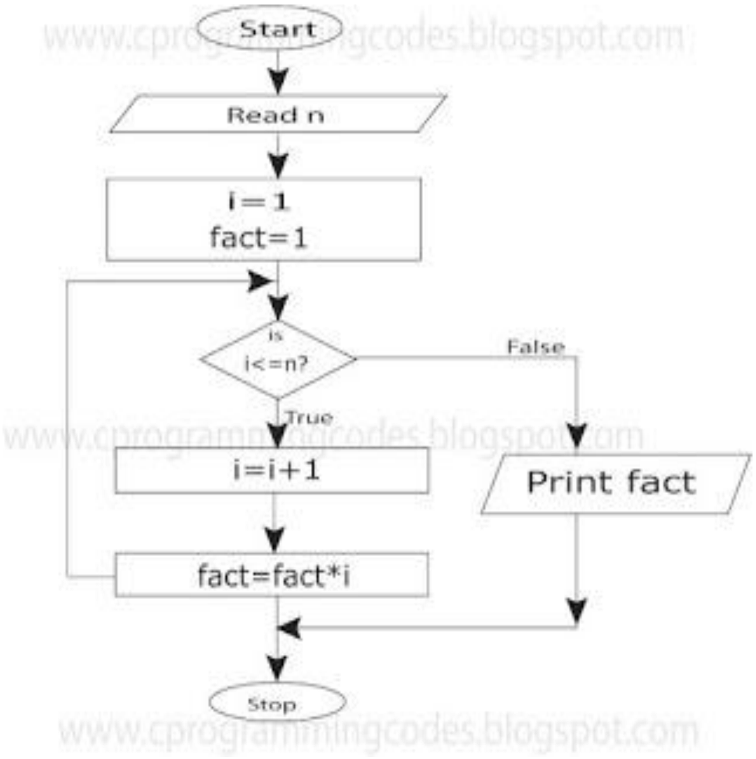
Week-1(a)

AIM: Write a C program to find the factorial of a positive integer

Algorithm:

```
step 1. Start
step 2. Read the number n
step 3. [Initialize]
        i=1, fact=1
step 4. If i<=n true goto step 5 else goto step 8
step 5. fact=fact*i
step 6. i=i+1
step 7: go to step 4
step 8. Print fact
step 9. Stop
```

Flowchart :



www.cprogrammingcodes.blogspot.com

Program:

```
#include<stdio.h>
void main()
{
 int n,i,fact=1;
 printf("Enter any number : ");
 scanf("%d", &n);
 for(i=1; i<=n; i++)
    fact = fact * i;
 printf("Factorial value of %d = %d",n,fact);

}
```

OutPut:

```
Enter any number :
5
Factorial value of 5=120
```

Week-1(b)

AIM: To find the roots of a quadratic equation

Description:

In mathematics, a quadratic equation is a polynomial equation of the second degree. The general form is

$$ax^2 + bx + c = 0,$$

where *x* represents a variable or an unknown, and *a*, *b*, and *c* are constants with $a \neq 0$.

A quadratic equation with real or complex coefficients has two solutions, called roots. These two solutions may or may not be distinct, and they may or may not be real. The roots are given by the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

where b^2-4ac is called discriminant.

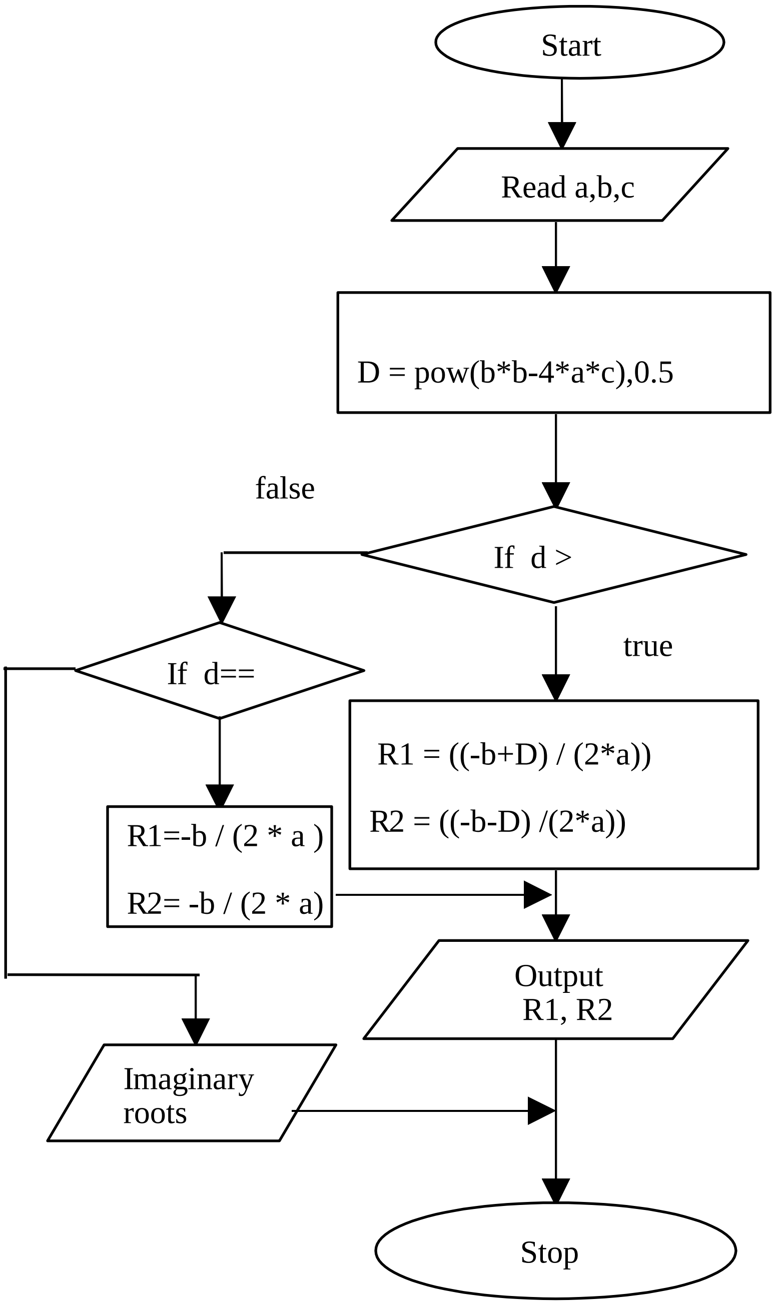
There are 3 cases:

1. b^2-4ac =0 then roots are real and equal
2. b^2-4ac >0 then roots are real and unequal
3. b^2-4ac <0 then roots are imaginary

Algorithm:

```
Step 1: read values a, b and c
Step 2: if  $b^2$-4ac >0
Step 3: two distinct real roots
         root1 = -b + sqrt (b * b – 4 * a * c) / 2 * a;
         root2 = -b – sqrt (b * b – 4 * a * c) / 2 * a;
Step 4: else if $b^2$-4ac =0
        two equal real roots
        root1= root2 = -b/2a
Step 5: else
        roots are imaginary
```

Flowchart


Start
Read a,b,c
D = pow(b*b-4*a*c),0.5
false
If d >
true
If d==
R1 = ((-b+D) / (2*a))
R2 = ((-b-D) /(2*a))
R1=-b / (2 * a)
R2= -b / (2 * a)
Output
R1, R2
Imaginary
roots
Stop

Program:

```
/* Write a C program to find the roots of a quadratic equation */
#include<stdio.h>
#include<math.h>
void main()
{
        float a,b,c,root1,root2,dis;
        clrscr();
        printf("\n Enter values of a,b,c for finding roots of a quadratic eq:\n");
        scanf("%f%f%f",&a,&b,&c);
        dis=b*b-4*a*c;
                if(dis>0)
                {
                        root1=-b+sqrt(dis)/2*a;
                        root2=-b-sqrt(dis)/2*a;
                        printf("\n*****ROOTS ARE REAL &
                UNEQUAL*****\n");
                        printf("\n root1=%f\n root2=%f",root1,root2);
                }
                else if (dis==0)
                {
                        root1=root2=-b/2*a;
                        printf("\n*****ROOTS ARE REAL & EQUAL*****\n");
                         printf("\n root1=%f\n root2=%f",root1,root2);
                }
                else
                printf("\n Imaginary Roots.");


 }
```

Result:

Input	Output
1. 1,-5,4	6.500000,3.500000
2. 4,-2,6	roots are imaginary

WEEK-2

Week-2(a)

AIM: Write a C program to determine if the given number is a prime number or not.

Description:

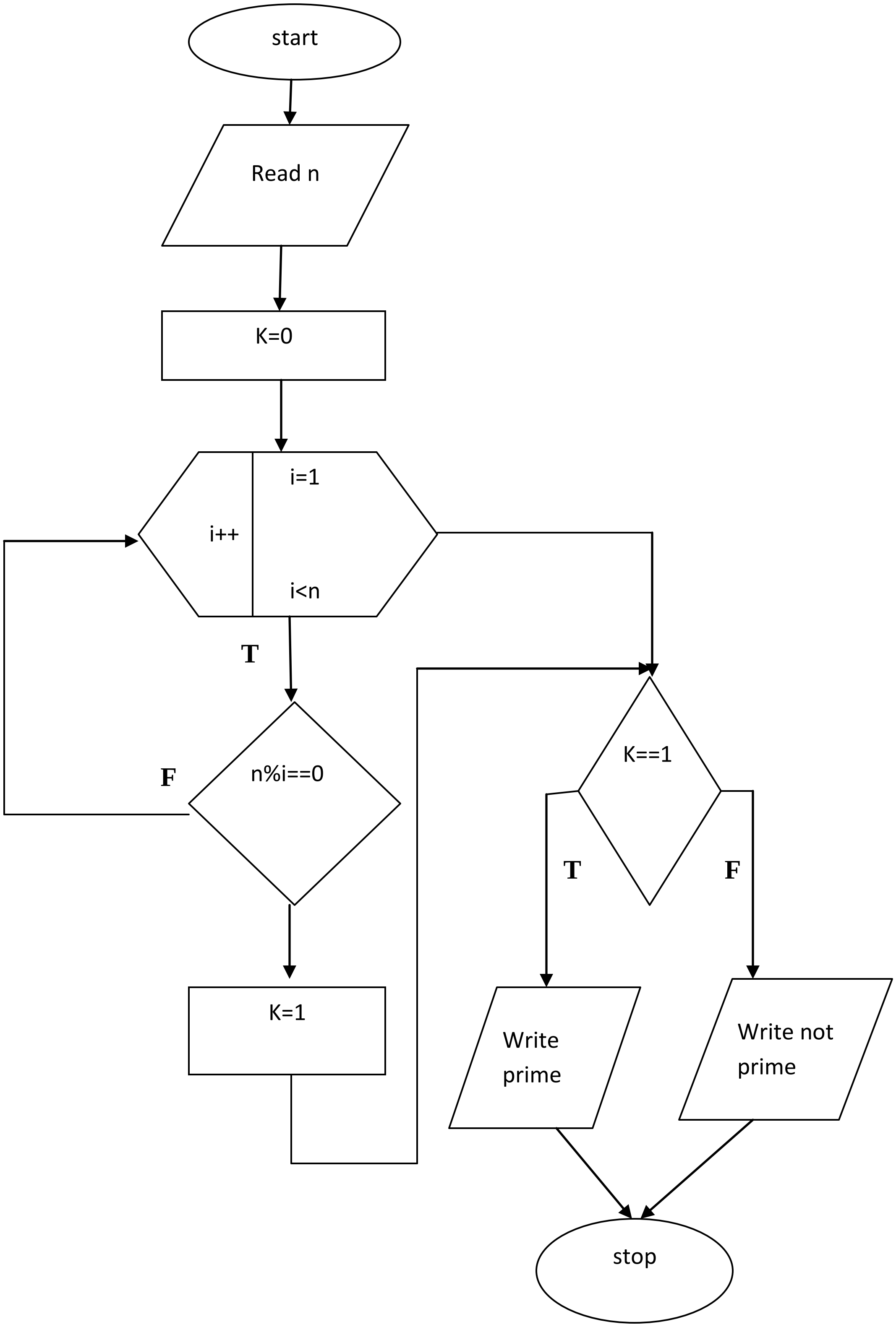
A natural number greater than one has not any other divisors except 1 and itself. In other word we can say which has only two divisors 1 and number itself. For example: 5

Their divisors are 1 and 5.
Note: 2 is only even prime number.

Algorithm:

```
Step1: Start.
Step2: Read n
Step3: K=1
Step4: for(i=2;i<=n/2;i++)
Step5: begin
Step6: if(n%i==0) goto 7 else goto 4
Step7: k=0
Step8: goto 10
Step9: end for
Step10: if (k==1) goto 11 else goto 12
Step11: WRIE "The given number is prime" goto 13
Step12: WRIE "The given number is not a prime"
Step13: stop
```

Flowchart:


start
Read n
K=0
i=1
i++
i<n
T
F
n%i==0
K=1
K==1
T
F
Write prime
Write not prime
stop

Program:

```
#include<stdio.h>
void main()
{
        int i,j,n,k=1;
        printf("Enter the value of n:");
        scanf("%d",&n);
        for(i=2;i<=n/2;i++)
        {
                if(n%i==0)
                {
                        k=0;
                         break;
                }
        }
if(k==1)
 printf("The given number is prime");
else
printf("The given number is not a prime");
}
```

Output:

```
Enter the value of n:5
The given number is prime
```

Week-2(b)

AIM: To print the Fibonacci series for 1 to n value

Description:

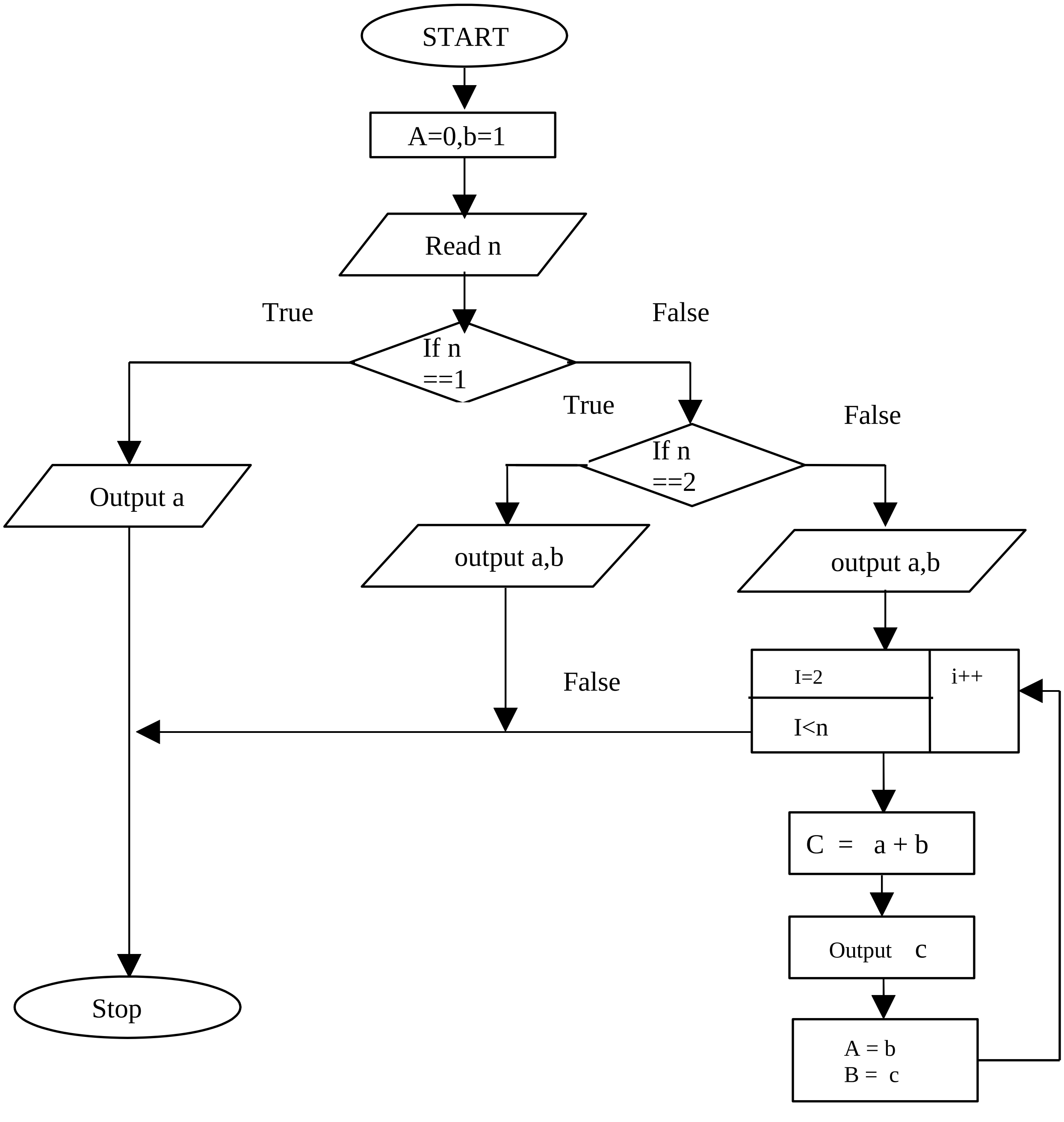
By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

e.g. 0, 1, 1, 2, 3, 5, 8, 13, 21, …

Algorithm:

```
Step  1 :declare variables called seed0,seed1,number,count
Step  2: set seed0 to 0
Step 3:set seed1to1
Step 4:read a number from user
Step 5:if the number is less than 1then
Step 6:print the Fibonacci series is not possible
            else
Step 7: if the number is equal to 1 then print seed0
            else
Step 8:if the number is equal to 2 then print seed0,seed1
            else
Step 9:print seed0
Step 10  :  print seed1
Step 11: set  count to 1
Step 12:loop (count <= number-2)
Step 13:fib = seed0 + seed1
Step 14:print fib
Step 15:seed0 = seed1
Step 16: seed1 = fib
 Step17: end loop
```

Flowchart:


START
A=0,b=1
Read n
True
False
If n
==1
True
False
If n
==2
Output a
output a,b
output a,b
False
I=2
i++
I<n
C = a + b
Output c
A = b
B = c
Stop

Program:

```
/*To print the Fibonacci series for 1 to n value*/
#include <stdio.h>
#include <conio.h>
void main()
{
    int num1=0, num2=1,n,counter,fab;
    clrscr();
    printf("\n\n\n\t\tENTER LENGTH OF SERIES FOR 1 TO N : ");
    scanf("%d",&n);
    if(n<1)
    printf("\n\n\n\t\ THE FIBONACCI SERIES IS NOT POSSIBLE");
    else
        if(n==1)
        {
            printf("\n\n\t\t\t<------------------FIBONACCI SERIES------------");
            printf("\n %d",num1);
        }
        else
            if(n==2)
            {
                printf("\n\n\t<-----------FIBONACCI SERIES--------->");
                printf("\n\n\t\t%d  %d", num1,num2);
            }
            else
            {
                printf("\n\n\t\t\t<----------FIBONACCI SERIES------->");
                printf("\n\n\t\t%d  %d",num1,num2);
                for(counter = 1; counter <= n-2; counter++)
                {
                    fab=num1 + num2;
                    printf("%3d",fab);
                    num1=num2;
                    num2=fab;
                }
            }
     }
```

Result:

Input	Output
1	0
2	0 1
5	0 1 1 2 3
7	0 1 1 2 3 5 8

WEEK-3

Week-3(a)

AIM:

To generate and display a pyramid of numbers.

Description :

In this program we will generate and display a pyramid of numbers

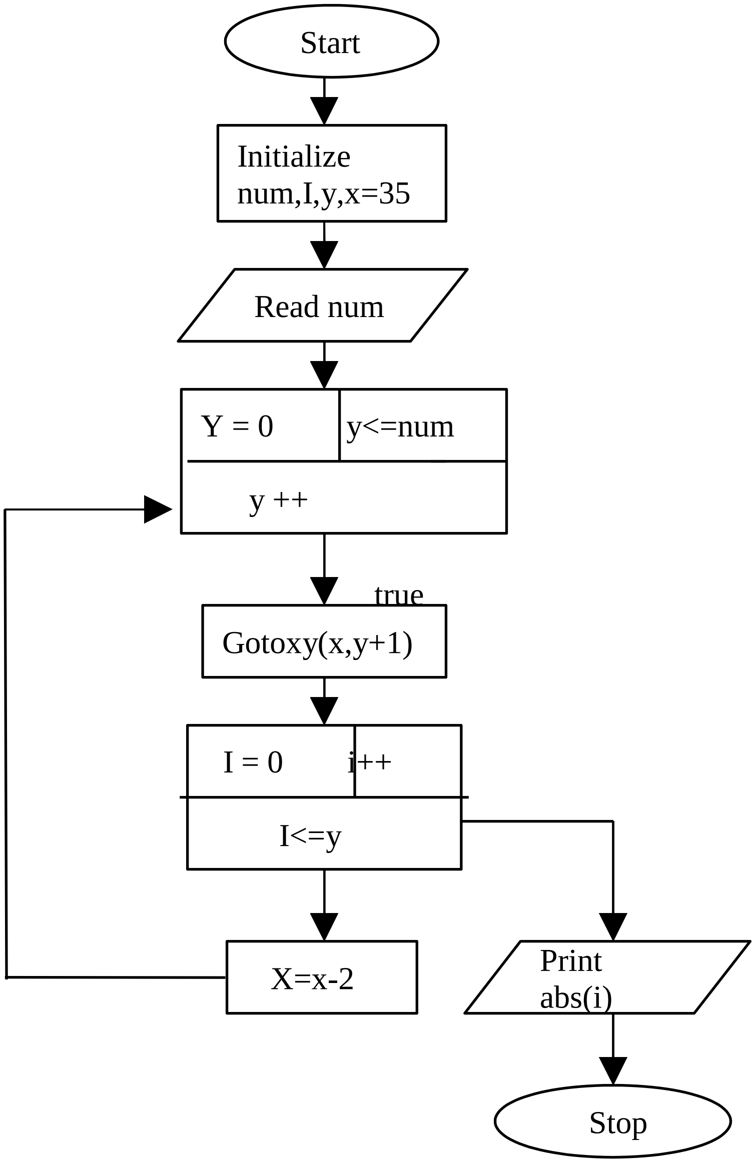
e.g**.: pyramid** of numbers having 3 rows

```
  0
 101
21012
```

Algorithm :

```
start
    declare pyramid ( int ),n
    read  n
    call pyramid(n)
    in pyramid(n) definition, declare i ,j
    set i to 0
    loop  i<n
        set  j to n-1
        loop  j>i
            print spaces
            decrement j by1
        end loop
        set j to i
        loop  j>=0
            print  j value
            decrement j by 1
         end loop
         set jto 1
          loop  j<i
            print j value
            increment j by 1
         print new line
         increment i by 1
    end loop

stop
```

Flowchart:


Start
Initialize
num,I,y,x=35
Read num
Y = 0
y<=num
y ++
true
Gotoxy(x,y+1)
I = 0
i++
I<=y
X=x-2
Print
abs(i)
Stop

Program:

/* To generate and display a pyramid of numbers*/

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

void pyramid ( int n )
{
        int i,j;
        for ( i=0; i<n; i++ )
        {
                for ( j=n-1; j>i; j-- )
                        printf("  ");
                for ( j=i; j>=0; j-- )
                        printf(" %d",j);
                for ( j=1; j<=i; j++ )
                        printf(" %d",j)

        printf("\n");
        }
    }
```

Result:

```
    Input                                  Output

      Ente n value 3                         0

                                            101

                                           21012



      Enter value 4                          0

                                            101

                                           21012

                                          3210123
```

Viva Questions:

1) What is a pascal triangle?
2) Define array? Types of arrays?
3) What is function signature? And explain it?
4) What is pyramid triangle?
5) What is the out put of following?

```
For(i=0;i<100;i++);

Printf(“%d”,i);
```

Week-3(b)

AIM: To calculate the following sum:

$$sum=1 - x^{2} / 2! + x^{4} / 4! - x^{6} / 6! + x^{8} / 8! - x^{10} / 10!$$

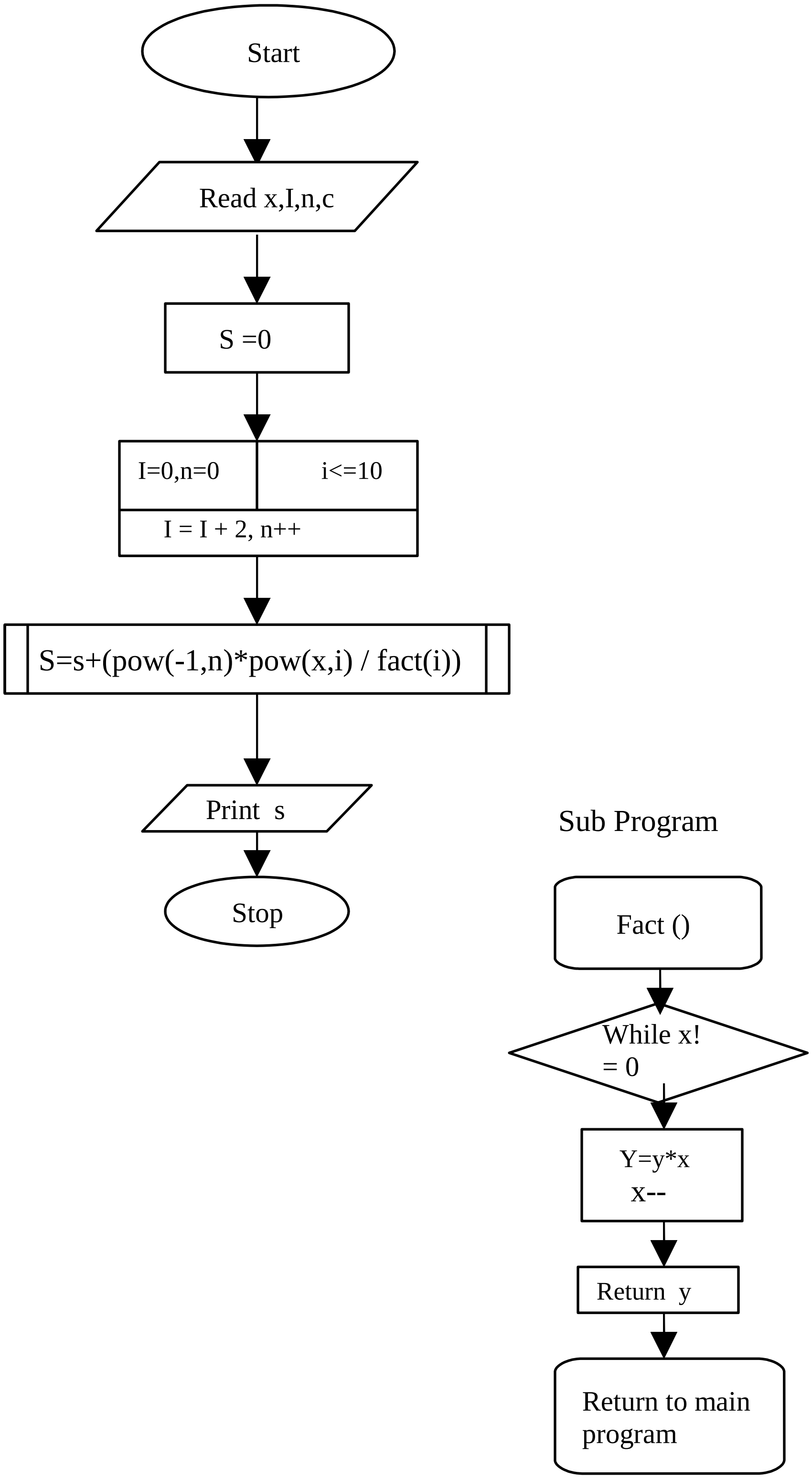
Description:

Calculate the given expression for different values of x.
$sum=1 - x^{2} / 2! + x^{4} / 4! - x^{6} / 6! + x^{8} / 8! - x^{10} / 10!$

Algorithm:

```
Step 1:   start
Step 2:    declare variables i, j, x, y, v,f,sum
Step 3:   set sum to 1
Step 4:    set f to 1
Step 5:    read x
Step 6:    set i to 1
Step 7:    loop( i <= 5)
Step 8:   y = pow (-1, i) * pow (x, 2 * i)
Step 9:    f = 1
Step 10:  set j to 1
Step 11:   f = f * j;
Step 12:   v = y / f;
Step 13:  end loop
Step 14:   sum=sum+v;
Step 15:   end loop
Step 16:  print sum
Step 17:  stop
```

flowchart:


Start
Read x,I,n,c
S =0
I=0,n=0
i<=10
I = I + 2, n++
S=s+(pow(-1,n)*pow(x,i) / fact(i))
Print s
Stop
Sub Program
Fact ()
While x!
= 0
Y=y*x
x--
Return y
Return to main
program

Program:

/* Write a C program to calculate the following sum
sum=1 - x^2 / 2! + x^4 / 4! - x^6 / 6! + x^8 / 8! - x^{10} / 10! */

```
#include<stdio.h>
#include<math.h>
void main()
{
        int i,j;
        float x,y,v,f=1,sum=1;
        clrscr();
        printf("Enter X value\n");
        scanf("%f",&x);
          for(i=1;i<=5;i++)
         {
                y=pow(-1,i)*pow(x,2*i);
                for(j=1;j<=2*i;j++)
                {
                        f=f*j;
                        v=y/f;
                }
                sum=sum+v;
         }
        printf(" The sum of series is %f",sum);
}
```

Result:

Input	Output
a) 0	1.000000
b) 1	0.540302
c) 5	1.069001

Viva Questions:

1.What is data type?
2.What is a variable?
3.Explain pow(_ , _) ?
4.Explain Nested loop?
5.What are Control Statements in C
6.What is an expression?
7.What is an operator?
8.What is an operand?
9.What is precedence?
10.What is Associativity?

WEEK-4

Week-4(a):

AIM :

The least common multiple (lcm) of two positive integers a and b is the smallest integer that is evenly divisible by both a and b. Write a C program that reads two integers and calls lcm (a, b) function that takes two integer arguments and returns their lcm. The lcm (a, b) function should calculate the least common multiple by calling the gcd (a, b) function and using the following relation:

LCM (a,b) = ab / gcd (a,b)

Algorithm:

Main program

Step 1: Start

Step 2: Declare integer variables a, b

Step 3: Read values of a, b

Step 4: Call the function lcm (a,b)

Step 5: Stop

Algoritm for lcr function

Step 1: Start

Step 2: return (a*b)/gcd(a, b)

Step 3: Stop

Algoritm for gcd function

Step 1: Start

Step 2: if a==0 true go to Step 3 else go to step 5

Step 3: return b

Step 4: go to step 12

Step 5: if b!=0 true go to step 6 else go to step 11

Step 6: if (a > b) true go to step 7 else go to step 9

Step 7: a = a - b;

Step 8: go to step 5

Step 9: b=b-a

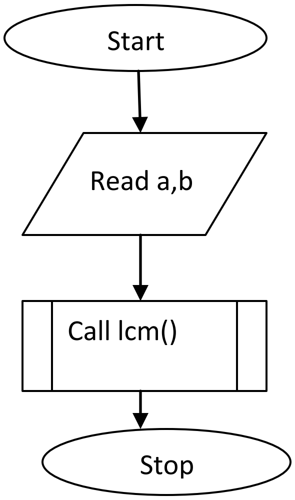
Step 10: go to step 5

Step 11: return a

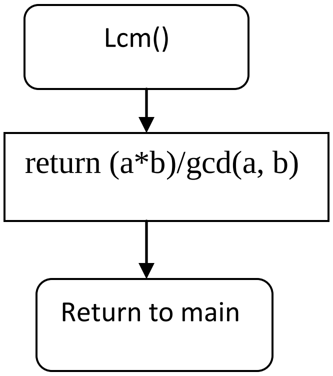
Step 12: Stop

Flow Chart:

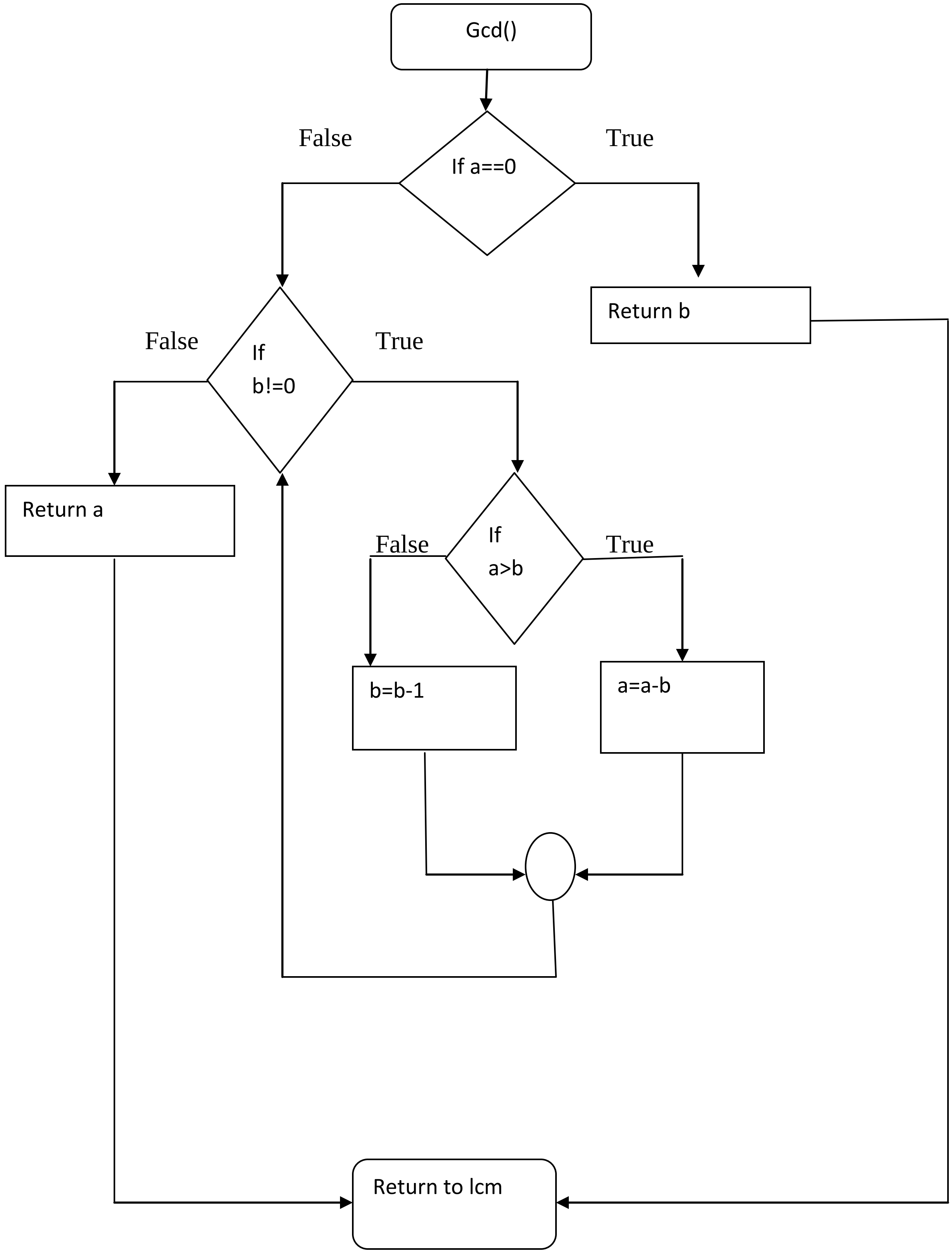
Main function



Lcm function



Gcd function



Program:

```
#include <stdio.h>
int gcd(int , int );
 int lcm(int , int) ;

// Recursive function to return gcd of a and b

int gcd(int a, int b)
{
  if (a == 0)
{
                return b;
  }

  while (b!= 0)
 {
                if (a > b)
                {
                                a = a - b;
                }
                else
                {
                                 b = b - a;
                }
  }

  return a;
}

// Function to return LCM of two numbers
int lcm(int a, int b)
{
    return (a*b)/gcd(a, b);
}

// Driver program to test above function
int main()
{
    int a,b;
            printf(“Enter any two positive integers\n”);
            scanf(“%d%d”,&a,&b);
    printf("LCM of %d and %d is %d ", a, b, lcm(a, b));
    return 0;
}
```

OutPut:

```
Enter any two positive integers

15

20

LCM of 15 and 20 is 60
```

Week-4(b):

AIM: Write a C program that reads two integers n and r to compute the ncr value using the following relation:

ncr (n,r) = n! / r! (n-r)! . Use a function for computing the factorial value of an integer.

Algorithm:

Main function

Step 1: Start

Step 2: Declare integer variables n,r,ncr

Step 3: Read values of n,r

Step 4: Call the function find_ncr()

Step 5: print ncr value

Step 6:Stop

Find_ncr

Step 1: Start

Step 2: Declare integer variable result

Step 3: result = factorial(n)/(factorial(r)*factorial(n-r));

Step 4: return result

Step 5: Stop

Factorial

Step 1: Start

Step 2: Declare integer variables result=1,c

Step 3: c=1

Step 4: if c<=n true goto step 5 else goto step 8

Step 5: result = result*c;

Step 6: c++

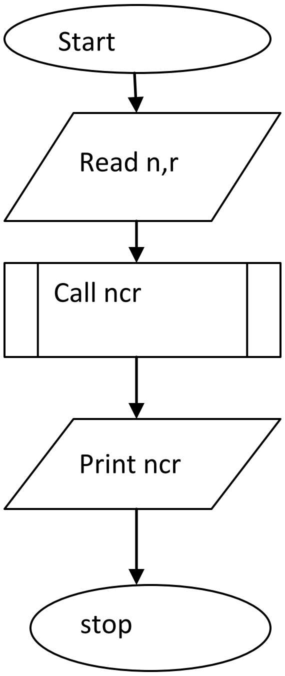
Step 7: goto step 4

Step 8: return result

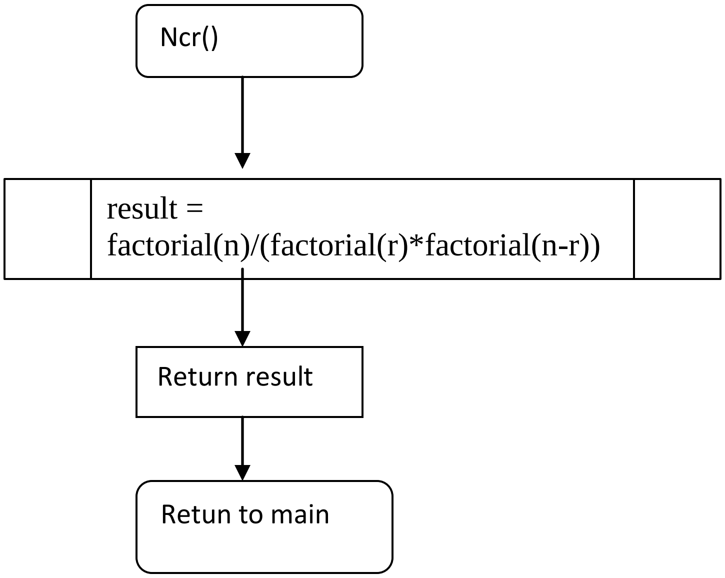
Step 9: Stop

Flow Chart:

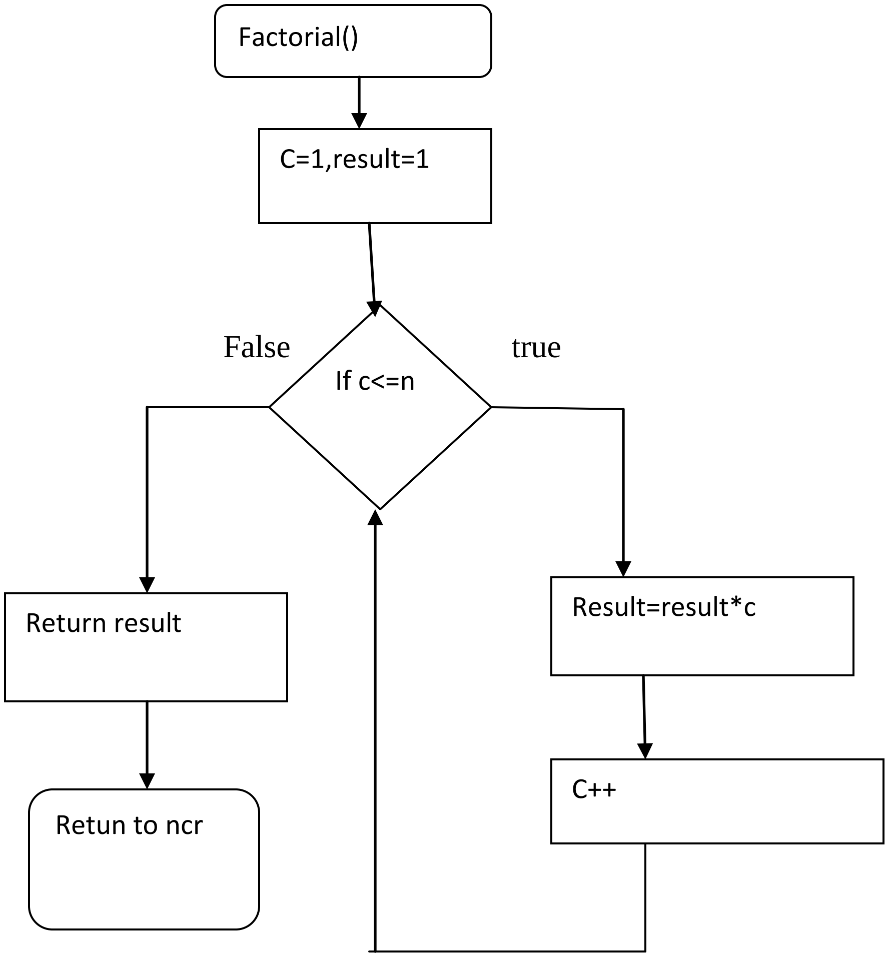
Main function



Ncr function



Factorial


Factorial()
C=1,result=1
False
true
If c<=n
Return result
Result=result*c
C++
Retun to ncr

Program:

```
#include<stdio.h>

void main()
{
   int n, r;
   int ncr;

   printf("Enter the value of n and r\n");
   scanf("%d%d",&n,&r);

   ncr = find_ncr(n, r);

   printf("%dC%d = %d\n", n, r, ncr);


}

int find_ncr(int n, int r)
 {
   int result;

   result = factorial(n)/(factorial(r)*factorial(n-r));

   return result;
}


int factorial(int n)
 {
   int c;
   int result = 1;

   for (c = 1; c <= n; c++)
      result = result*c;

   return result;
}
```

Result:

```
Enter the value of n and r
5
2
5C2=10
```

WEEK-5

Week-5(a)

Write C program that reads two integers x and n and calls a recursive function to compute xn

```
#include <stdio.h>
int     power(int);
void main()
{
            int       x;

            int       n;

            printf("\n enter values of x and n");
            scanf("%d%d",&x,&n);

            printf("\n %d to the power of % d is %d",x,n,power(x,n));
}


int  power(int  x, int n)
{

            if (x==0)
            {
                        return 0;

            }
            else
            if(n==0)
            {
                        return 1;

            }
            else
            if (n>0)
            {
                        return( x* power(x,n-1));
            }
}
```

Output:

```
Enter values of x and n 6 4
6 to the power of 4 is 1296
```

Week-5(b)

Write a C program that uses a recursive function to solve the Towers of Hanoi problem.

```
#include<stdio.h>
void towers(int n,char from,char to,char aux);
void main()
{
	int n;
	printf("Program for towers of Hanoi problem\n");
	printf("Enter the total number of disks\n");
	scanf("%d",&n);
	towers(n,'A','C','B');
}

void towers(int n,char from,char to,char aux)
{
	if(n==1)
	{
		printf("Move disk 1 from %c peg to %c peg\n",from,to);
		return;
	}
	towers(n-1,from,aux,to);
	printf("Moves disk %d from %c peg to %c peg\n",n,from,to);
	towers(n-1,aux,to,from);
}
```

Output:

```
Enter the total number of disks
3
Move disk 1 from A peg to C peg
Moves disk 2 from A peg to B peg
Move disk 1 from C peg to B peg
Moves disk 3 from A peg to C peg
Move disk 1 from B peg to A peg
Moves disk 2 from B peg to C peg
Move disk 1 from A peg to C peg
```

Week-5(c)

Write a C program that reads two integers and calls a recursive function to compute ncr value.

```
#include<stdio.h>
int ncr(int n,int r);
 void  main()
 {
            int n,r,y;
            printf("enter the value of n and r resp.\n");
            scanf("%d %d",&n,&r);
             y=ncr(n,r);
             printf("the value of ncr is %d\n",y);
}

 int ncr(int n,int r)
 {
             int y,res;
             if(r==0)
            {
                         y=1;
                         return y;
             }
             else
             if(n==r)
             {
                        y=1;
                        return y;
            }
            else
            {
                        res=ncr(n-1,r-1)+ncr(n-1,r);
            }
            return res;

}
```

Output:

```
enter the value of n and r resp 6 5
the values of ncr is 6
```

WEEK-6

Week-6(a)

Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user using Sieve of Eratosthenes algorithm.

```
#include<stdio.h>
void main ()
{
        int  n, count, j, i=1;
        printf(“enter the value of n”);
        scanf(“%d”,&n);
        while(i<=n)
        {
                count=0;
                for(j=1;j<=i;j++)
                {
                        if(i%j==0)
                        {
                                count++;
                        }
                }
                if(count==2)
                {
                        printf(“%d”, i);
                }
                i++;
        }
}
```

Output:

```
enter the value of n
6
2 3 5
```

Week-6(b)

AIM :

To Write a C program that uses non recursive function to search for a Key value in a given list of integers using Linear search.

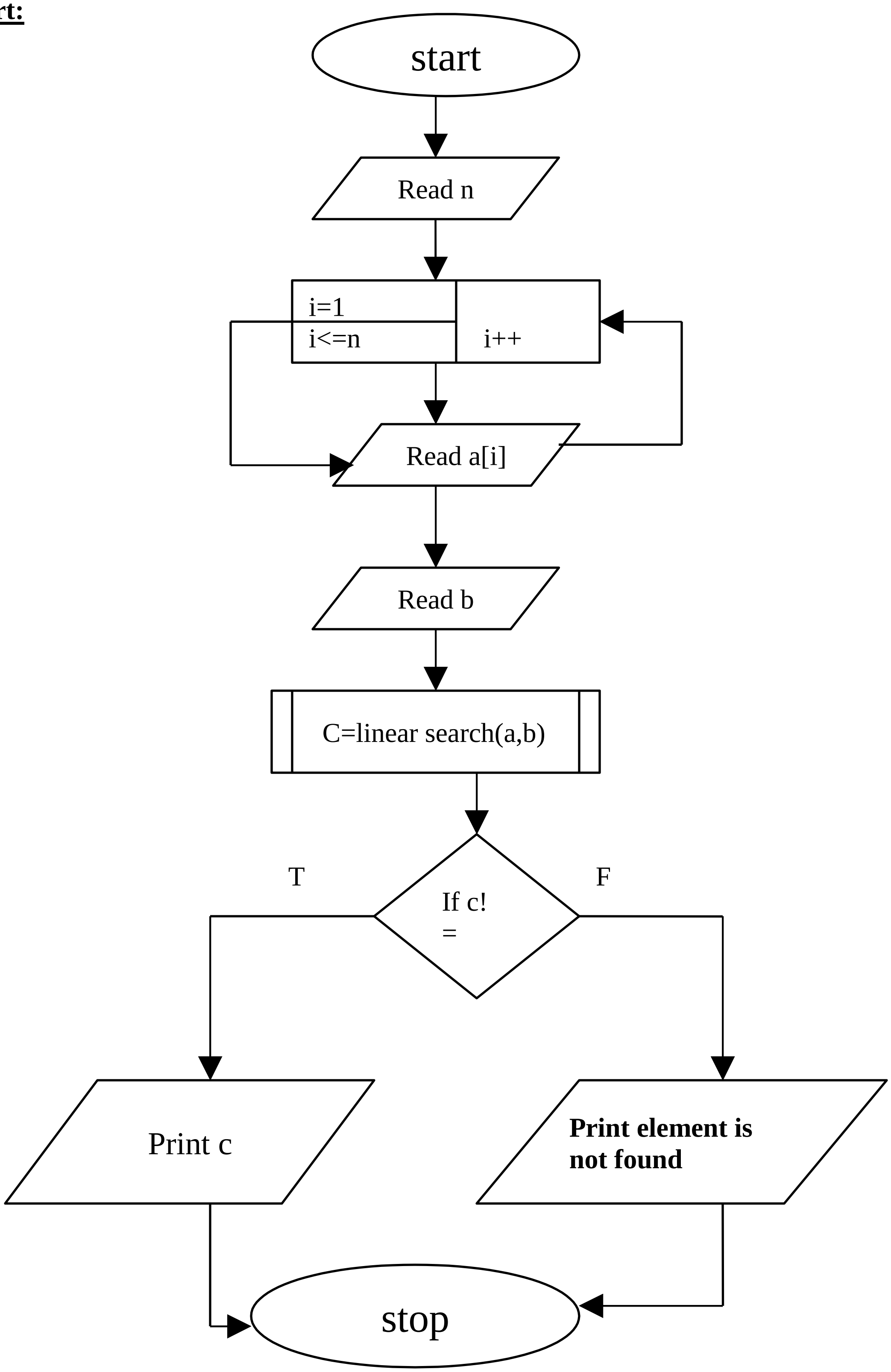
Description:

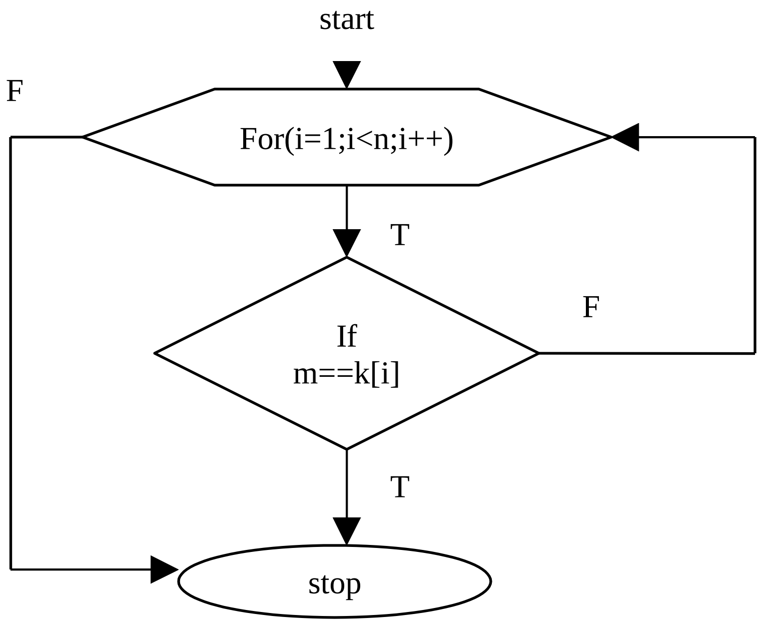
The linear search is most simple searching method. The key which is to be searched is compared with each element of the list one by one. If a match exists, the search is terminated and it returns the position of the element. If the end of list is reached it means that the search has failed and key is not in the list.

Algorithm:

```
start
    declare an array, variables
    read n value
    read elements into an array and key element
    loop
        if key is matched with element of an array
            set found flag
            break
        if (found)
            print value and position
        else
            print “not found” message
    end loop
stop
```

flowchart:


start
Read n
i=1
i<=n
i++
Read a[i]
Read b
C=linear search(a,b)
T
F
If c!
=
Print c
Print element is
not found
stop


start
F
For(i=1;i<n;i++)
T
F
If
m==k[i]
T
stop

Program:

```
/* C Program to implement linear search. */

void main()
{
        int list[60];
        int looker, found, i,n, target;
        clrscr();
        found=0;
        printf("enter size iz");
        scanf("%d",&n);
        printf("enter key  element");
        scanf("%d",&target);
        printf("enter elements....");
        for (i=0; i <= n; i++)
        {
                scanf("%d",&list[i]);
        }
        for (i=0; i <= 6; i++)
        {
                if (target == list[i])
                {
                        found  = 1;
                        looker =i;
                        break;
                }
        }
        if (found)
            printf ("Target %d is found at location %d\n", target, looker);
        else
            printf ("Target %d not found \n", target);
        getch();
    }
```

Result:

```
     Input                              Output
    enter elements size: 5             28 is found at position 3
    7 10 28 -5 100 67
    key element is:  28

    enter elements size: 5             158 is not found
    7 10 28 -5 100 67
    target element is:  158
```

WEEK-7

Week-7(a)

Write a menu-driven C program that allows a user to enter n numbers and then choose between finding the smallest, largest, sum, or average.
The menu and all the choices are to be functions. Use a switch statement to determine what action to take.
Display an error message if an invalid choice is entered.

```
#include<stdio.h>
void main()
{
        int array[50],c,small,large,sum=0,n,i;
        float avg;
        printf("Enter the size of the elements\n");
        scanf("%d",&n);
        printf("Enter the elements into the array\n");
        for(i=0;i<n;i++)
        scanf("%d",&array[i]);
        printf("1: Smallest\n2:Largest\n3:Sum\n4:Average\n");
        printf("Enter your choice\n");
        scanf("%d",&c);
        switch(c)
        {
        case 1: small=array[0];
                for(i=0;i<n;i++)
                {
                        if(small>array[i])
                        small=array[i];
                }
                printf("Smallest element in the list=%d",small);
                break;
        case 2:large=array[0];
                for(i=0;i<n;i++)
                {
                        if(large<array[i])
                        large=array[i];
                }
                printf("Largest element in the list=%d",large);
                break;
        case 3:
                for(i=0;i<n;i++)
                {
                        sum=sum+array[i];
                }
                printf("sum of  elements in the list=%d",sum);
                break;
        case 4:for(i=0;i<n;i++)
                {
                        sum=sum+array[i];
                }
                avg=sum/n;
                printf("avg of  elements in the list=%f",avg);
                break;
        default:
                printf("Invalid choice\n");
        }
}
```

Output:

```
Enter the size of the elements  2

Enter the elements into the array
4    6
1: Smallest
2:Largest
3:Sum
4:Average
Enter your choice 1
Smallest element in the list=4
```

Week-7(b)

AIM :

To Write a C program that uses non recursive function to search for a Key value in a given sorted list of integers using Binary search.

Description:

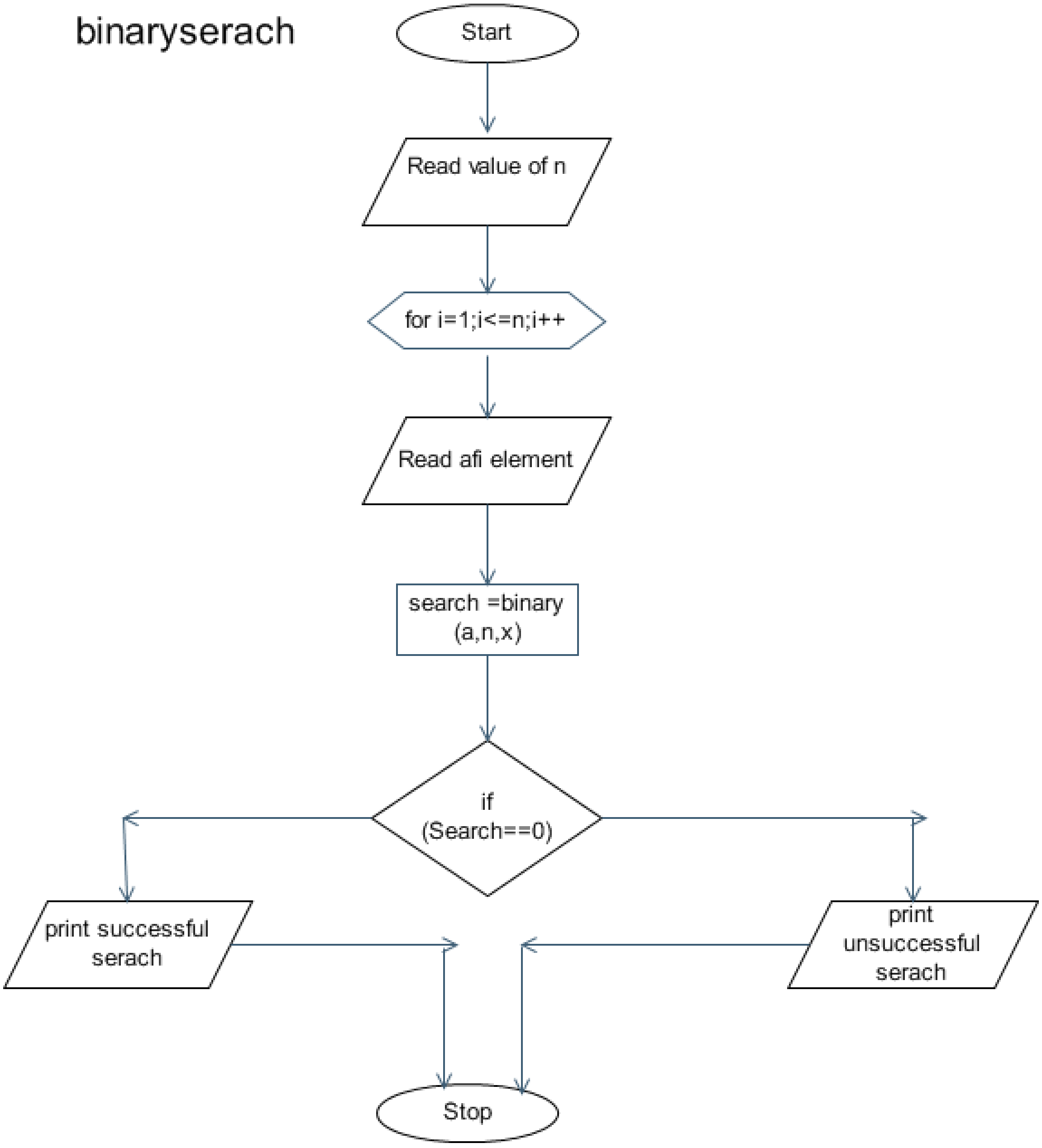
For binary search, the elements in the list must be in sorted order. In this search method the key element compared with middle element of the list. If the key match, then return the position. If the key less than the middle element of the list, repeat above procedure on the sub list to the left of the middle element.

If the key more than the middle element of the list, repeat above procedure on the sub list to the right of the middle element.

Algorithm:

```
start
    declare an array, variables
    read n value
    read elements into an array and key element
    middle = (first + last) / 2
    loop (first<last)
        if (a [middle] <key)
            first = middle + 1
        else if (a [middle] ==key)
            print value and position
         else
            last = middle – 1
            middle = (first + last) / 2
    end loop
    if (first > last)
        print “not found” message
stop
```

flowchart:


binaryserach
Start
Read value of n
for i=1;i<=n;i++
Read afi element
search =binary (a,n,x)
if (Search==0)
print successful serach
print unsuccessful serach
Stop

Program:

```
/* C Program to implement binary search. */
void main()
{
    int array[60];
    int first, middle, last, search,n,i;
    clrscr();
    first = 0;
    printf("ENTER SIZE OF AN ARRAY\n");
    scanf("%d",&n);
    printf("ENTER SEARCH ELEMENT...\n")
    scanf("%d",&search);
    printf(" ENTER ELEMENTS....\n");
    last = n-1;
    for(i=0;i<=n;i++)
    {
        scanf("%d",&array[i]);
    }
    middle = (first+last)/2;
    while( first <= last )
    {
        if ( array[middle] < search )
            first = middle + 1;
        else if ( array[middle] == search )
        {
            printf("%d found at location %d.\n", search, middle+1);
            break;
```

```
         else
            last = middle - 1;
            middle = (first + last)/2;
        }
        if ( first > last )
        printf("Not found! %d is not present in the list.\n", search);
        getch();
    }
}
```

Result:

```
 Input                          Output
enter elements size: 5          28 is found at position 3
7 10 28 -5 100 67
key element is:  28

enter elements size: 5          158 is not found
7 10 28 -5 100 67
target element is:  158
```

WEEK-8

Week-8(a)

AIM :

To Write a C program that implements the Bubble sort method to sort a given list of names in ascending order.

Description:

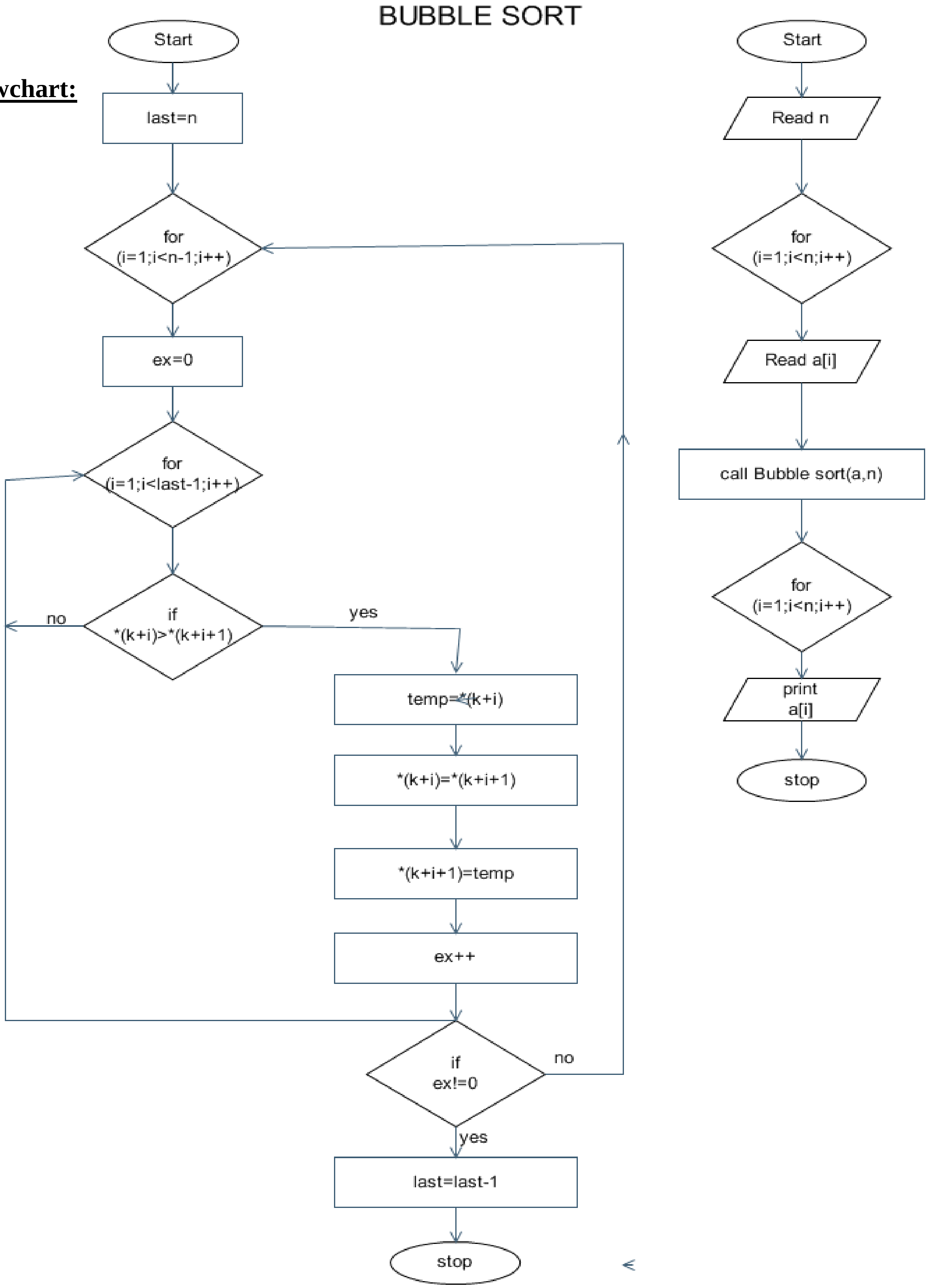
Bubble sort works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. In each pass, this results in the largest element ending up at the end of the array and the smallest element bubbles to the beginning of the array.

```
e.g.:   The list before sorting:  23 78 45    8   32 56

        The list after one pass:    8  23 78 45   32 56

        After all passes          :     8 23 32 45    56 78
```

Algorithm :

```
start
    declare an array, variables
    read n value
    read elements into an array
    loop (i<n)
        loop (n>i)
            if (a[n]>a[n-1])
                swap
    end loop
    print sorted elements
stop
```

Flowchart:


BUBBLE SORT
Start
last=n
for
(i=1;i<n-1;i++)
ex=0
for
(i=1;i<last-1;i++)
if
(k+i)>(k+i+1)
no
yes
temp=*(k+i)
(k+i)=(k+i+1)
*(k+i+1)=temp
ex++
if
ex!=0
no
yes
last=last-1
stop
Start
Read n
for
(i=1;i<n;i++)
Read a[i]
call Bubble sort(a,n)
for
(i=1;i<n;i++)
print
a[i]
stop

Program:

/* C Program to implement Bubble sort.*/

```
#include<stdio.h>

main()

{

int i,j,t,a[5],n;

clrscr();

printf("enter the range of array:");

scanf("%d",&n);

printf("enter elements into array:");

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

scanf("%d",&a[i]);

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++)

 if(a[i]>a[j])

 {

  t=a[i];

  a[i]=a[j];

  a[j]=t;

  }

  printf("the sorted order is:");

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

  printf("\t%d",a[i]);

  getch();

}
```

Result:

```
enter the range of array:3
enter elements into array:3
2
1
the sorted order is:    1       2       3
enter the range of array:5

enter elements into array:56
23
34
12
8
the sorted order is:    8       12      23      34      56
```

Viva Questions:

1. What is an Algorithm?
2. What is Algorithmics?
3. What is Big ‘O’ for bubble and selection sort?
4. display the efficiency of bubble   sort?
5. Define bubble sort ?

Week-8(b)

AIM :

To perform addition and multiplication of two matrices by using functions.

Description:

In mathematics, a matrix is a rectangular array of numbers, symbols, or expressions.

The individual items in a matrix are called its elements or entries.

Matrices of the same size can be added or subtracted element by element.

A (m x n)	B (m x n)	A + B (m x n)	C (m x n)
1 3 1	0 0 5	1+0 3+0 1+5	1 3 6
1 0 0	7 5 0	1+7 0+5 0+0	8 5 0

Two matrices can be multiplied only when the number of columns in the first equals the number of rows in the second.

Multiply elements of each row by elements of each column.

A (m x n)	B (n x p)	A X B	C (m x p)
2 -3	1 0	(2)(1) + (-3)(-2) (2)(0) + (-3)(1)	8 -3
4 5	-2 1	(4)(1) + (5)(-2) (4)(0) + (5)(1)	-6 5
6 0		(6)(1) + (0)(-2) (6)(0) + (0)(1)	6 0

Algorithm:

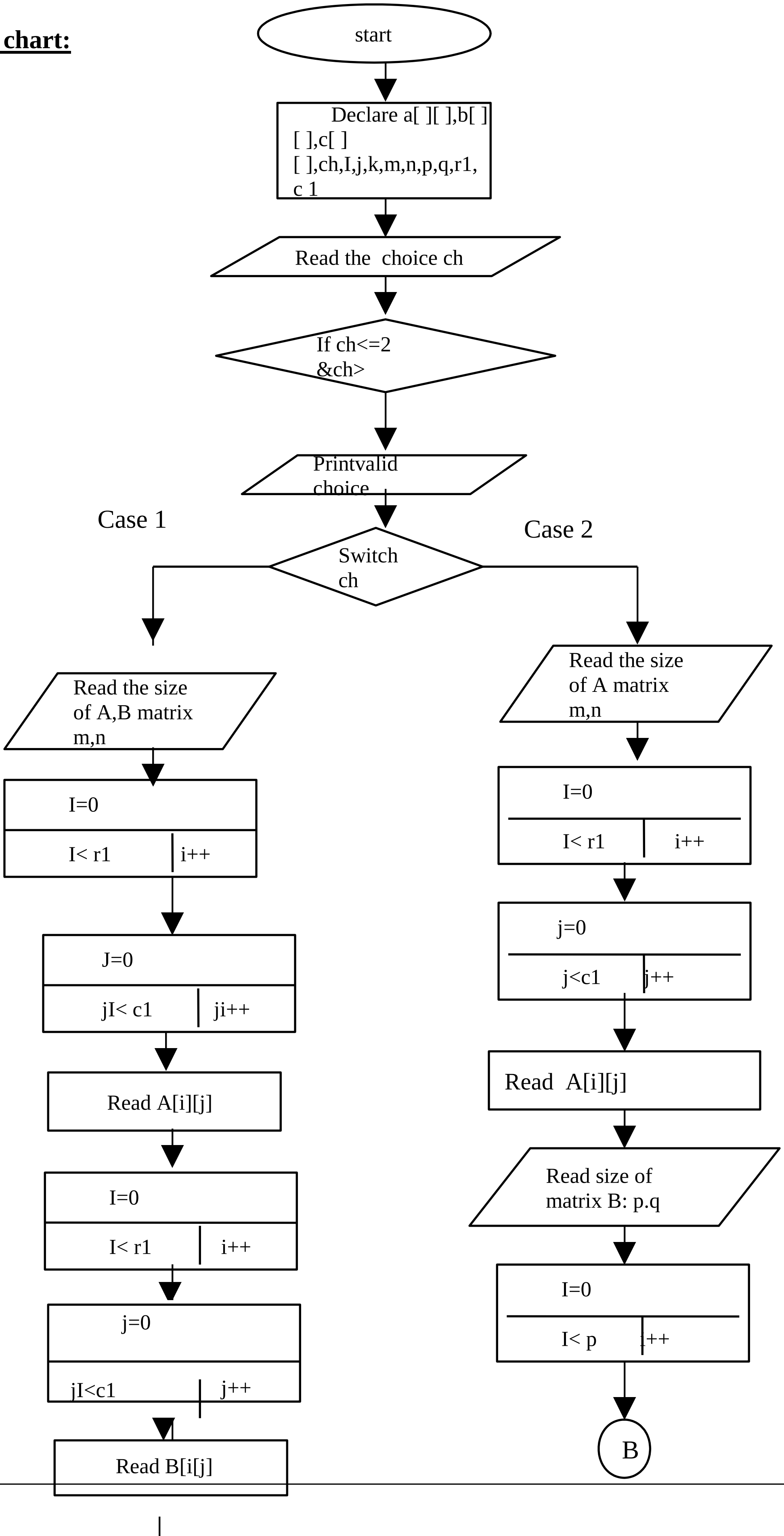
```
start
    declare matrices A, B and C
    declare variables for loop…
    read the size of matrix A
    read matrix A
```

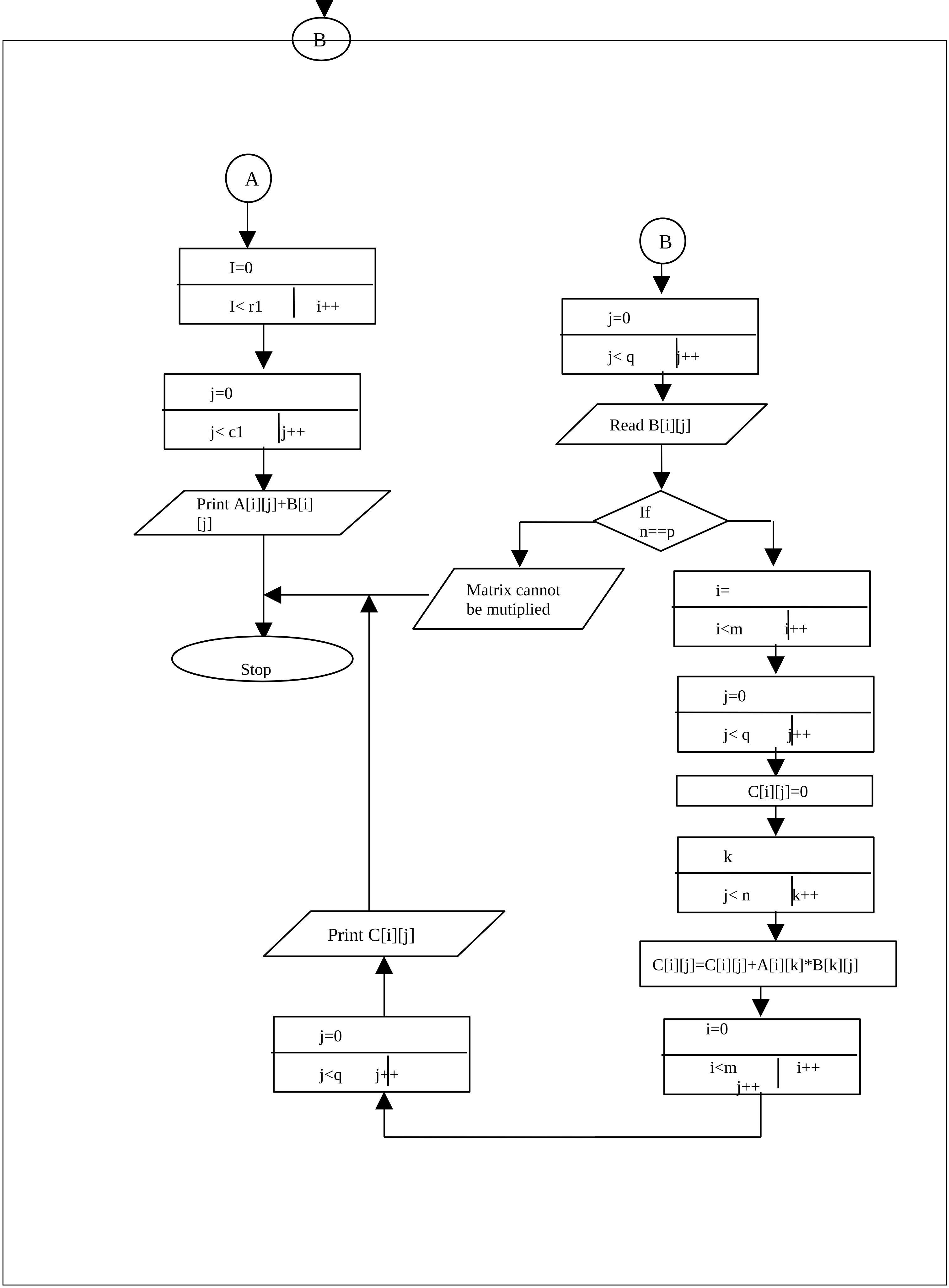
```
        verify matrix A
        read the size of matrix B
        read matrix B
        verify matrix B
        calcSumMatrixAB and store in C
        verify matrix C
        calcProductMatrixAB and store in C
        verify matrix C
    stop
    read Matrix (M, rowmax, colmax)
    start
        loop i=0 to rowmax-1
            loop j=0 to colmax-1
            M[i][j]=value
    stop

    verify Matrix(M, rowmax, colmax)
    start
        loop i=0 to rowmax-1
            loop j=0 to colmax-1
            print M[i][j]
    stop
    add Matrix(A, B, C, rowmax, colmax)
    start
        loop i=0 to rowmax-1
            loop j=0 to colmax-1
            C[i][j]  = A[i][j] + B[i][j];
```

```
Stop
```

```
mul Matrix(A, B, C, rowmaxA, rowmaxB, colmaxB)
start
    loop i=0 to rowmaxA -1
        loop j=0 to colmaxB -1
            C[i][j] = 0;
            loop k=0 to rowmaxB - 1
                C [i][j] = C[i][j] + A[i][k] * B[k][j];
stop
```

Flow chart:


start
Declare a[ ][ ],b[ ][ ],c[ ][ ],ch,I,j,k,m,n,p,q,r1, c 1
Read the choice ch
If ch<=2 &ch>
Printvalid choice
Case 1
Switch ch
Case 2
Read the size of A,B matrix m,n
I=0
I< r1
i++
J=0
jI< c1
ji++
Read A[i][j]
I=0
I< r1
i++
j=0
jI<c1
j++
Read B[i[j]
Read the size of A matrix m,n
I=0
I< r1
i++
j=0
j<c1
j++
Read A[i][j]
Read size of matrix B: p.q
I=0
I< p
i++
B


B
A
I=0
I< r1
i++
j=0
j< c1
j++
Print A[i][j]+B[i]
[j]
Stop
B
j=0
j< q
j++
Read B[i][j]
If
n==p
Matrix cannot
be mutiplied
i=
i<m
i++
j=0
j< q
j++
C[i][j]=0
k
j< n
k++
C[i][j]=C[i][j]+A[i][k]*B[k][j]
i=0
i<m
i++
j++
Print C[i][j]
j=0
j<q
j++

Program:

```
            /* To perform addition and multiplication of two matrices by using functions.*/

#include<conio.h>
#include<stdio.h>
void read_Matrix ( int x[10][10], int rows, int columns )
{
        int i,j;
        for ( i=0; i<rows; i++)
         {
                for ( j=0; j<columns; j++)
                 {
                        scanf("%d",&x[i][j]);
                 }
         }
 }

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

```
     }
 }

void add_Matrix ( int x[10][10], int y[10][10], int z[10][10], int rows, int columns )
 {
    int i,j;
    for ( i=0; i<rows; i++ )
     {
        for ( j=0; j<columns; j++ )
         {
            z[i][j] = x[i][j] + y[i][j];
         }
     }
 }

void mult_Matrix ( int x[10][10], int y[10][10], int z[10][10], int rows1, int rows2, int columns2)
{

    int i,j,k;
    for ( i=0; i<rows1; i++ )

    {
        for ( j=0; j<columns2; j++ )
         {
            z[i][j] = 0;
            for ( k=0; k<rows2; k++ )
             {
                z[i][j] = z[i][j] + ( x[i][k] * y[k][j] );
            }
        }
    }
}

void main ()
 {
    int a[10][10],b[10][10],c[10][10];
    int m,n,o,p;
    clrscr();
    printf(" Enter the order of 1st matirx......\n");
    scanf("%d%d",&m,&n);
    printf("\n Enter the elements in the %dX%d matrix..... \n",m,n);
    read_Matrix( a,m,n );
    printf("\n Matrix a is as follows...... \n\n");
    print_Matrix( a, m,n );
    getch();
    printf("\n\n Enter the order of 2st matirx......\n");
    scanf("%d%d",&o,&p);
    printf("\n\n Enter the elements in the %dX%d matrix..... \n",o,p);
    read_Matrix( b, o, p );
    printf("\n Matrix b is as follows...... \n\n");
    print_Matrix( b, o, p );
    getch();
    if ( m==o && n==p )
    {
        add_Matrix( a, b, c, m, n);
```

```
                printf("\n\n Sum of two matrices is ...... \n\n");
                print_Matrix( c, m, n );


        }
         else
        {
                printf("\n\n To do addition of two matrices.... the orders of both the matrices
                should be equal..... \n");
         }
        getch();
        if ( n==o )
        {


                mult_Matrix( a, b, c, m, o, p );
                printf("\n\n Multiplication of two matrices is ...... \n\n");
                print_Matrix( c, m, p );
         }
        else
         {
                printf("\n\n To do multiplication of two matrices.... the columns of first matrix
                should be equal to the columns of second matrix..... \n");
         }
        getch();
}
```

Result:

Viva Questions:

1. What is an Array?
2. Differentiate between size and Index of an array?
3. Explain different types of arrays?

WEEK-9

Week-9(a)(i)

AIM :

To insert a sub string into given main string from a given position.

Description:

In this program we insert any sub string into given string from a given position.

e.g.: First string cplab

Second string ds

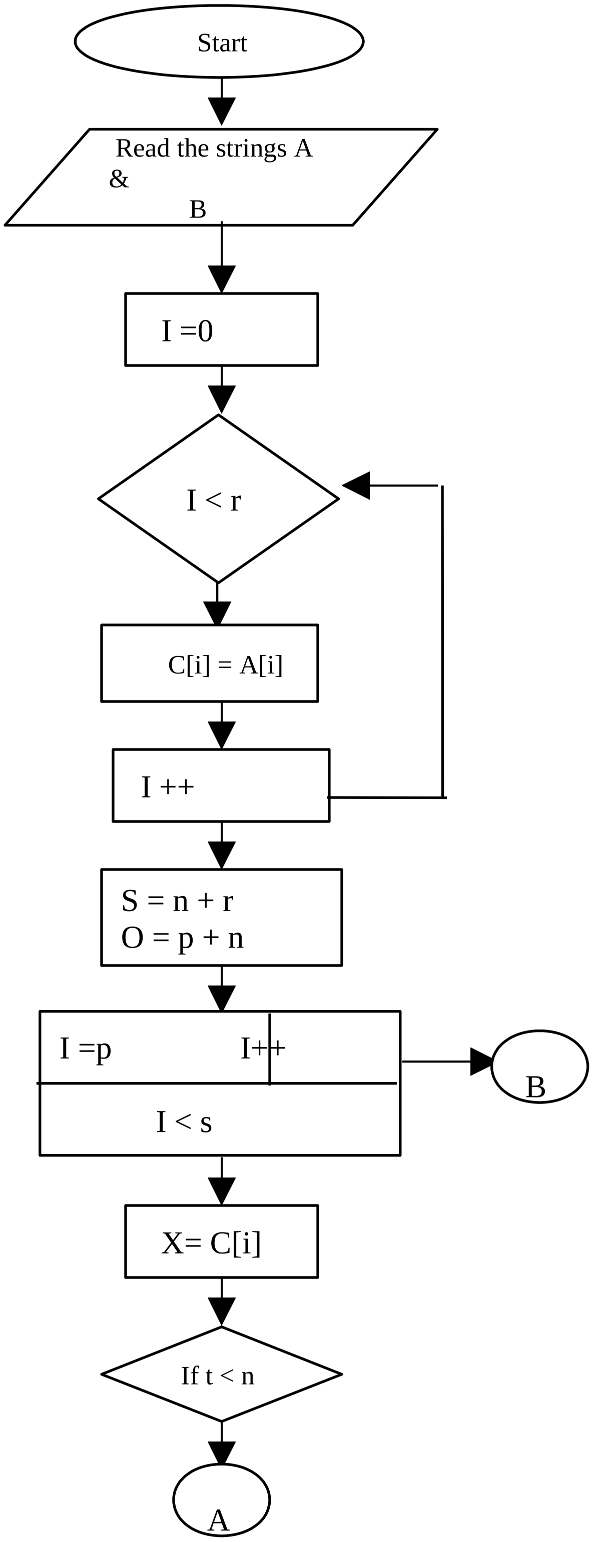
Enter position: 2

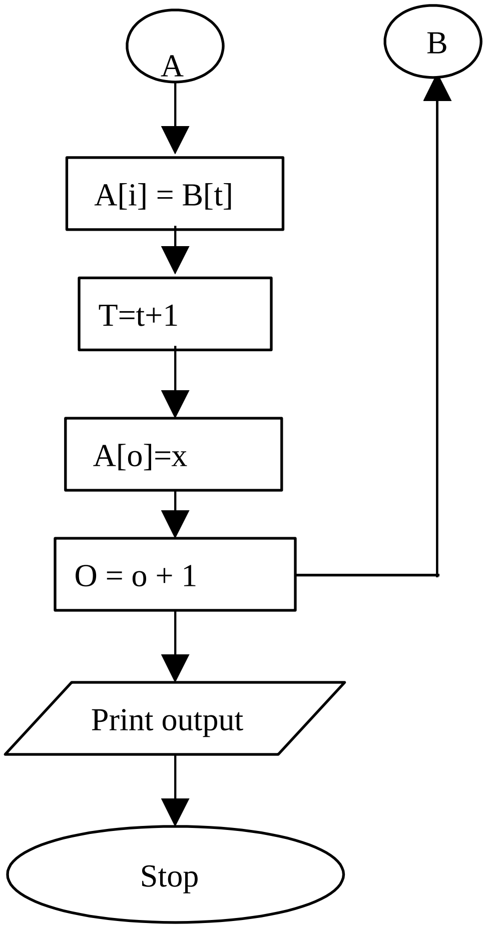
Result is cpdslab

Algorithm:

```
start
    declare string variables for first string, second string and output string
    get first string
    get second string
    get position
    split first string using strncpy
    use strcpy and strcat to generate output string
stop
```

Flow chart:


Start
Read the strings A
&
B
I =0
I < r
C[i] = A[i]
I ++
S = n + r
O = p + n
I =p
I++
I < s
B
X= C[i]
If t < n
A


A
B
A[i] = B[t]
T=t+1
A[o]=x
O = o + 1
Print output
Stop

Program:

```
 /* Functions to insert a sub string into given main string from a given position*/
#include<stdio.h>
#include<string.h>
void main()
{
            char str[20],str1[20],str2[20]="";
            int n;
            clrscr();
            printf("enter string\n");
            gets(str);
            printf("enter second string\n");
            gets(str1);
            printf(" enter the position where the substring is to be inserted \n");
            scanf("%d",&n);
            strncpy(str2,str,n);
            strcat(str2,str1);
            strcat(str2,str+n);
            puts(str2);
}
```

Result:

```
Input                                                              Output
enter string :cplab
enter second string:ds
enter the position where the substring is to be inserted:2            cpdslab
```

Week-9(a)ii

AIM :

To delete n characters from a given position in a given string.

Description:

In this program we need to delete n chars form a given position in a string.

e.g.: Enter string is cpdslab

Enter position is: 2

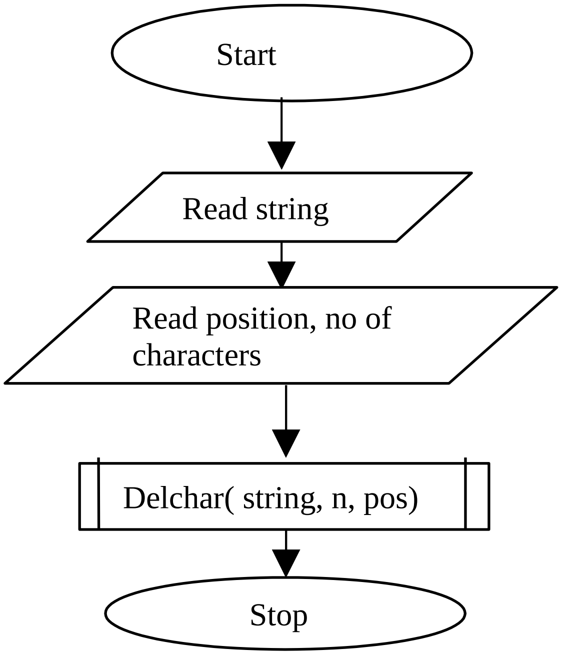
The number of characters to be deleted: 2

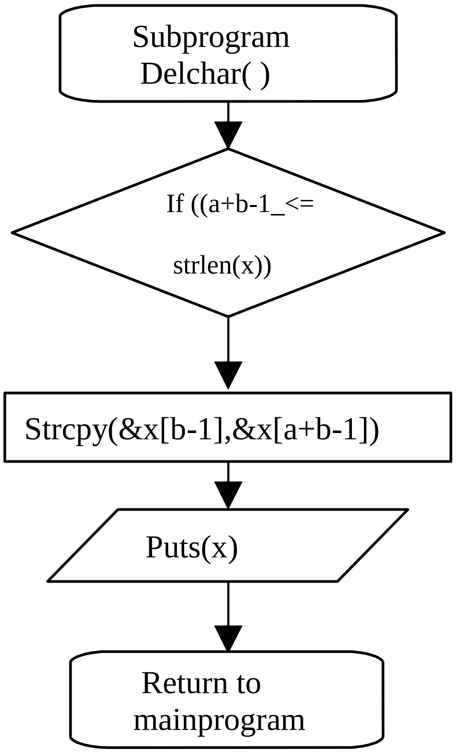
Result is: cplab

Algorithm:

```
start
    declare string variables for first string.
    declare position and number of characters to be deleted.
    get first string
    get position and number
    use strcpy to generate output string.
stop
```

Flow chart:


Start
Read string
Read position, no of characters
Delchar(string, n, pos)
Stop


Subprogram Delchar( )
If ((a+b-1_<=
strlen(x))
Strcpy(&x[b-1],&x[a+b-1])
Puts(x)
Return to mainprogram

Program:

```
/* To delete n characters from a given position in a given string.*/
#include<stdio.h>
#include<string.h>
void main()
{
        char str[20],str2[20]="";
        int n,pos;
        clrscr();
        printf("enter string\n");
        gets(str);
        printf("enter position\n");
        scanf("%d",&pos);
        printf(" enter number of charecters tobe delete");
        scanf("%d",&n);
        strncpy(str2,str,pos);
        strcat(str2,str+pos+n);
        puts(str2);
}
```

Result:

```
Input                                           Output
cpdslab
enter position  :2
enter number of charecters tobe delete:2       cplab
```

Week-9(b)

AIM :

To determine if the given string is a palindrome or not.

Description:

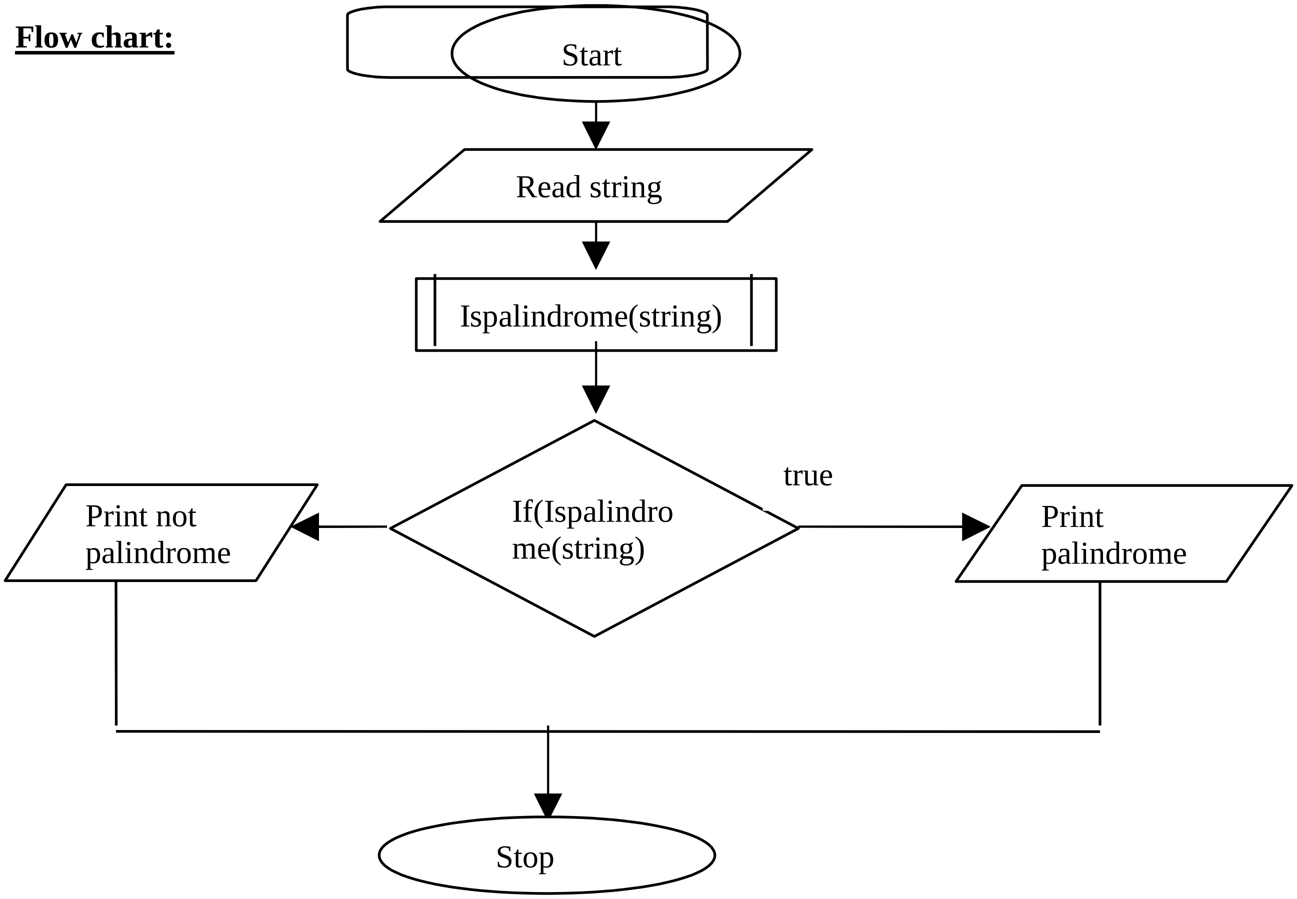
A palindrome is a string or a number that can be read the same way in either direction.

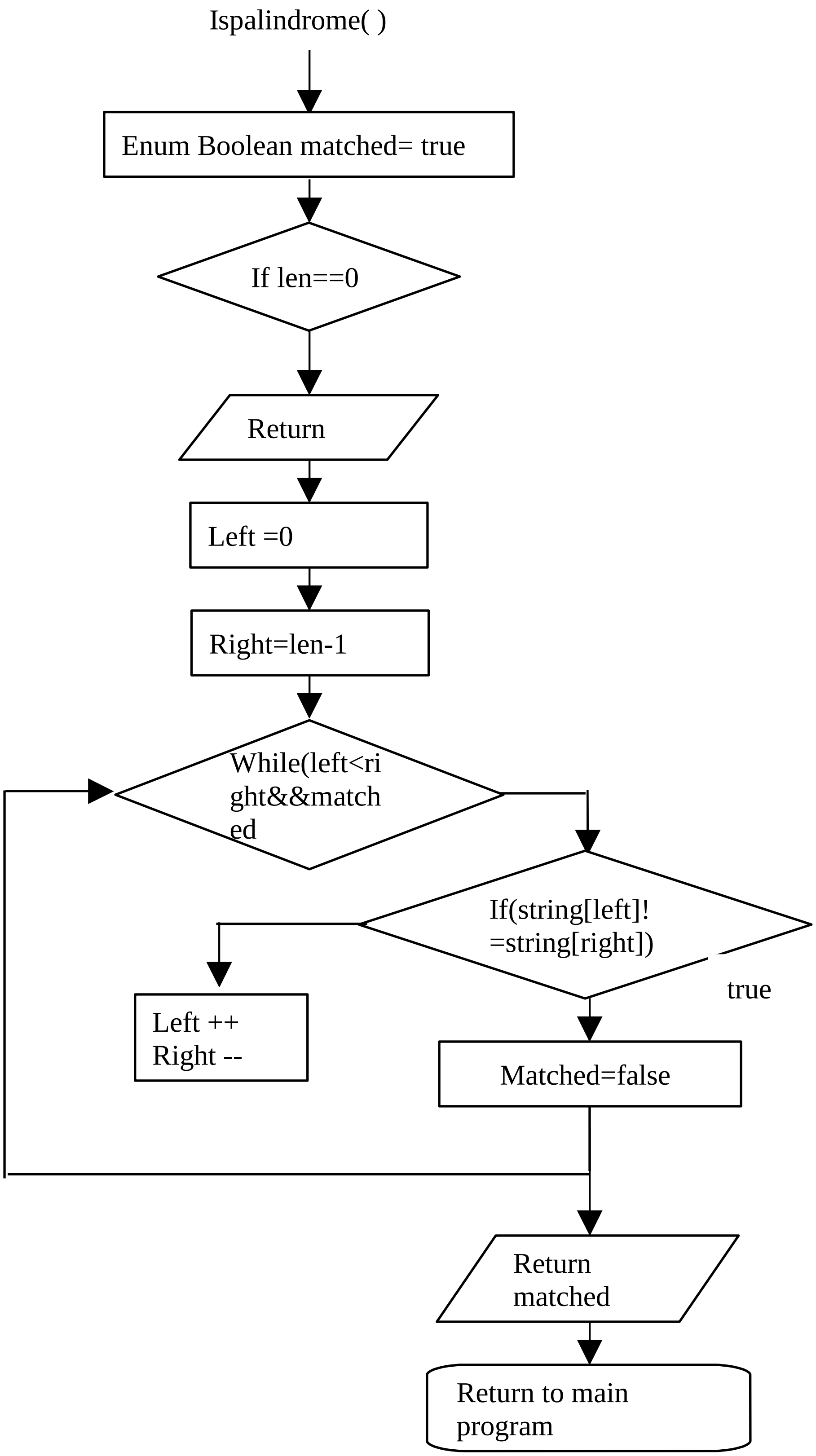
e.g.: “madam”

Algorithm:

```
start
    declare string variables for string
    get string
     use strrev and strcmp to determine string is palindrome
stop
```

Flow chart:


Start
Read string
Ispalindrome(string)
true
Print not palindrome
If(Ispalindrome(string)
Print palindrome
Stop


Ispalindrome( )
Enum Boolean matched= true
If len==0
Return
Left =0
Right=len-1
While(left<right&&matched
If(string[left]!=string[right])
true
Left ++
Right --
Matched=false
Return matched
Return to main program

Program:

/* To write a program to determine if the given string is a palindrome or not.*/

```
#include<stdio.h>

#include<String.h>

void main()

 {

        int temp;

        char str[20],str1[20];

        clrscr();

        printf(" Enter a string : ");

        gets(str);

        strcpy(str1,str);

        strrev(str1);

        temp=strcmp(str1,str);

        if ( temp==0 )

                printf("\n\n Entered stirng is a palindrome");

        else

                printf("\n\n Entered stirng is not a palindrome");

        getch();

 }
```

Result:

Input	Output
madam	the given string is palindrome

<u>Viva Questions:</u>

1. What is string data type in C?
2. How it is declared?
3. What is NULL character and what is its usage?
4. Why string.h header file is used?
5. Explain the following functions:

    strlen

    strcmp

    strcat

8. What is meant by palindrome?

9. What is the use of gets () and puts functions?

10. What is the difference between printf () and puts ()?

WEEK-10

Week-10(a)

Write a C program to replace a substring with another in a given line of text.

```
#include <stdio.h>
#include <string.h>
void main()
{
        char text[100],word[10],rpwrd[10],str[10][10];
        int i=0,j=0,k=0,w,p;

    printf("PLEASE WRITE ANY TEXT.\n");
    printf("GIVE ONLY ONE SPACE AFTER EVERY WORD\n");
    printf("WHEN COMPLETE PRESS  Ctrl-Z  \n");
    gets(text);
    printf("\nENTER WHICH WORD IS TO BE REPLACED\n");
    scanf("%s",word);
    printf("\nENTER BY WHICH WORD THE %s IS TO BE REPLACED\n",word);
    scanf("%s",rpwrd);
    p=strlen(text);

    for (k=0; k<p; k++)
         {

          if (text[k]!=' ')
            {
             str[i][j] = text[k];
             j++;
            }
          else
            {
             str[i][j]='\0';
             j=0; i++;
            }
        }
        str[i][j]='\0';
        w=i;
     for (i=0; i<=w; i++)
        {
           if(strcmp(str[i],word)==0)
             strcpy(str[i],rpwrd);

           printf("%s ",str[i]);
     }
}
```

Output:

```
PLEASE WRITE ANY TEXT.
GIVE ONLY ONE SPACE AFTER EVERY WORD
welcome to mlritm

ENTER WHICH WORD IS TO BE REPLACED
mlritm

ENTER BY WHICH WORD THE mlritm IS TO BE REPLACED
cse
welcome to cse
```

Week-10(b)

**Write a C program that reads 15 names each of up to 30 characters, stores them in an array,
and uses an array of pointers to display them in ascending (ie. alphabetical) order.**

```
#include <stdio.h>
#include <string.h>
void main()
{
   char name[10][8], tname[10][8], temp[8];
   int i, j, n;
   printf("Enter the value of n \n");
   scanf("%d", &n);
   printf("Enter %d names n", \n);
   for (i = 0; i < n; i++)
   {
      scanf("%s", name[i]);
      strcpy(tname[i], name[i]);
   }
   for (i = 0; i < n - 1 ; i++)
   {
      for (j = i + 1; j < n; j++)
      {
         if (strcmp(name[i], name[j]) > 0)
         {
            strcpy(temp, name[i]);
            strcpy(name[i], name[j]);
            strcpy(name[j], temp);
         }
      }
   }
   printf("Input NamestSorted names\n");
   for (i = 0; i < n; i++)
   {
      printf("%s\t\t%s\n", tname[i], name[i]);
   }

}
```

Output:

```
Enter the value of n
2
Enter 2 names
MLRITM
MLRIT
Input NamestSorted names
MLRITM              MLRIT
MLRIT               MLRITM
```

WEEK-11

Week-11(a)

AIM :

To find the 2’s complement of a binary number.

Description:

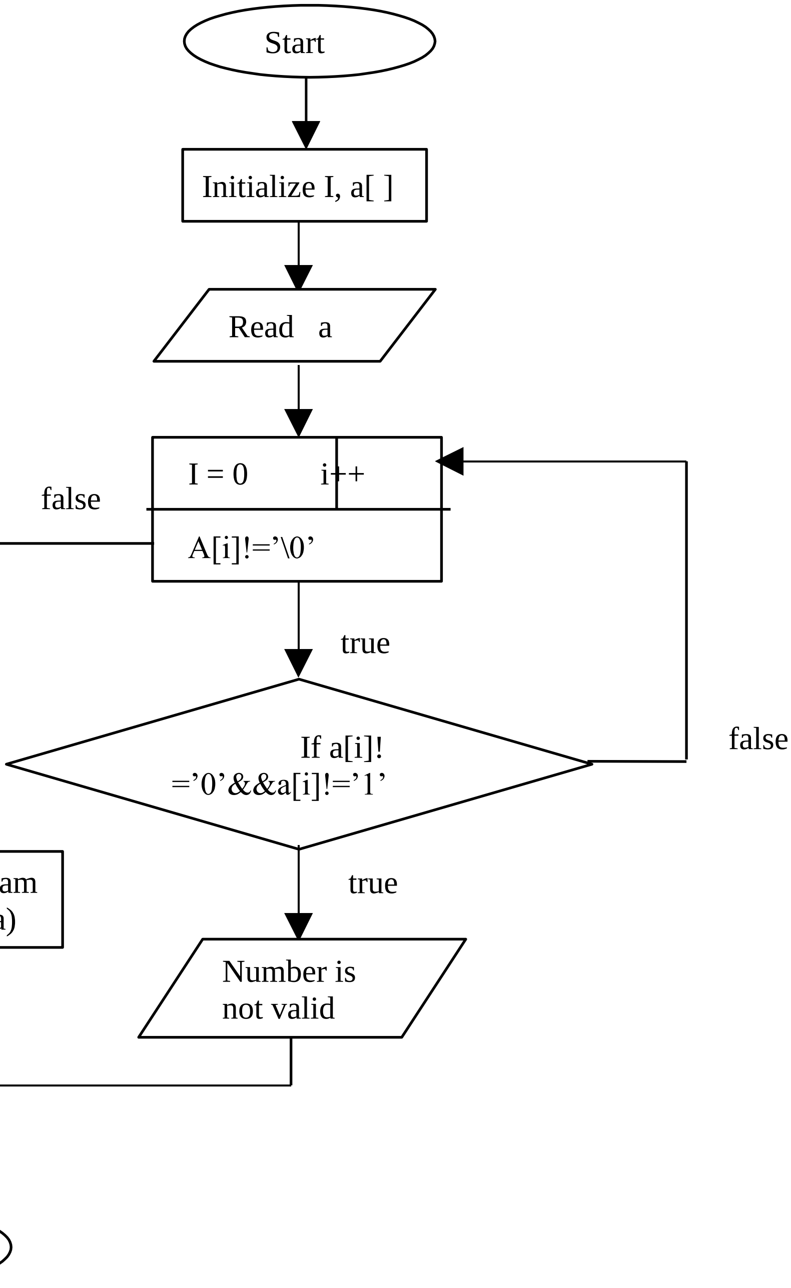
2’s complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2’s complement

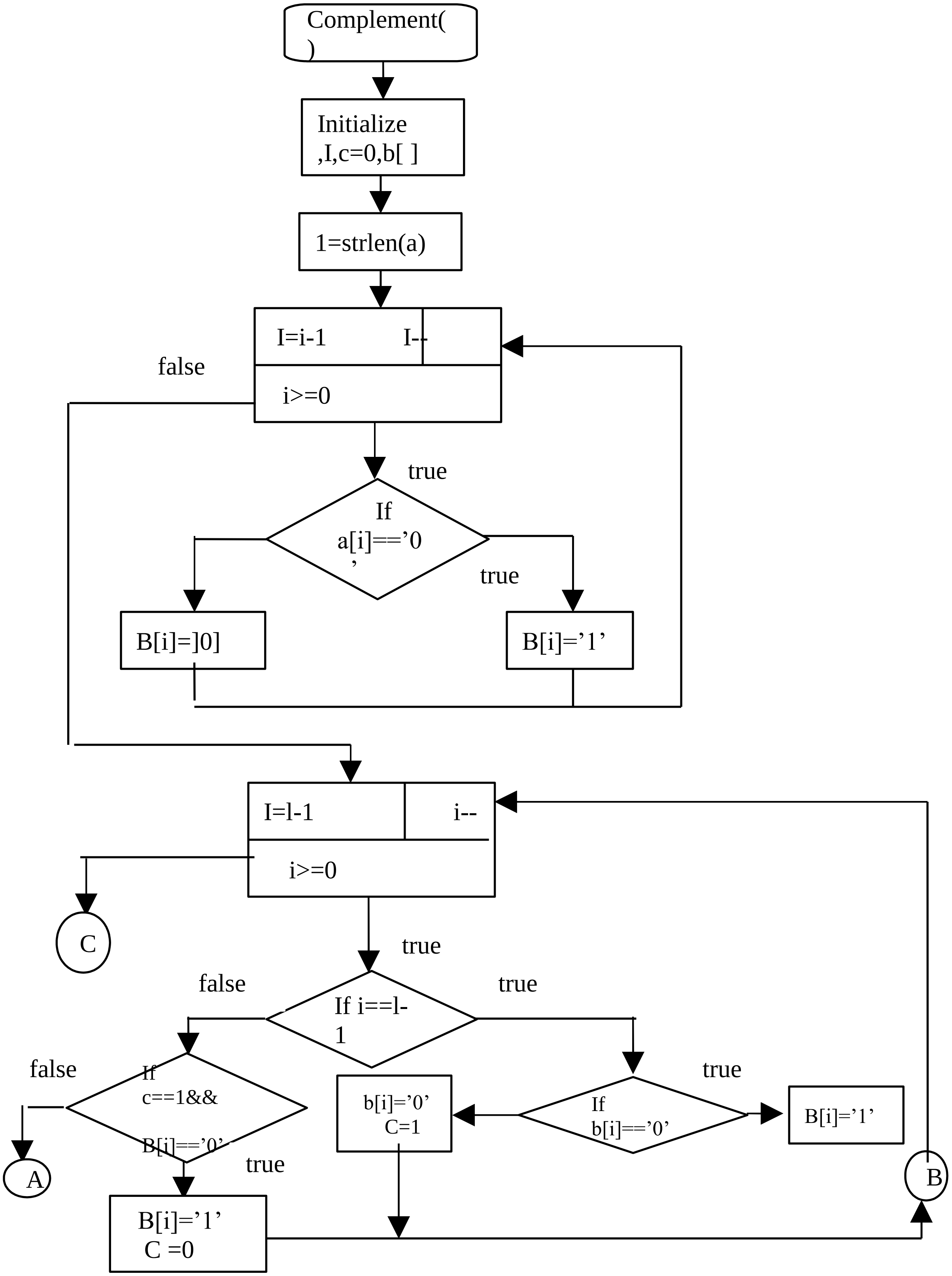
of 11100 is 00100.

Algorithm:

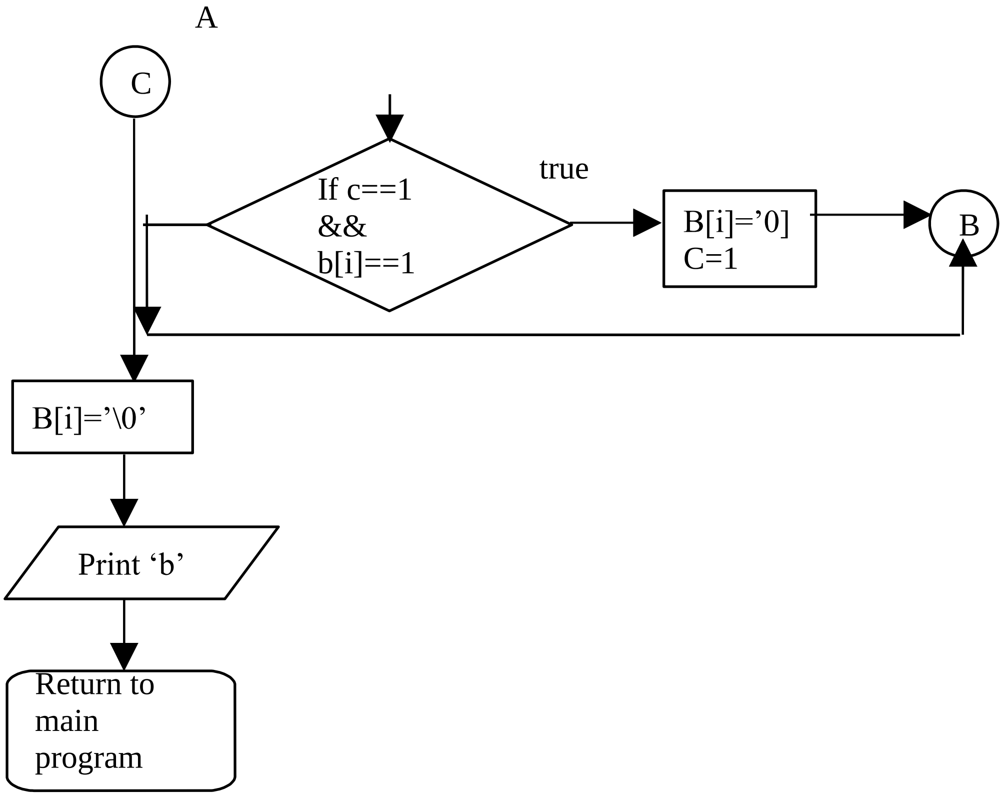
```
start
    read string
    loop
        scan the elements from last of the string
        if ( element is ‘1’)
            set foundFlg
            break;
    end loop
    if (foundFlg)
    loop
        if (element is ’1’)
            element = ‘0’
        else
            element = ‘1’
        end loop
    print the string
stop
```

flowchart:


Start
Initialize I, a[ ]
Read a
I = 0
i++
A[i]!='\0'
false
true
If a[i]!='0'&&a[i]!='1'
false
true
Call sub program Complement(a)
Number is not valid
Stop



Complement( )
Initialize ,I,c=0,b[ ]
1=strlen(a)
I=i-1
I--
i>=0
false
true
If a[i]=='0 '
true
B[i]=]0]
B[i]='1'
I=l-1
i--
i>=0
C
true
false
true
If i==l-1
false
If c==1&& B[i]=='0'
true
A
B[i]='1' C =0
b[i]='0' C=1
If b[i]=='0'
true
B[i]='1'
B


A
C
If c==1
&&
b[i]==1
true
B[i]='0]
C=1
B
B[i]='\0'
Print 'b'
Return to
main
program

Program:

```
/* Write a C program to find the 2’s complement of a binary number.*/


#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[10];
    char temp[10];
    int i,j,found;
    found = 0;
     puts("Enter a binary number as 1's and 0': ");
    gets(str);
    strcpy(temp, str);
     j=strlen(str);
    for(i=j-1;i>=0;i--)
    {
        if(str[i]=='1') str[3]=
        {
            found = 1;
            break;
        }
    }
     if (found)
    {
        for(i=j-1;i>=0;i--)
        {
            if(str[i]=='1')
                str[i] = '0';
            else
                str[i] = '1';
        }
    }
    printf ("2's complement of %s is %s\n", temp, str);
    getch();
}
```

Result:

Input	Output
10010	01110
00000	00000
11100	00100

VIVA QUESATIONS:

1) Expand ASCII ?
2) What is binary number ?
3) Define 2"s complement ?

Week-11(b)

AIM :

To convert a Roman numeral to its decimal equivalent.

Description:

Roman numerals, as used today, are based on seven symbols:

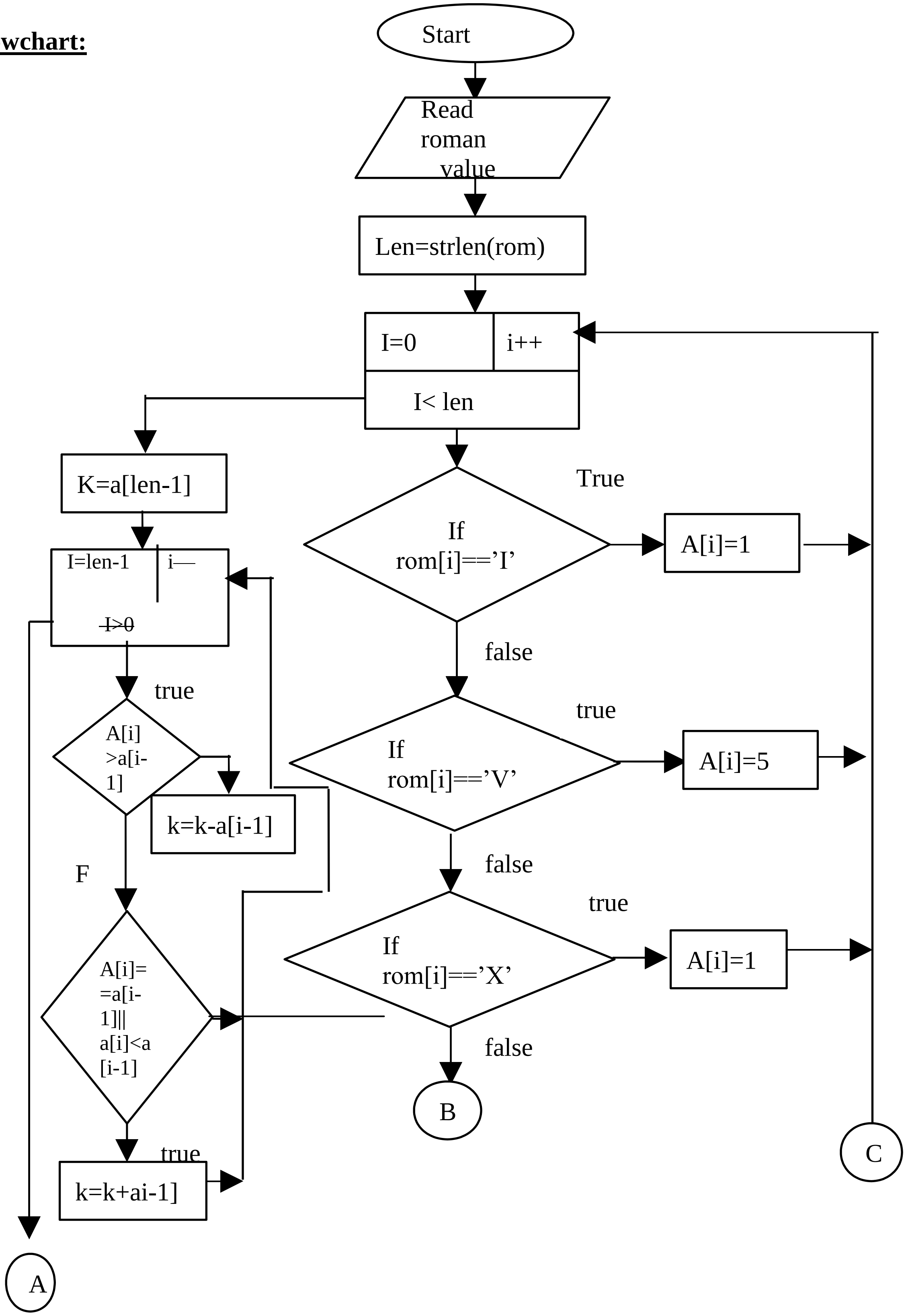
Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

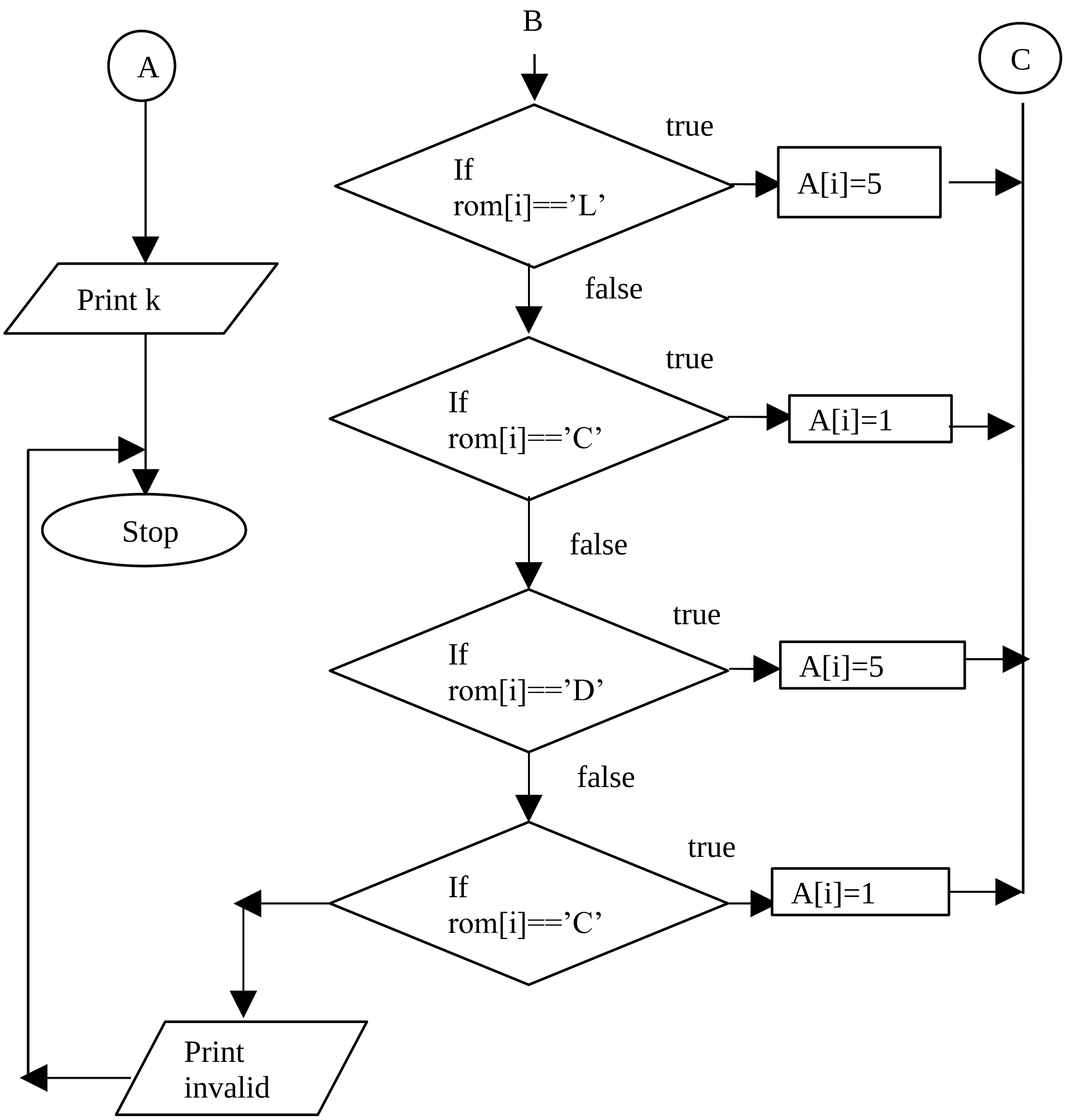
Roman Numbers are formed by combining symbols together and adding the values. For example, MMXII is 1000 + 1000 + 10 + 1 + 1 = 2012. Generally, symbols are placed in order of value, starting with the largest values. When smaller values precede larger values, the smaller values are subtracted from the larger values, and the result is added to the total. For example IV = (5 − 1) = 4.

Algorithm:

```
start
        read roman number string
        loop
            assign decimal values to the roman symbols in the string
        end loop
        loop
            scan the string from right to left
            if value on right  > value on left
                number  +=  (right – left)
            else
                number  +=  (right + left)
        end loop
        print number
stop
```

flowchart:


Start
Read roman value
Len=strlen(rom)
I=0
i++
I< len
K=a[len-1]
I=len-1
i—
~~I>0~~
true
A[i] >a[i-1]
k=k-a[i-1]
F
A[i]= =a[i-1]|| a[i]<a [i-1]
true
k=k+ai-1]
A
True
If rom[i]==’I’
A[i]=1
false
true
If rom[i]==’V’
A[i]=5
false
true
If rom[i]==’X’
A[i]=1
false
B
C


B
A
C
If
rom[i]=='L'
true
A[i]=5
Print k
false
If
rom[i]=='C'
true
A[i]=1
Stop
false
If
rom[i]=='D'
true
A[i]=5
false
If
rom[i]=='C'
true
A[i]=1
Print
invalid

Program:

```
/* Write a C program to convert a Roman numeral to its decimal equivalent.*/

#include<stdio.h>
#include<conio.h>
#include<string.h>
int  main()
{
        int len,i,j, decimal;
        int a[10];
        char rom [10];
        printf("Enter the Roman Numeral:");
        scanf("%s",rom);
         len=strlen(rom);
        for(i=0;i<len;i++)
        {
                if(rom[i]=='I')
                        a[i]=1;
                 else if(rom[i]=='V')
                        a[i]=5;
                 else if(rom[i]=='X') ….
                        a[i]=10; a[1]=10
                else if(rom[i]=='L')
                        a[i]=50;
                else if(rom[i]=='C')
                        a[i]=100;
                else if(rom[i]=='D')
                         a[i]=500;
                else if(rom[i]=='M')
                        a[i]=1000;
                else
                {
                        printf("\nInvalid Value");
                        getch();
                        return -1;
                 }
        }
        decimal=a[len-1];
        for(i=len-1; i>0; i--)
        {
                if(a[i]>a[i-1])
                        decimal-= a[i-1];
                else
                        decimal+= a[i-1];
        }
        printf("\nIts Decimal Equivalent of %s is %d\n", rom, decimal);
        return 0;
}
```

Result:

Input	Output
D	500
X	10
23	Invalid value

Viva Questions:

1.What is a binary number?

2.What is a bit and what is byte?

3.What are the two different way to implement multiway selection?

4.What is structure of else-if ladder?

5.What is nested-if?

WEEK-12

Week-12(a)

AIM :

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

Description:

Need to print content of an existing file.

e.g.: In cpds.txt file has the content “Welcome to CPDS lab - do bring your thinking hats”

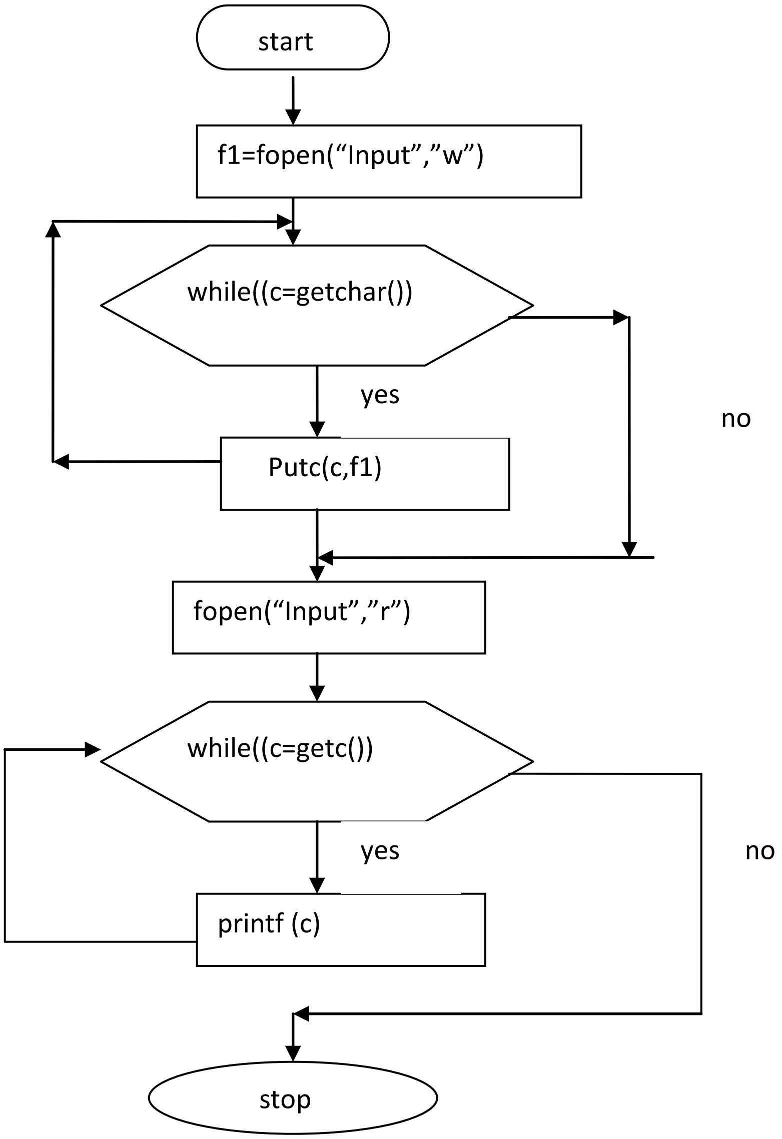
This content should be read from the file and displayed on the monitor.

Note**:** Source file should exist in path c:\tc\userid **(**default**)**

Algorithm:

```
start
    declare file pointers and character variable.
    read file name.
    open file with given file name in read mode.
    if the file pointer is null
        print error message
        read the characters until EOF
        print characters
    close file
stop
```

flowchart:


start
f1=fopen(“Input”,”w”)
while((c=getchar())
yes
no
Putc(c,f1)
fopen(“Input”,”r”)
while((c=getc())
yes
no
printf (c)
stop

<u>Program:</u>

```
    /*C program to read the contents of a file*./
    #include <stdio.h>
    #include <conio.h>
    #include <process.h>
    void main()
    {   FILE *fs;
        char ch;
        clrscr();
        fs = fopen("source.txt","r");
         if(fs==NULL)           {
            puts(" 1st Source file doesn't exist.");
            getch();
            exit(0);            }
        while(1)            {
ch=fgetc(fs);
            if (ch==EOF)
                break;
            else
                putchar(ch);        }
        getch();}
```

<u>Result:</u>

<u>Input</u>	<u>Output</u>
cpds.txt	Welcome to CPDS lab -Do bring your thinking hats

Week-12(b)

AIM :

To copy one file to another file.

Description:

We have to use the file functions to perform the copy operation from one file to another file.

e.g.: cpds.c file contains

This is cpds lab….

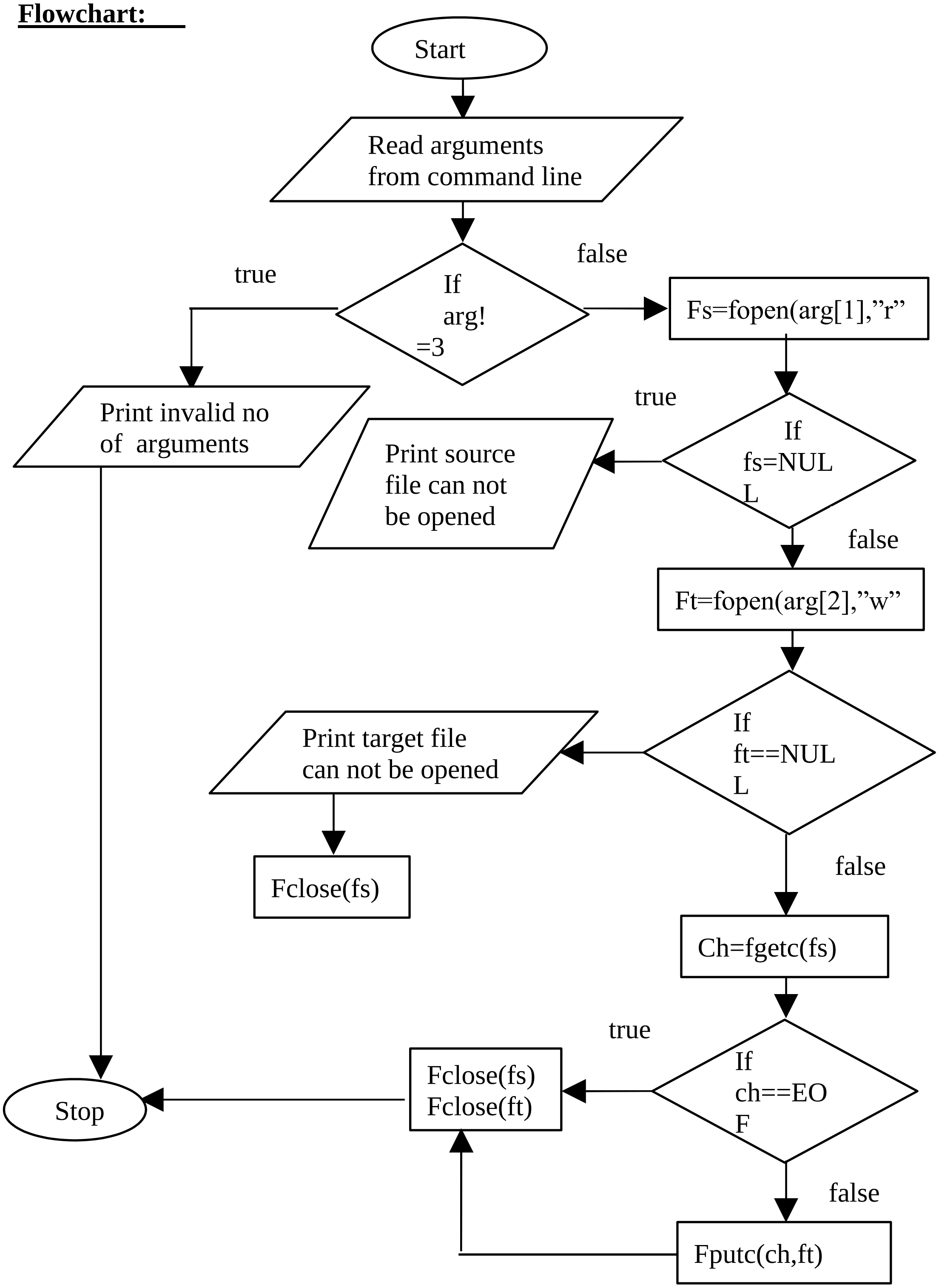
Manual.c

This is cpds lab…. (This is copy file from cpds.c)

Algorithm:

```
start
        declare two file pointers and variable.
        open file with file pointer as read mode // the file should exist in tc/bin
        condition file pointer is null
                cannot open file
        else
                second file open as write mode
                read characters up to EOF and copy into target file
        close two files
stop
```

Flowchart:


Start
Read arguments from command line
true
false
If arg! =3
Fs=fopen(arg[1],”r”
Print invalid no of arguments
true
If fs=NULL
Print source file can not be opened
false
Ft=fopen(arg[2],”w”
If ft==NULL
Print target file can not be opened
Fclose(fs)
false
Ch=fgetc(fs)
true
If ch==EOF
Fclose(fs)
Fclose(ft)
Stop
false
Fputc(ch,ft)

Program:

```
 /* C program which copies one file to another file. */
#include <stdio.h>
#include <conio.h>
#include <process.h>
void main()
{
         FILE *fs,*ft;
         char ch;
         clrscr();
         fs = fopen("source.txt","r");
         if(fs==NULL)
        {
                 puts("Source file cannot be opened.");
                exit(0);
         }
        ft = fopen("target.txt","w");
         if (ft==NULL)
        {
                puts("Target file cannot be opened.");
                fclose(fs);
                exit(0);
         }
         while(1)
         {
                ch=fgetc(fs);
                 if (ch==EOF)
                break;
            else
                fputc(ch,ft);
        }
        fclose(fs);
        fclose(ft);
        getch();
}
```

Result:

Note:

Source file should exist in tc/bin

Input	Output
in source file write something	destination file displays the contents same in source file

WEEK-13

Week-13(a)

AIM :

To count the lines, words and characters in a given text.

Description:

In this program we will read text and we will find and display no. of lines, words and characters in that given text.

e.g.:Text: sreyas institute of engineering & technology

besides indu aranya

No. of lines = 2

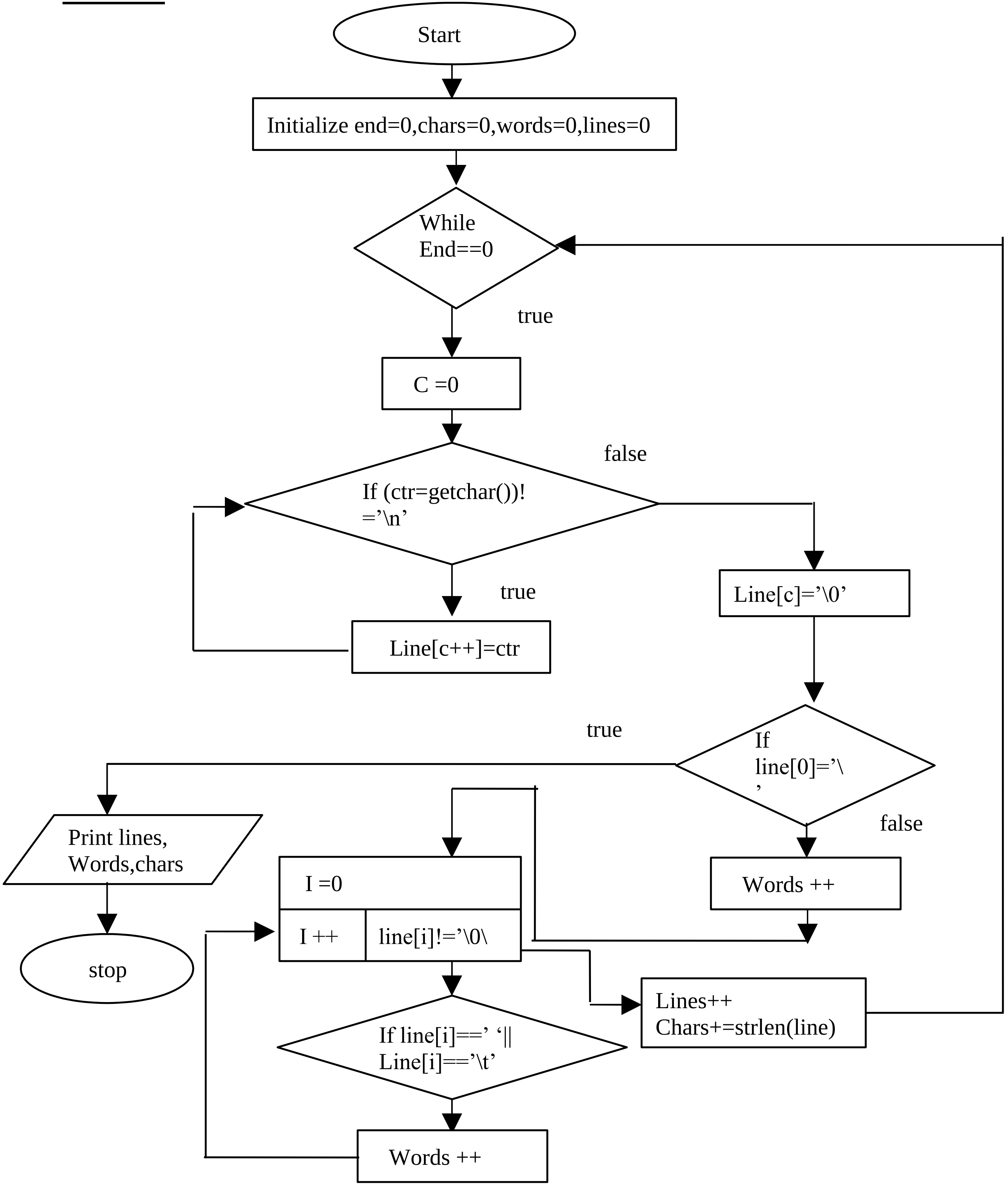
No of words = 9

No of characters =62

Algorithm:

```
start
    read the text until an empty line.
    compare each character with newline char ‘\n’ to count number of lines.
    compare each character with tab char ‘\t\’ or space char ‘ ‘ to count number
    of words.
    compare first character with NULL char ‘\0’ to find the end of text.
    number of characters = length of each line of text.
    print number of lines, number of words, number of chars.
stop
```

flowchart:


Start
Initialize end=0,chars=0,words=0,lines=0
While
End==0
true
C =0
false
If (ctr=getchar())!
='\n'
true
Line[c++]=ctr
Line[c]='\0'
true
If
line[0]='\
'
false
Print lines,
Words,chars
stop
I =0
I ++
line[i]!='\0\
Words ++
Lines++
Chars+=strlen(line)
If line[i]==' '||
Line[i]=='\t'
Words ++

Program:

```
            /*Write a C program to count the lines, words and characters in a given text.*/
           #include<string.h>
           #include<stdio.h>
           void main()
           {
                   char ch;
                   int line=0,chars=0, words=0;
                   printf(" Enter text\n");
                   while((ch=getchar())!=EOF)
                   {

                           chars++;
                           if(ch=='\n')
                                   line++;
                           if(ch=='\t' || ch=='\n'|| ch==' ')
                                   words++;

                   }
                    printf(" no of lines are %d\n  no of words are %d \n no of characters are
                   %d",line,words, chars);

 }
```

Result:

```
Input                          Output
Welcome to the  cpds lab       No. of lines = 3
We had 24 weeks programs       No of words = 13
All the best                   No of characters =59
```

Week-13(b)

Write a C program to compare two files, printing the first line where they differ.

```
        #include <stdio.h>
        #include<string.h>
        void main()
        {
           char c1[1000], c2[1000];
           FILE *f1;
           FILE *f2;
           f1 = fopen ("output.txt", "r");
           f2 = fopen ("output_copy.txt", "r");
           printf("\n");
           while((fgets(c1,1000,f1)!= NULL) && (fgets(c2,1000,f2)!= NULL))
           {
              if(strcmp(c1,c2) != 0)
              printf("%s", c1);
              printf("%s", c2);

           }
           printf("\n");
           fclose(f1);
           fclose(f2);
}
```

Output:

```
welcome to mlritm
welcome to iare
```

WEEK-14

<u>Week-14(a)</u>

Write a C program to change the nth character (byte) in a text file. Use fseek function.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
        FILE *fp;
        char ch;
        int n;
        fp = fopen("random.txt","r+");
        printf("Enter n value\n");
        scanf("%d",&n);

        printf("Enter charcter value\n");
        scanf(" %c",&ch);
        fseek(fp,n-1,0);
        fputc(ch,fp);
        fclose(fp);

 }
```

Before excution

Random.txt

```
abcdefghijklmnopqrstuvwxyz
```

Output:

```
Enter the n value
4
Enter the character
a
```

after execution

```
abca abcdefghijklmnopqrstuvwxyz
```

Week-14(b)

AIM :

To reverse the first n characters in a file.

(Note: The file name and n are specified on the command line.)

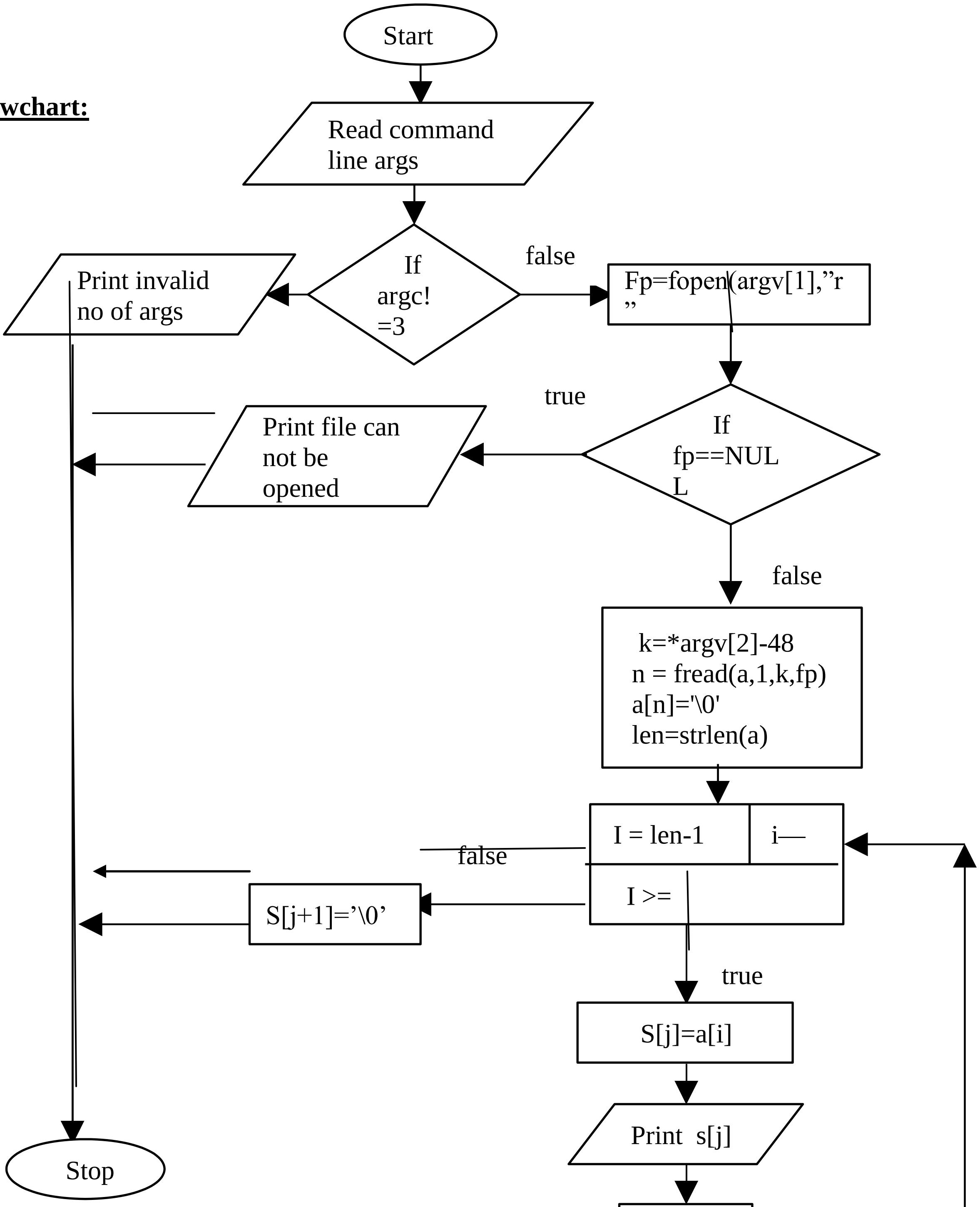
Description:

This program perform the reverse operation of n characters in the file

Algorithm:

```
start
        declare file pointer, variable
        argv [1] assign to n
        open file with read mode with argv [2]
        if file pointer is NULL
                Give error message
        loop up to n
                read values and store in string1
        end loop
        reverse the string1 and copy into str2
        print str2
        close file pointer
stop
```

Flowchart:


Start
Read command line args
If argc! =3
false
Print invalid no of args
Fp=fopen(argv[1],"r "
true
If fp==NULL
Print file can not be opened
false
k=*argv[2]-48
n = fread(a,1,k,fp)
a[n]='\0'
len=strlen(a)
I = len-1
i—
false
I >=
S[j+1]='\0'
true
S[j]=a[i]
Print s[j]
J++
Stop

Program:

```
/* 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.)*/


#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();
        }
```

Result:

Input	Output
Filerev file1.c 7	si siht

Viva Questions:

1.What is file?

2.What are the basic modes of a File in C?

3.When process.h headerfile is included?

4. What are the String manipulation functions?

5. What is buffer?

6. What is the difference between getc and getchar?

WEEK-15

Week-15(a)

AIM :

To merge the two files into a third file.

Description:

Read contents of two existing files, merge second file content to the first file and put it in to the third file.

e.g.: file1, cplab.txt contains This is cpds lab….

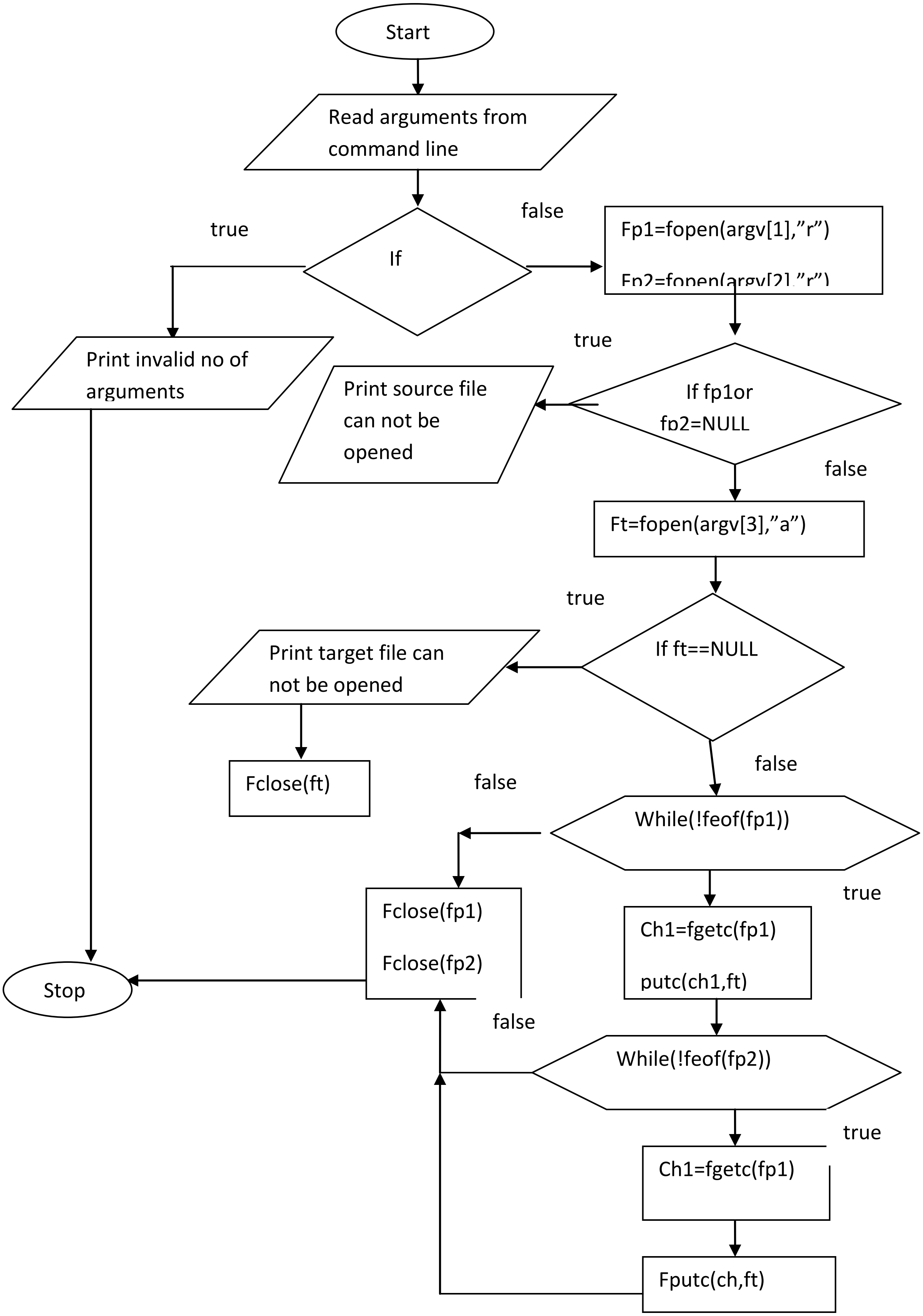
file2, manual.txt contains It has 21 weeks programs….

file3, cds.txt should contain This is cpds lab…. It has 21 weeks programs….

Algorithm:

```
start
    declare file pointers
    read two file names
    open two files in read mode
    open third file in write mode
    if any of the file pointer is null
        print error message
    read two files until EOF
        put two files data into third file
    close all files
stop
```

Flowchart:


Start
Read arguments from command line
If
true
false
Print invalid no of arguments
Fp1=fopen(argv[1],”r”)
Fp2=fopen(argv[2],”r”)
true
If fp1or fp2=NULL
Print source file can not be opened
false
Ft=fopen(argv[3],”a”)
true
If ft==NULL
Print target file can not be opened
Fclose(ft)
false
false
While(!feof(fp1))
true
Fclose(fp1)
Fclose(fp2)
Stop
Ch1=fgetc(fp1)
putc(ch1,ft)
false
While(!feof(fp2))
true
Ch1=fgetc(fp1)
Fputc(ch,ft)

Program:

```
 /*   C program to merge the two files into a third file.*/
#include <stdio.h>
#include <conio.h>
#include <process.h>
void main()
{
        FILE *fs1,*fs2,*ft;
        char ch;
        clrscr();
        fs1 = fopen("source.txt","r");
        if(fs1==NULL)
         {
                puts(" 1st Source file doesn't exist.");
                getch();
                exit(0);
         }
                fs2 = fopen("source.txt","r");

         if(fs2==NULL)
         {
                 puts(" 2nd Source file doesn't exist.");
                 getch();
                exit(0);
         }
        ft = fopen("target.txt","w");
        if (ft==NULL)
```

```
    {
        puts("Target file cannot be opened.");
        getch();
        exit(0);
    }

    while((ch=fgetc(fs1))!=EOF)
    {
        fputc(ch,ft);


    }
    while((ch=fgetc(fs2))!=EOF)
    {
        fputc(ch,ft);
    }
    fclose(fs1);
    fclose(fs2);
    fclose(ft);
    getch();
}
```

Result:

Input	Output
cplab.txt	This is cpds lab…. It has 24 weeks programs….
Manual.txt	
Cpds.txt	

Week-15(b)

Define a macro that finds the maximum of two numbers. Write a C program that uses the macro and prints the maximum of two numbers

```
#include<stydio.h>
#define a 10
#define b 20
void main()
{
	if(a<b)
	printf("b is biggest\n");
	else
	printf("a is biggest\n");
}
```

Output:

```
b is biggest
```