

Exercise : 1

a) Convert temperature from centigrade to Fahrenheit.

Description:

To convert Centigrade to Fahrenheit, multiply by 1.8 and add 32 degrees.

Algorithm:

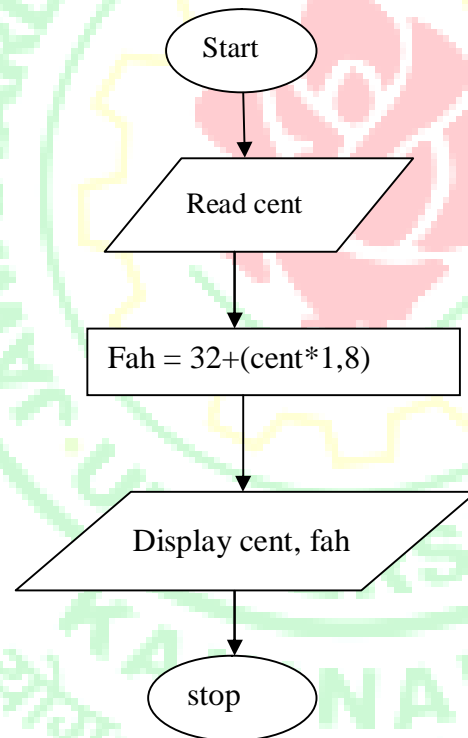
Step 1 : start

Step 2 : read temperature in centigrade

Step 3 : calculate Fahrenheit = $32 + (\text{centigrade} * 1.8)$;

Step 4 : display centigrade and Fahrenheit

Step 5 : stop

Flow chart

Program

```
#include<stdio.h>

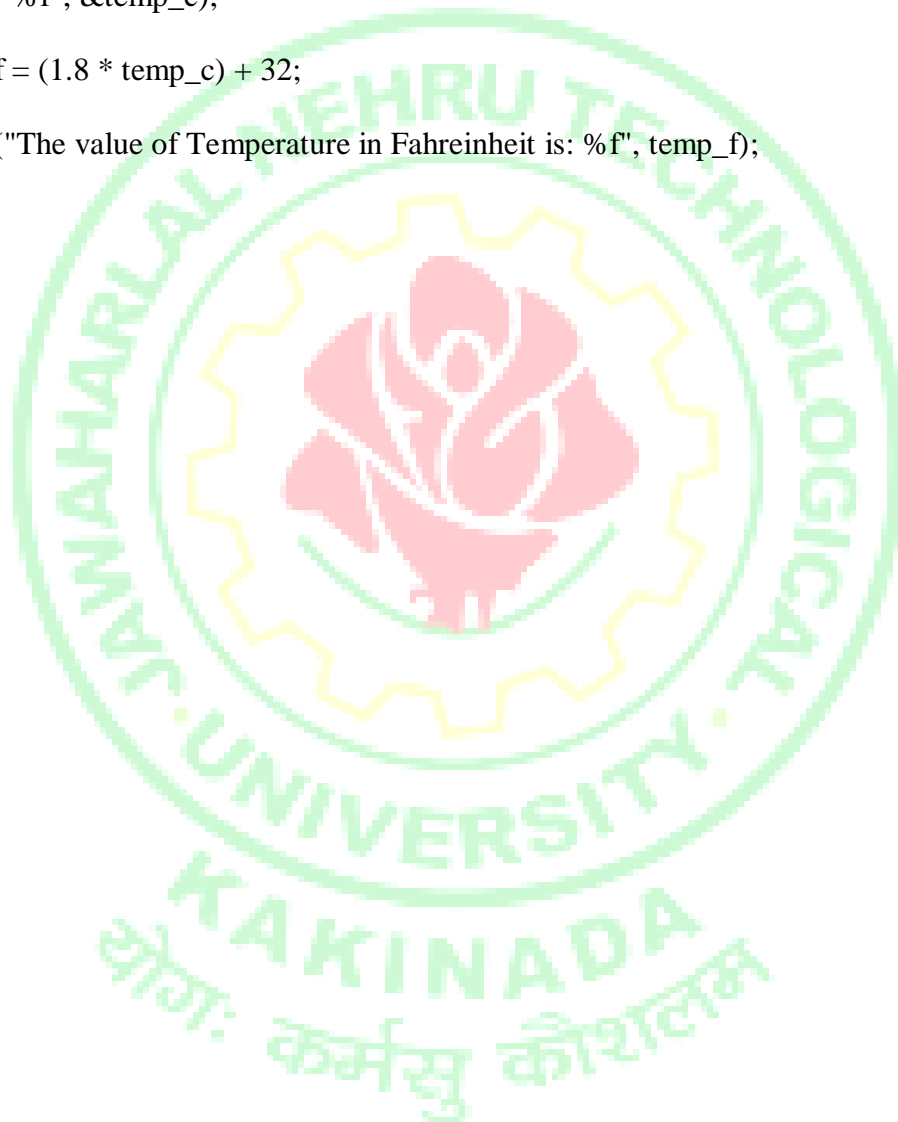
main ()
{
float temp_c, temp_f;

printf ("Enter the value of Temperature in Celcius: ");

scanf ("%f", &temp_c);

temp_f = (1.8 * temp_c) + 32;

printf ("The value of Temperature in Fahrenheit is: %f", temp_f);
}
```



Exercise :1

a) To find Student grading.

Description:

Find the grade of a student by reading marks.

Algorithm

Step 1 : start

Step 2 : read marks

Step 3 : if marks ≥ 80 then grade =A go to step 7

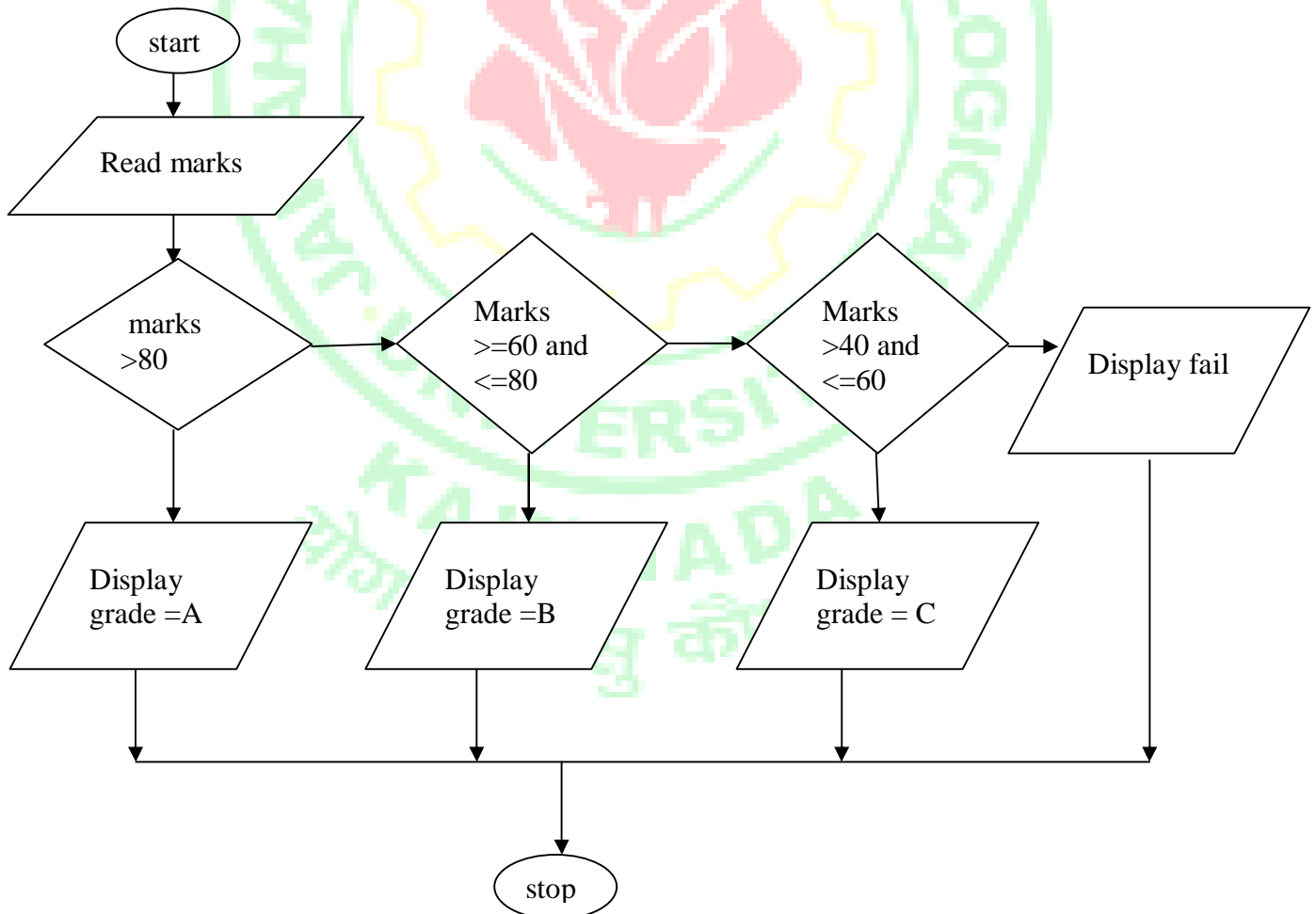
Step 4 : if marks ≥ 60 and marks ≤ 80 then grade = B go to step 7

Step 5 : if marks ≥ 40 and marks ≤ 60 then grade = C go to step 7

Step 6 : display failed

Step 7 : display grade.

Step 8 : stop

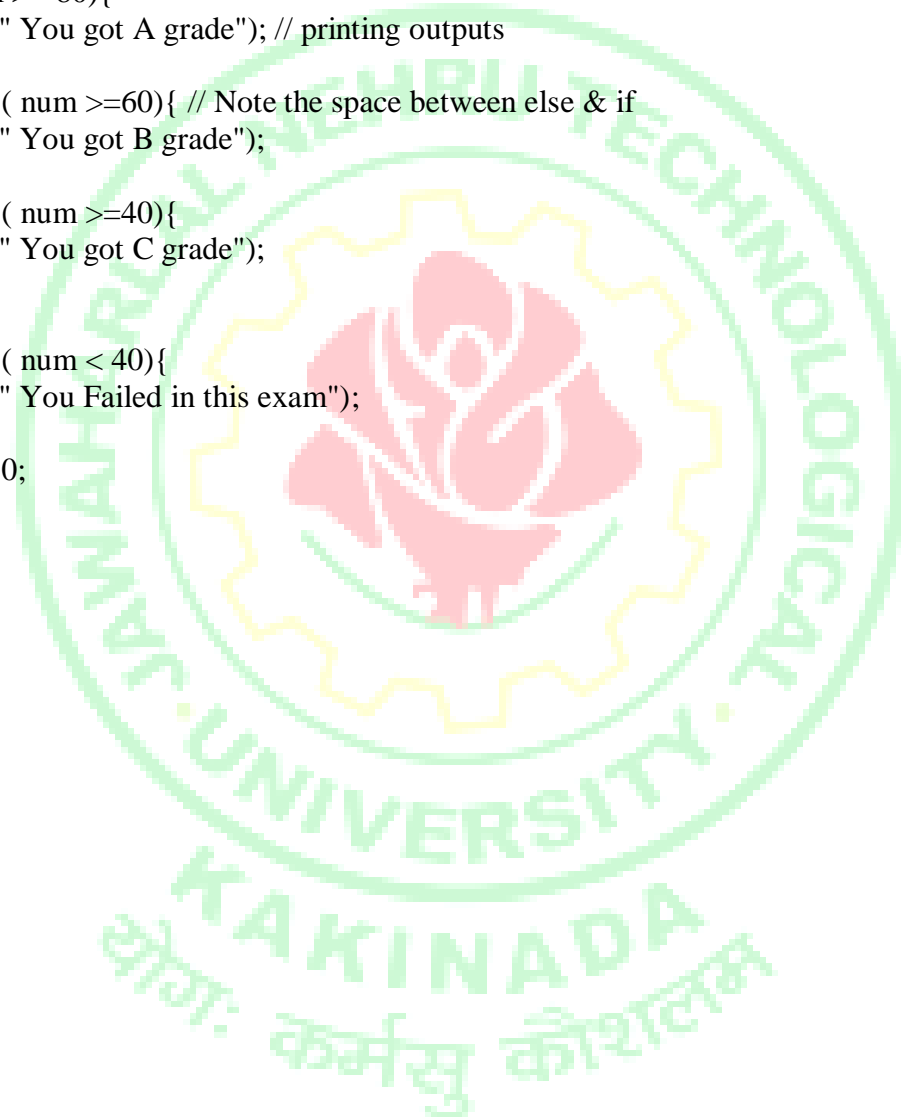
Flow chart

Program

```
#include <stdio.h>
int main(void){
int num;
printf("Enter your mark ");
scanf("%d",&num);
printf(" You entered %d", num); // printing outputs

if(num >= 80){
printf(" You got A grade"); // printing outputs
}
else if ( num >=60){ // Note the space between else & if
printf(" You got B grade");
}
else if ( num >=40){
printf(" You got C grade");
}

else if ( num < 40){
printf(" You Failed in this exam");
}
return 0;
}
```



Exercise : 1

c)Program for income tax for calculation.

Algorithm

Step 1: start

Step 2 : read income

Step 3 : if income <100000 display "no tax" other wise goto step 4

Step 4 : if income < 200000 then tax = income – 100000 *0.1 go to step

Setp 5 : if income > 200000 then tax = tax=((income-100000)*0.2+(income-200000)*0.1)
go step 6

Step 6 : display tax

Step 7 : stop

Program

```
main()
{
int income,tax;
printf("enter the income");
scanf("%d",&income);
{
if(income<100000)
{
printf("no tax");
{
else
if(income<=200000)
{
tax=(income-100000)*0.1;
printf("tax is:%d",tax);
}
}
else
if(income>=200000)
{
tax=((income-100000)*0.2+(income-200000)*0.1);
printf("tax is:%d",tax);
}
}
printf("completed")
}
```

Exercise : 2

To convert the given binary number to 2's complement.

Description:

In this program the given binary number is first convert the numbers 0 to 1 and 1 to 0. And finally add the 1 to the converted number. Then we will get the 2's complement number.

Algorithm: main program

Step 1: Start

Step 2: declare the subprogram "complement(char *a)"

Step 3: initialize the variable i

Step 4: read the binary number

Step 5: perform the loop operation. if it is true then follows. if not goto step 7

i) for(i=0;a[i]!='\0';i++)

ii) if(a[i]!='0' && a[i]!='1') then displayed the number is not valid. enter the correct number.

iii) Exit the loop

Step 6: call sub program 'complemt(a)'

Step 7: stop

Sub program:

Step 1: initialize the variable I,c=0,b[160]

Step 2: l=strlen(a)

Step 3: perform the loop operation. if it is true then follows. if not goto

i)for(i=l-1;i>=0;i--)

ii)if(a[i]=='0') then b[i]='1' else

iii)b[i]='0'

Step 4: for(i=l-1;i>=0;i--) is true

i) if(i==l-1) then

ii) if(b[i]=='0') then b[i]='1' else

iii) b[i]='0',c=1 if not goto step 5

Step 5: if(c==1 && b[i]=='0') is true then

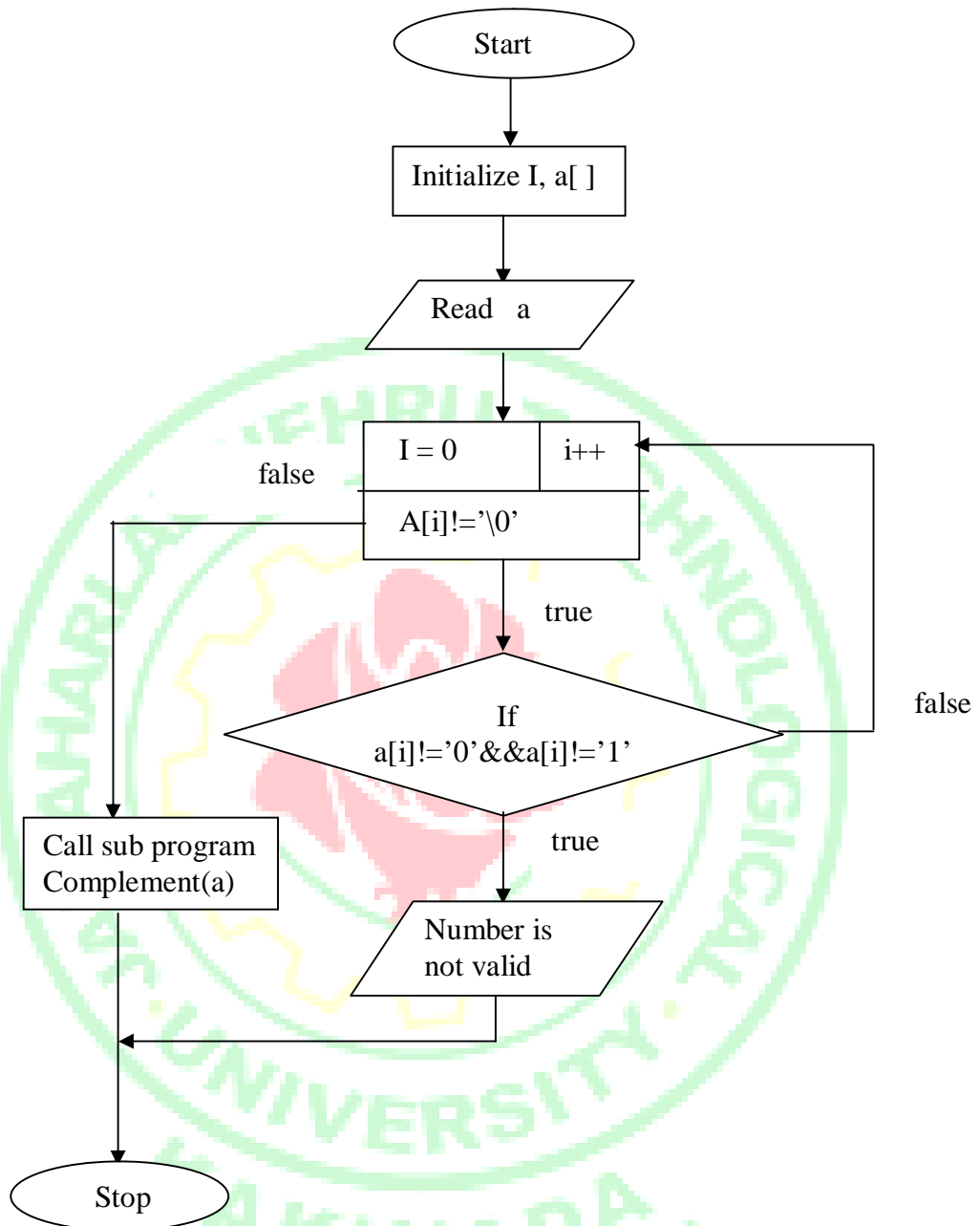
i) b[i]='1', c=0 if not goto Step 6

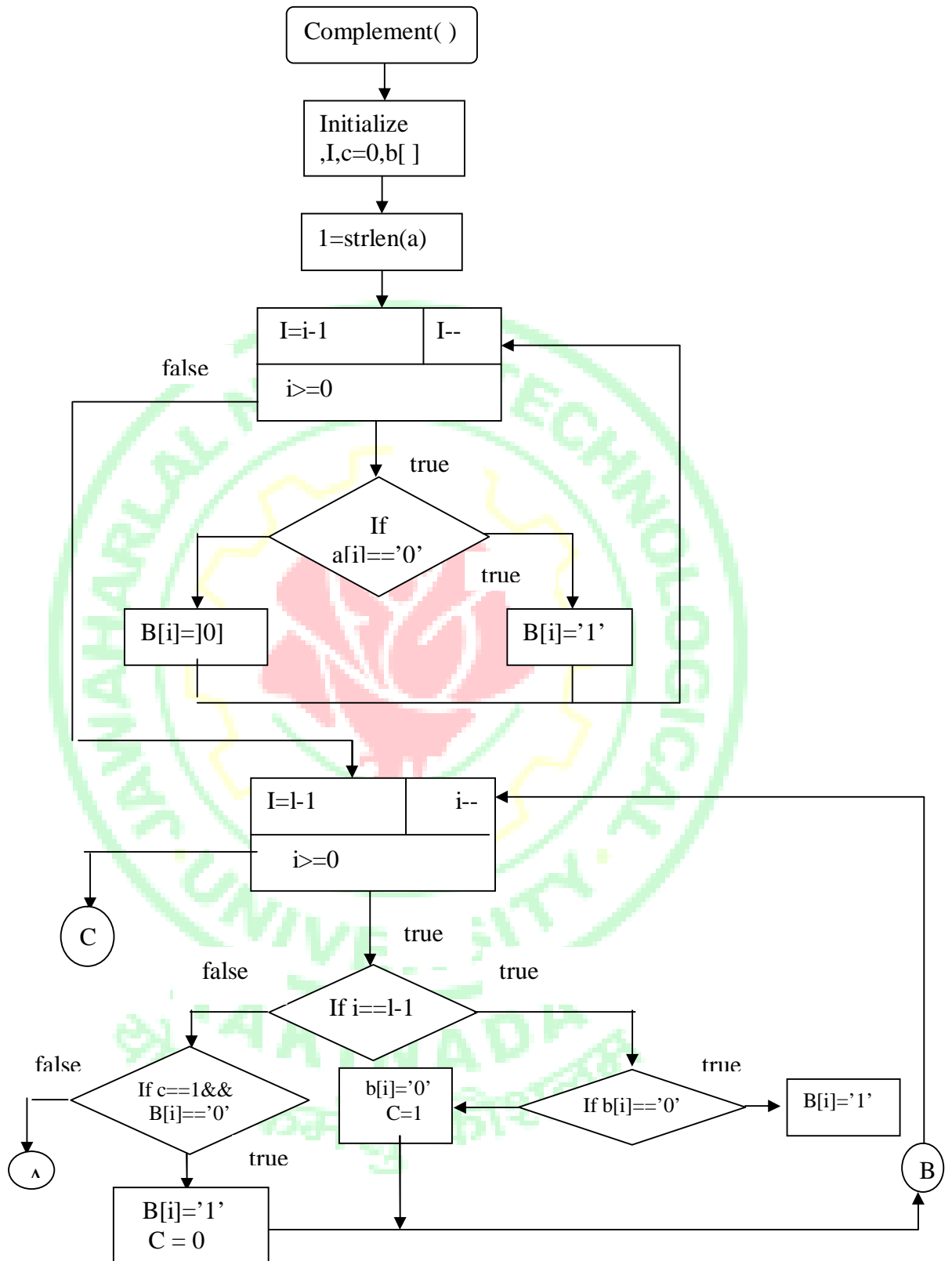
Step 6: if(c==1 && b[i]=='1') then b[i]='0',c=1

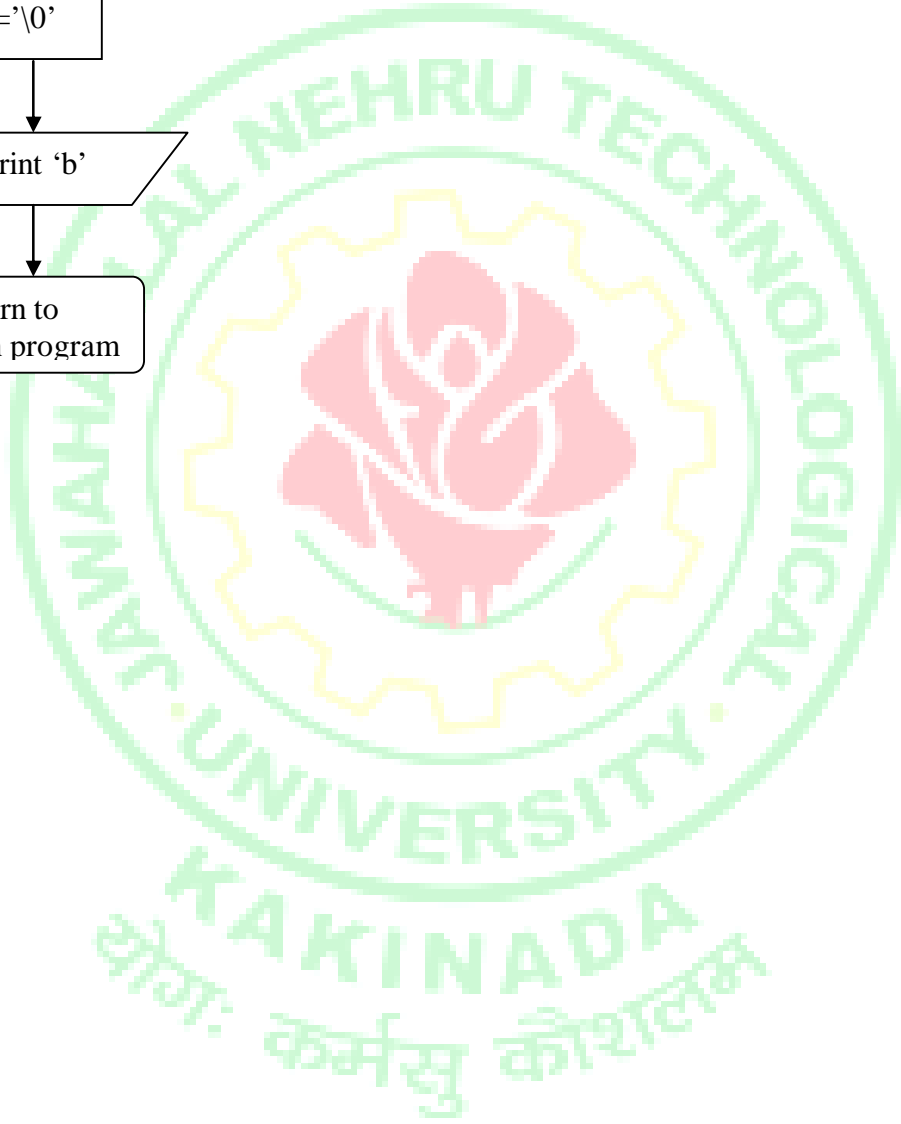
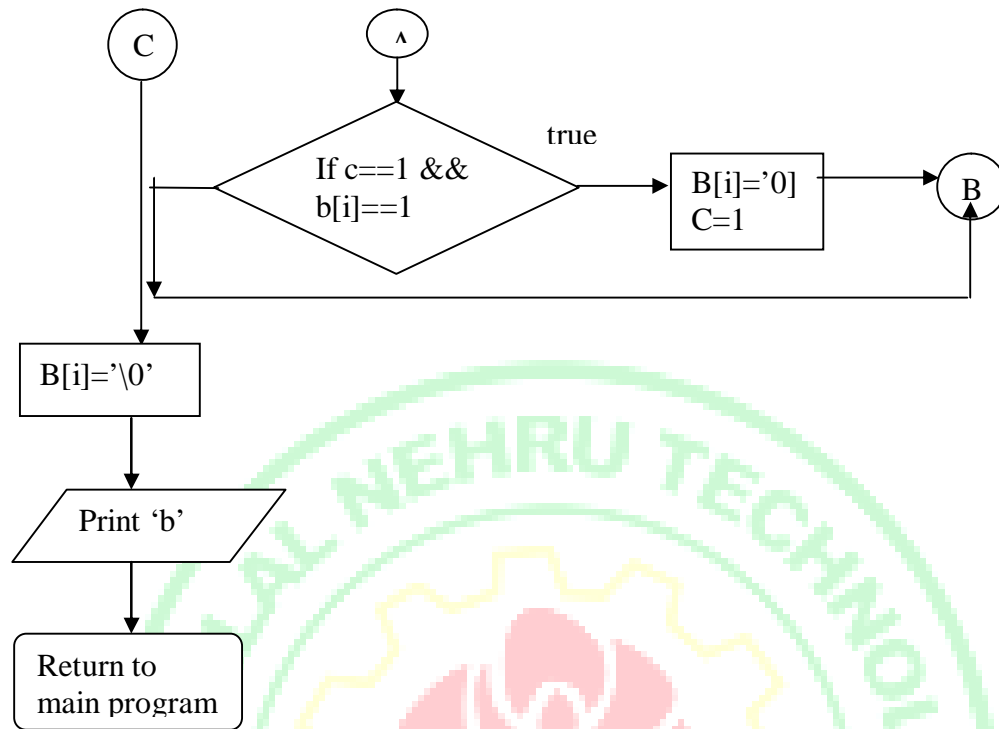
Step 7: displayed b[l]='\0'

Step 8: print b and return to main program

Flow chart:







Program:

```

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

void complement (char *a);
void main()
{
    char a[16];
    int i;
    clrscr();
    printf("Enter the binary number");
    gets(a);
    for(i=0;a[i]!='\0'; i++)
    {
        if (a[i]!='0' && a[i]!='1')
        {
            printf("The number entered is not a binary number. Enter the
            correct number");
            exit(0);
        }
    }
    complement(a);
    getch();
}

void complement (char *a)
{
    int l, i, c=0;
    char b[16];
    l=strlen(a);
    for (i=l-1; i>=0; i--)
    {
        if (a[i]=='0')
            b[i]='1';
        else
            b[i]='0';
    }
    for(i=l-1; i>=0; i--)
    {
        if(i==l-1)
        {
            if (b[i]=='0')
                b[i]='1';
            else
            {
                b[i]='0';
                c=1;
            }
        }
        else

```

```
{
    if(c==1 && b[i]=='0')
    {
        b[i]='1';
        c=0;
    }
    else if (c==1 && b[i]=='1')
    {
        b[i]='0';
        c=1;
    }
}
}
b[1]='\0';
printf("The 2's complement is %s", b);
}
```

Output:

1.Enter the binary number101010
The 2's complement is 010110

Enter the binary number11111

The 2's complement is 00001

Enter the binary number2222

The number entered is not a binary number. Enter the correct number

Conclusion: the program is error free

VIVA QUESATIONS:

1) Expand ASCII ?

Ans: American standard code for information interchange

2)What is binary number ?

Ans: The number which contains only 0 and 1 is called binary number.

3) Define 2's complement ?

Ans: The given binary number is first convert the numbers 0 to 1 and 1 to 0. And finally add the 1 to the converted number. Then we will get the 2's complement number.

Exercise 3

a) To find the sum of individual digits of a given number

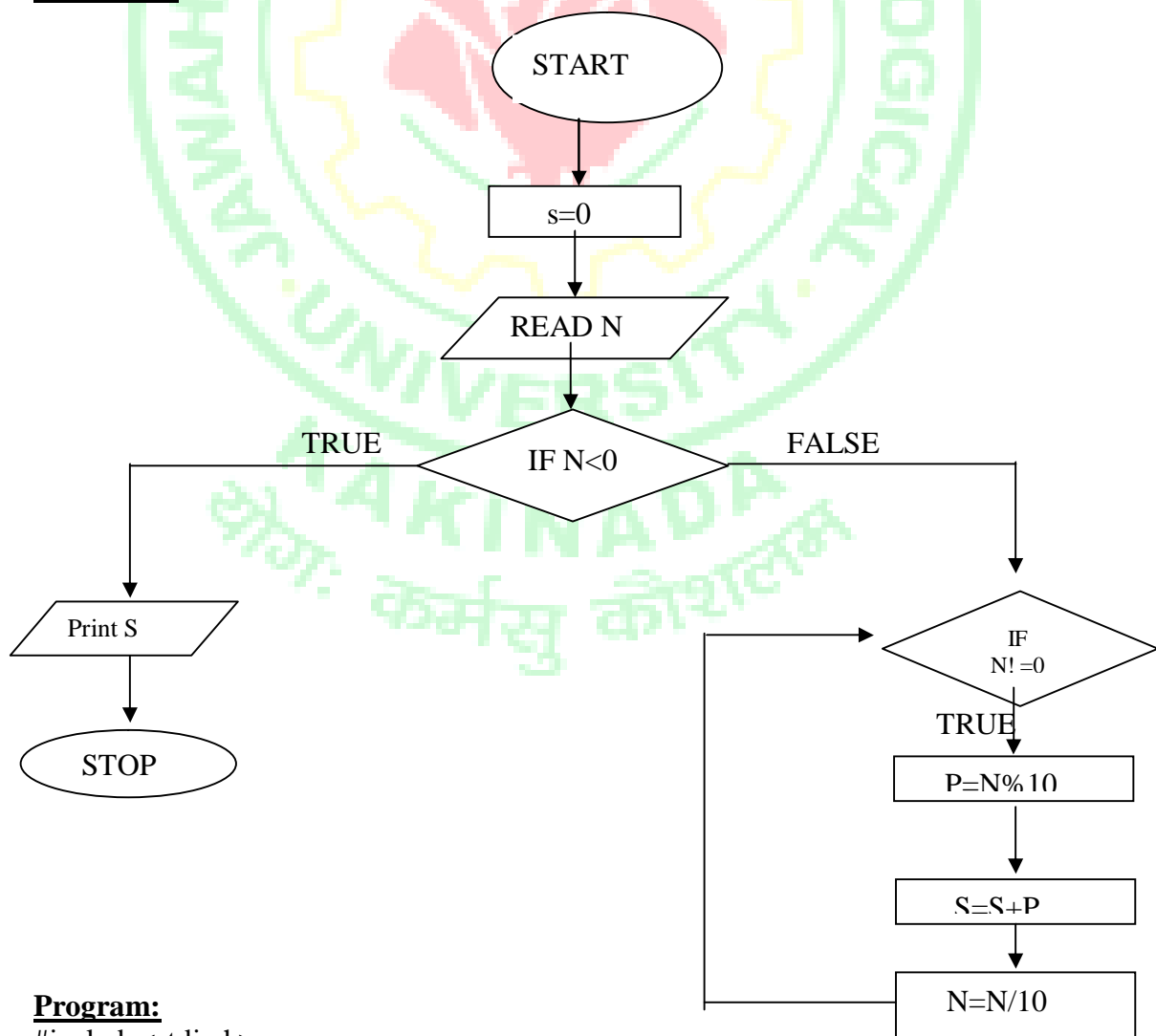
Description:

Sum of the individual digits means adding all the digits of a number

Ex: 123 sum of digits is $1+2+3=6$

Algorithm:

- Step 1: start
 Step 2: read n
 Step 3: initialize the $s=0$
 Step 4: if $n<0$ goto Step 7
 Step 5: if $n!=0$ goto Step 6 else goto step 7
 Step 6: store $n\%10$ value in p
 Add p value to s
 Assign $n/10$ value to n
 Goto Step 5
 Step 7: print s
 Step 8: stop

Flowchart:**Program:**

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

```
main()
{
    int n,s,p;
    clrscr();
    printf("enter the vaue for n:\n");
    scanf("%d",&n);
    s=0;
    if(n<0)
    printf("The given number is not valid");
    else
    {
        while(n!=0) /* check the given value =0 or not */
        {
            p=n%10;
            n=n/10;
            s=s+p;
        }
        printf("sum of individual digits is %d",s);
    }
    getch();
}
```

Output:

1. Enter the value for n: 333
Sum of individual digits is 9
2. Enter the value for n: 4733
Sum of individual digits is 17
3. Enter the value for n: -111
The given number is not valid

Conclusion : The program is error free

VIVA QUESATIONS:

1) What is the mean of sum of the individual digits?

Ans: Sum of the individual digits means adding each digit in a number

2) What is positive integer?

Ans: if the integer value is grater than zero then it is called positive integer

3) Define preprocessor ?

Ans: Before compiling a process called preprocessing is done on the source code by a program called the preprocessor.

Exercise: 3

b) To print the Fibonacci series for 1 to n value

Description

A fibonacci series is defined as follows

The first term in the sequence is 0

The second term in the sequence is 1

The sub sequent terms are found by adding the preceding two terms in the sequence

Formula: let t_1, t_2, \dots, t_n be terms in fibonacci sequence

$t_1=0, t_2=1$

$t_n=t_{n-2}+t_{n-1}$where $n>2$

algorithm:

Step 1: start

Step 2: initialize the $a=0, b=1$

Step 3: read n

Step 4: if $n== 1$ print a go to step 7. else goto step 5

Step 5: if $n== 2$ print a, b go to step 7 else print a,b

Step 6: initialize $i=3$

i) if $i \leq n$ do as follows. If not goto step 7

$c=a+b$

print c

$a=b$

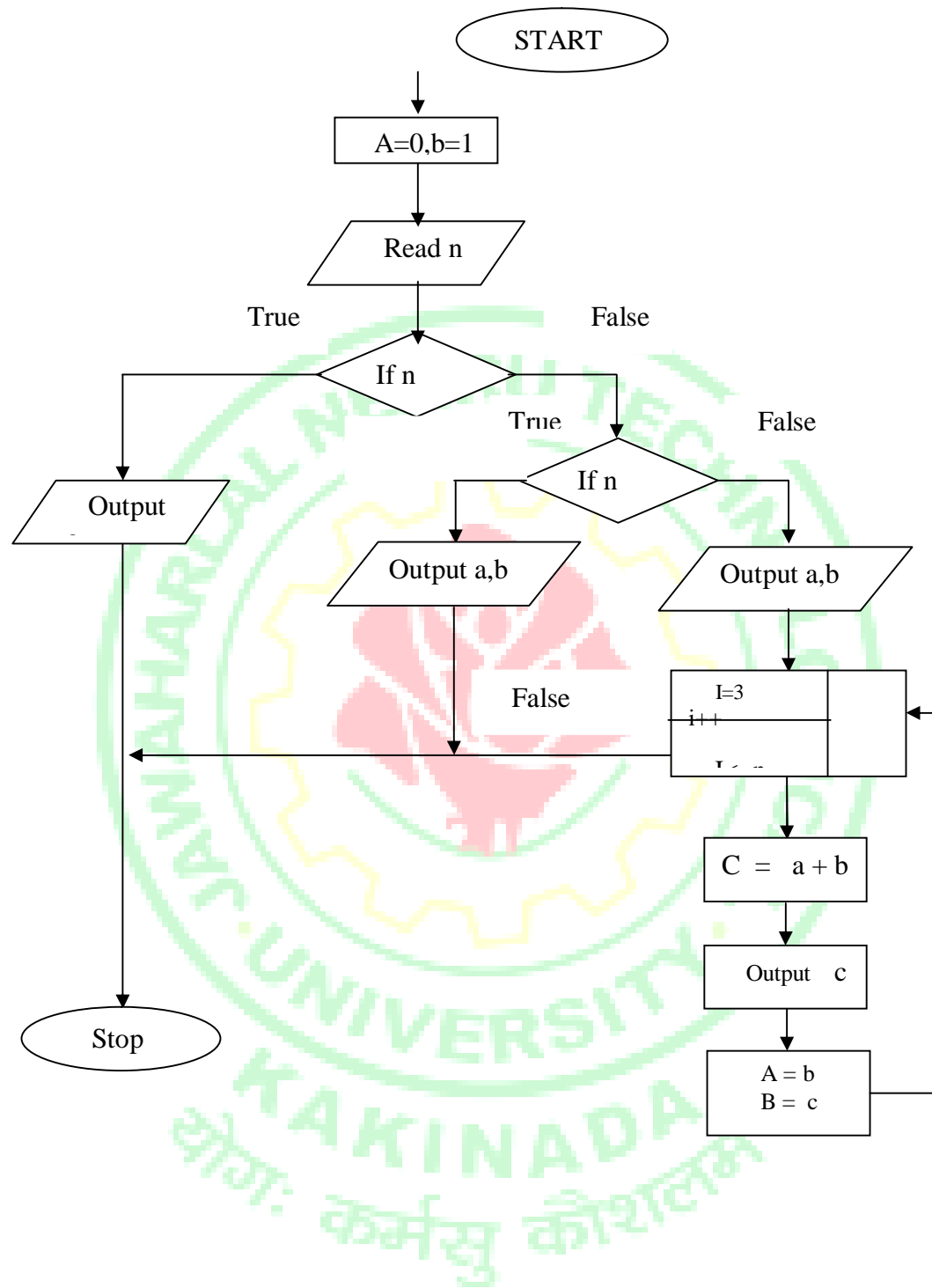
$b=c$

increment I value

goto step 6(i)

Step 7: stop

Flowchart:



Program:

```

#include<stdio.h>
void main()
{
    int a,b,c,n,i;
    clrscr();
    printf("enter n value");
    scanf("%d",&n);
    a=0; b=1;
    if(n==1)
    printf("%d",a);
    else if(n==2)
    printf("%d%d",a,b);
    else
    {
        printf("%d%d",a,b);
        //LOOP WILL RUN FOR 2 TIME LESS IN SERIES AS THESE WAS
        PRINTED IN ADVANCE
        for(i=3;i<=n;i++)
        {
            c=a+b;
            printf("%d",c);
            a=b;
            b=c;
        }
        getch();
    }
}

```

Output:

1. Enter n value : 5
0 1 1 2 3
2. Enter n value : 7
0 1 1 2 3 5 8
3. Enter n value : -6
0 1

Conclusion : The program is error free

VIVA QUESATIONS:**1) What is Fibonacci series ?**

Ans: A fibonacci series is defined as follows

The first term in the sequence is 0

The second term in the sequence is 1

The sub sequent terms 1 found by adding the preceding two terms in the sequence

Formulae : let t_1, t_2, \dots, t_n be terms in fibinacci sequence

$t_1=0, t_2=1$

$t_n=t_{n-2}+t_{n-1} \dots$ where $n>2$

2) What are the various types of unconditional statements?

Ans: goto,Break and continue

3)What are the various types of conditional statements?

Ans: if , if else ,switch statements

Exercise : 3

- c) 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

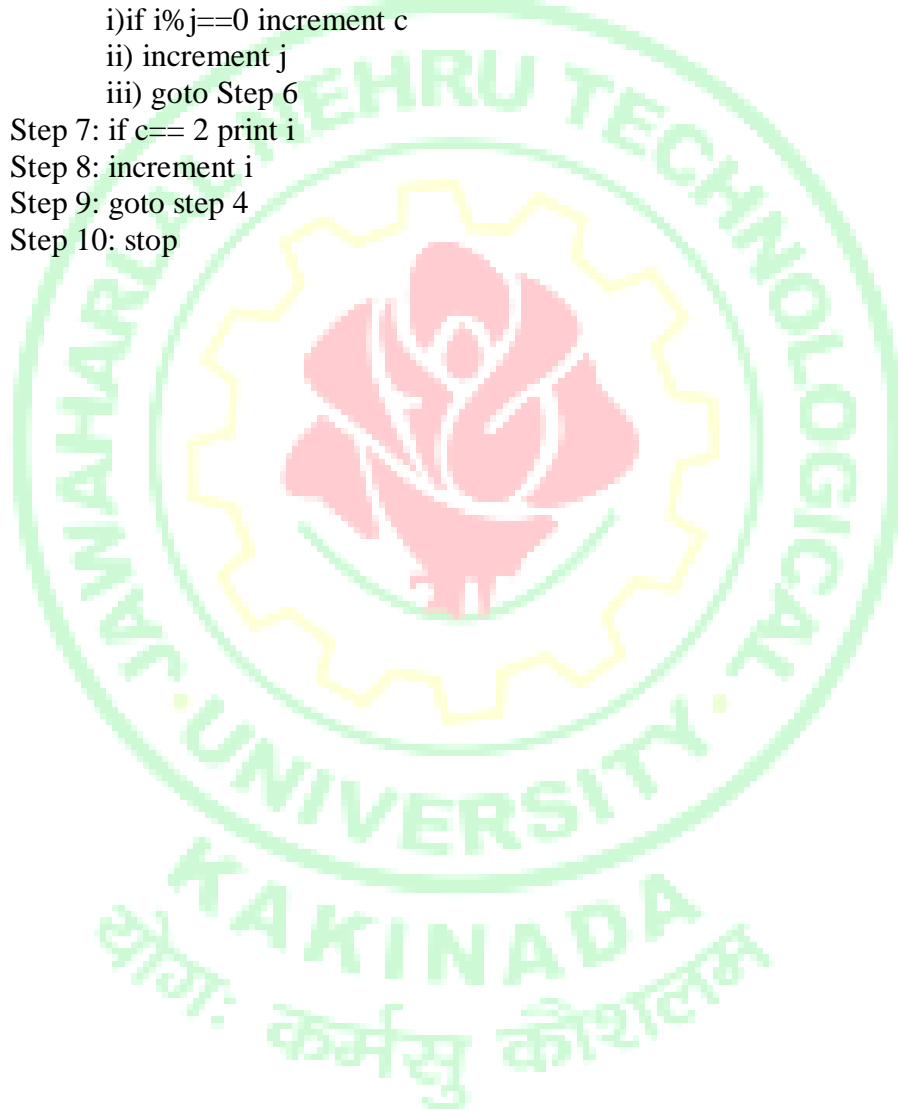
 iii) goto Step 6

Step 7: if c== 2 print i

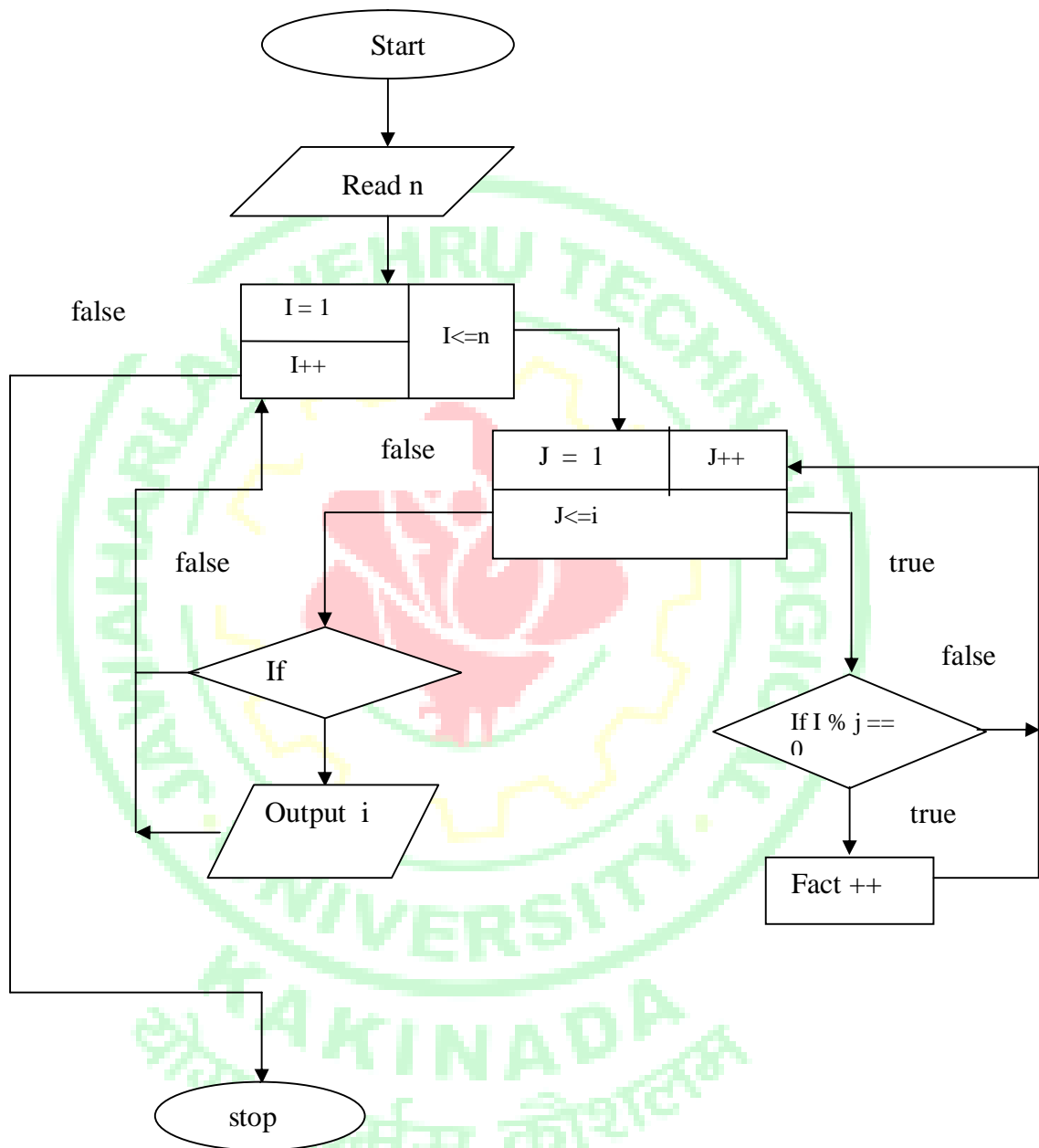
Step 8: increment i

Step 9: goto step 4

Step 10: stop



Flow chart:



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

Enter the number : 10

2 3 5 7

Enter the number : 12

2 3 5 7,11

Conclusion : The program is error free

VIVA QUESATIONS:**1) What is prime number ?**

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

2)What is an algorithm?

Ans : A step by step procedure is called algorithm

3)What is flow chart?

Ans: A pictorial representation an algorithm is called a flow chart

4)What is program?

Ans : A collection of statements is called

Exercise : 3

d) To check a given integer is Fibonacci number or not.

Description :

Find the given number is Fibonacci or not. It means if the given number is present in the Fibonacci series it satisfies the condition otherwise it fails.

Algorithm

Step 1: start
 Step 2: initialize a=0 , b=1
 Step 3 : read n
 Step 4 : compute c=a+b
 Step 5 : if c<=n go to step 6 otherwise goto step 12
 Step 6 : compute c = a+b
 Step 7 : a=b
 Step 8 : b = c goto step 5
 Step 9 : if (c==n) goto step 10 otherwise goto step 11
 Step 10 : display “ given number is Fibonacci “ n
 Step 11 : display “ given number is not Fibonacci “ n
 Step 12 : stop

Program

```
main()
{
  int a=0,b=1,n,c;
  printf("\nEnter no to find : ");
  scanf("%d", &n);
  c =a+b;
  while (c<=n)
  {
    c=a+b;
    a=b;
    b=c;
  }
  if(c==n)
    printf("%d is Fibonacci ", n);
  else
    printf("\n%d is not Fibonacci ", n);
}
```

Exercise : 4

a) To calculate the sum. $\text{Sum} = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8! - x^{10}/10!$

Algorithm: main program:

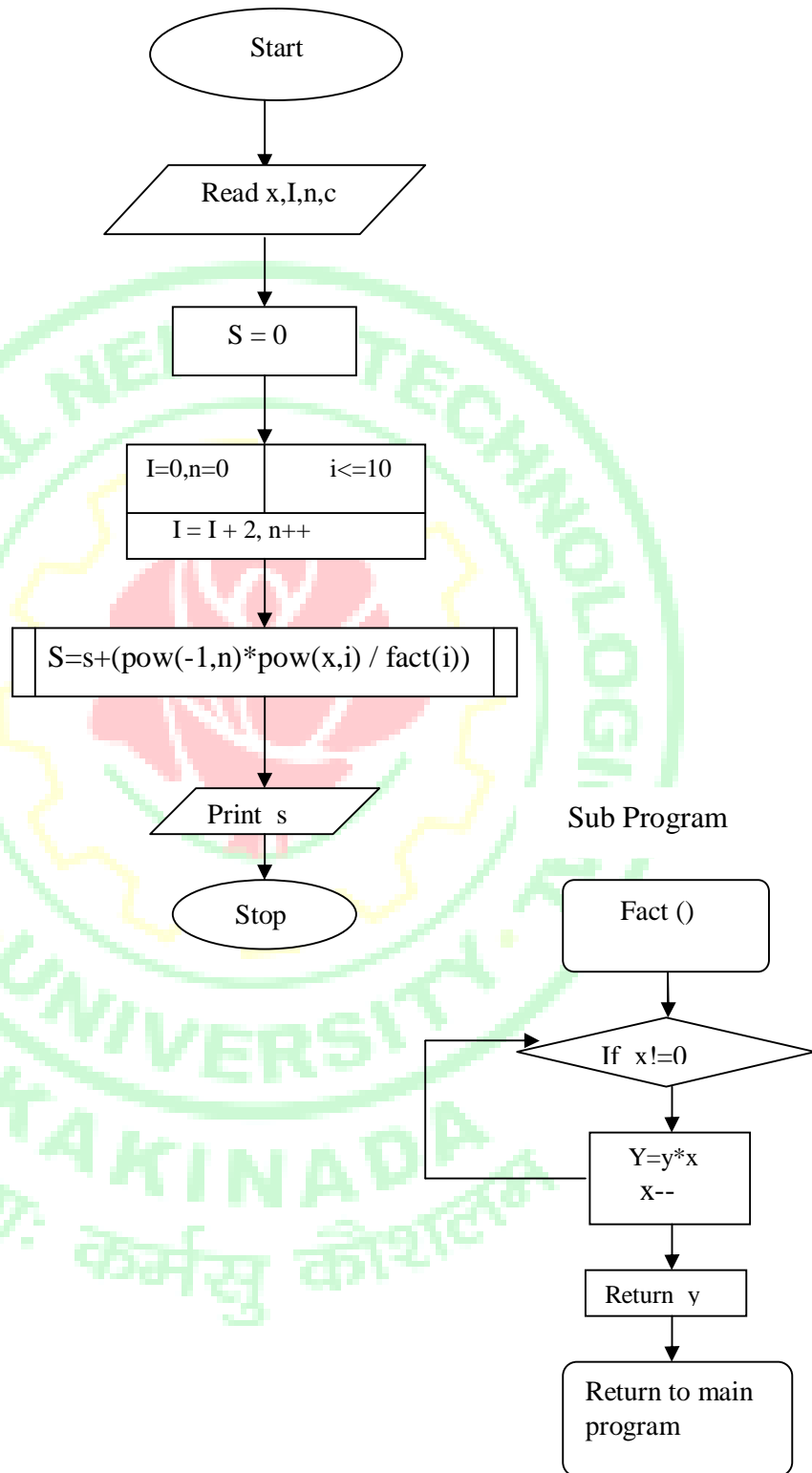
Step 1: start
Step 2: declare x,i,n,s=0,c
Step 3: read x value
Step 4: for i=0 , n=0; i<=10; i=i+2, n++ goto step 5
Step 5: s=s+(pow(-1,n)*pow(x,i)/fact(i))
Step 6: print s value
Step 7: stop

Sub program:

Step 1: while x!=0 goto Step 2
Step 2: y=y+x; x—
Step 3: return y
Step 4: return to main program



Flowchart:



Program:

```

#include<stdio.h>
#include<math.h>
long fact(int);
void main()
{
    int x,i,n;
    float s=0,c;
    clrscr();
    printf("\n enter the value of x\t");
    scanf("%d",&x);
    /*perform the looping operation*/
    for(i=0,n=0;i<=10;i=i+2,n++)
    s=s+(pow(-1,n)*pow(x,i)/fact(i));
    printf("\n the result is %f",s);
    getch();
}
/* calling sub program*/
long fact(int x)
{
    long int y=1;
    while(x!=0)
    {
        y=y*x;
        x--;
    }
    return y;
}

```

Output:

1. Enter the value of x : 1
The result is 0.540302
- 2 Enter the value of x: 2
The result is -0.416155

Conclusion: The program is error free

VIVA QUESATIONS:

1) What is function ?

Ans: A function is a sub program it returns a value.

2) What is procedure ?

Ans: A procedure is a sub program it does not returns a value

3) What are the basic data types in C ?

Ans: int, char, float, double

4) How to define preprocessor ?

Ans: By using the # symbol Ex: #include<stdio.h>

a)

Exercise : 4

b) To find the roots of the quadratic equation

Description:

Nature of roots of quadratic equation can be known from the quadrant $\Delta = b^2 - 4ac$

If $b^2 - 4ac > 0$ then roots are real and unequal

If $b^2 - 4ac = 0$ then roots are real and equal

If $b^2 - 4ac < 0$ then roots are imaginary

Algorithm:

Step 1: start

Step 2: read the a,b,c value

Step 3: if $(b^2 - 4ac) > 0$ then

Root 1 = $(-b + \text{pow}((b^2 - 4ac), 0.5)) / 2a$

Root 2 = $(-b - \text{pow}((b^2 - 4ac), 0.5)) / 2a$

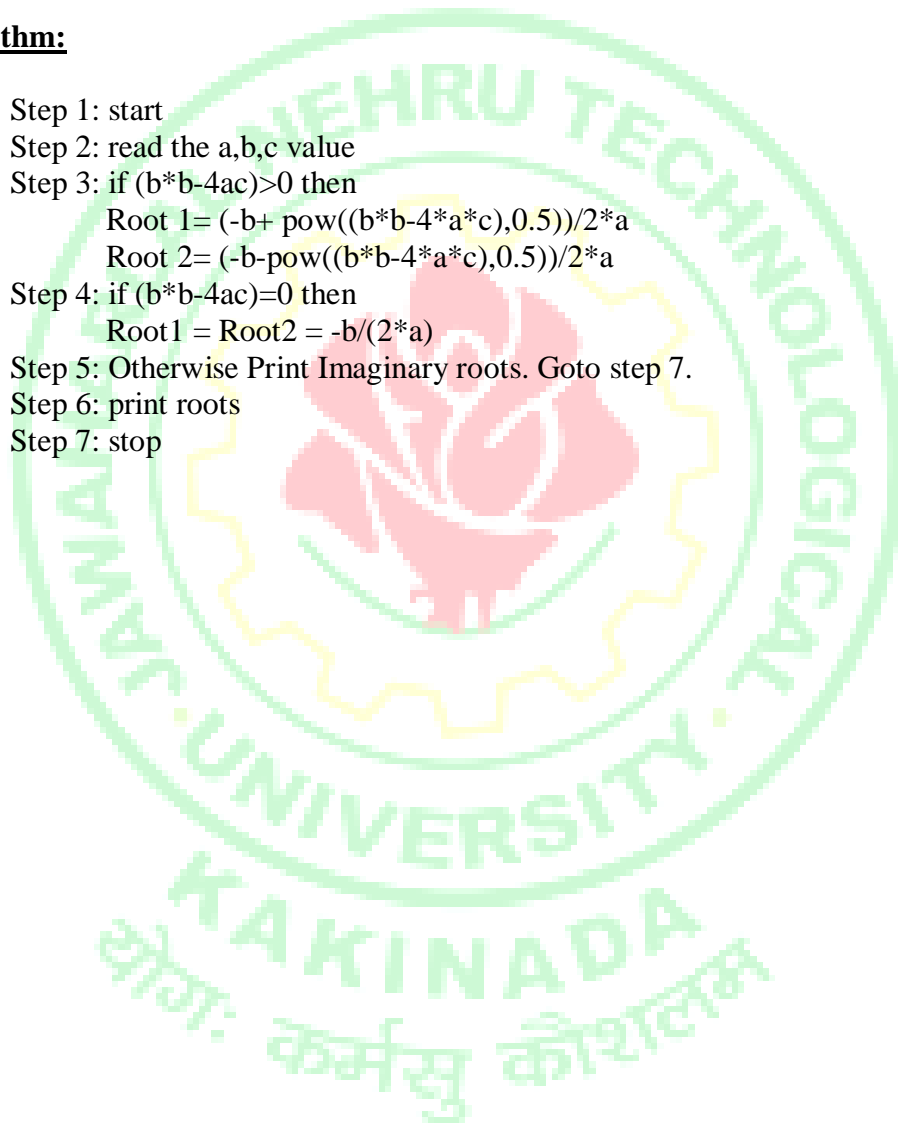
Step 4: if $(b^2 - 4ac) = 0$ then

Root1 = Root2 = $-b / (2a)$

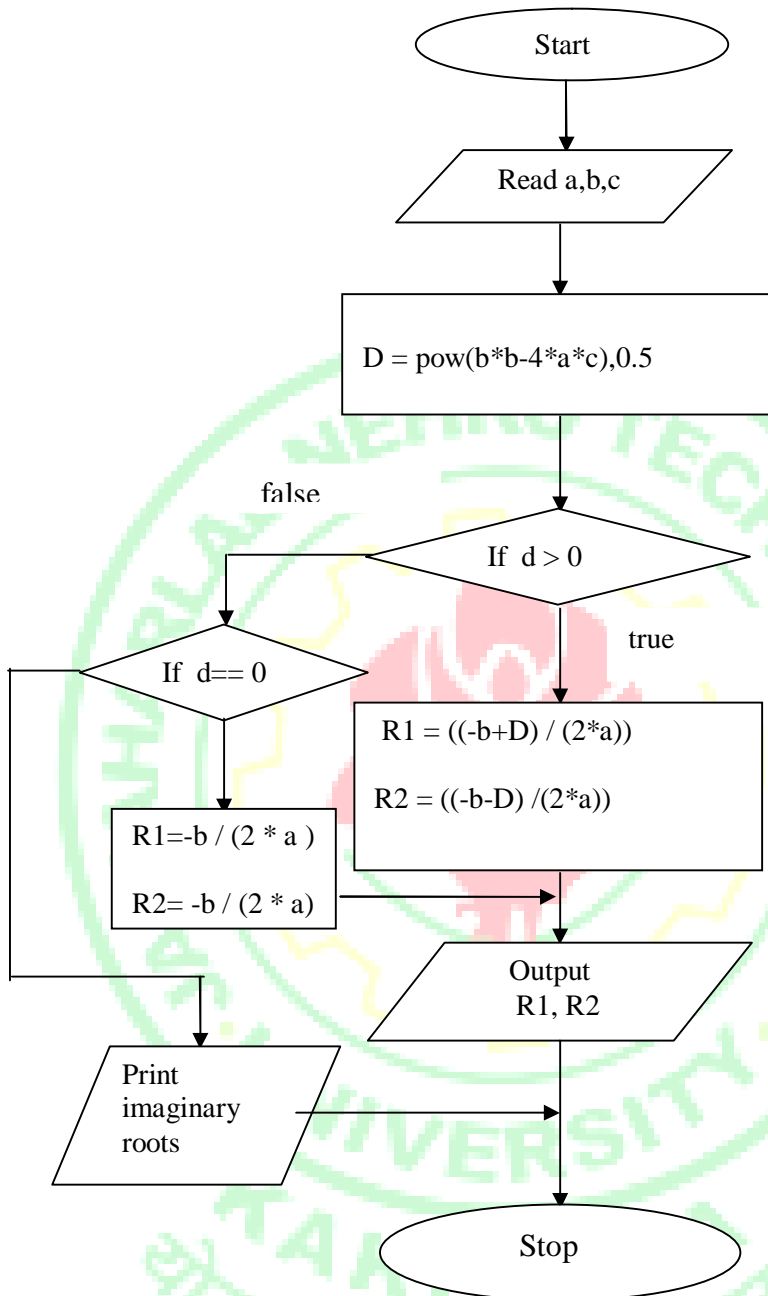
Step 5: Otherwise Print Imaginary roots. Goto step 7.

Step 6: print roots

Step 7: stop



Flowchart:



Program:

```

#include<stdio.h>
#include<math.h>
void main()
{
float a,b,c,r1,r2,d;
clrscr();
printf("Enter the values for equation:");
scanf("%f%f%f",&a,&b,&c);
/* check the condition */
if(a==0)
printf("Enter value should not be zero ");
else
{
d=b*b-4*a*c;
/* check the condition */
if(d>0)
{
r1=(-b+sqrt(d)/(2*a));
r2=(-b-sqrt(d)/(2*a));
printf("roots are real and unequal\n");
printf("%f\n%f\n",r1,r2);
}
else
if(d==0)
{
r1=-b/(2*a);
r2=-b/(2*a);
printf("roots are real and equal\n");
printf("root=%f\n",r1);
printf("root=%f\n",r2);
}
else
printf("roots are imaginary");
}
getch();
}

```

Output:

1. Enter the values for equation: 1, 6, 9
Roots are real and equal
Root= -3.0000
Root= -3.0000
2. Enter the values for equation: 2, 7, 6
Roots are real and unequal
Root= -6.75
Root= -7.25
3. Enter the values for equation: 1, 2, 3
Roots are imaginary

Conclusion: The program is error free

VIVA QUESATIONS:

1) What are various types of loop statements?

Ans : While, do- while, for loop statements

4) What is the difference between while and do-while statements?

Ans: In while the condition will be checked first and then enter into a loop.
But in do- while the statements will be executed first and then finally check the
Condition.

3) How to find the roots of quadratic equations ?

Ans: Nature of roots of quadratic equation can be known from the discriminant

$$\Delta = b^2 - 4ac$$

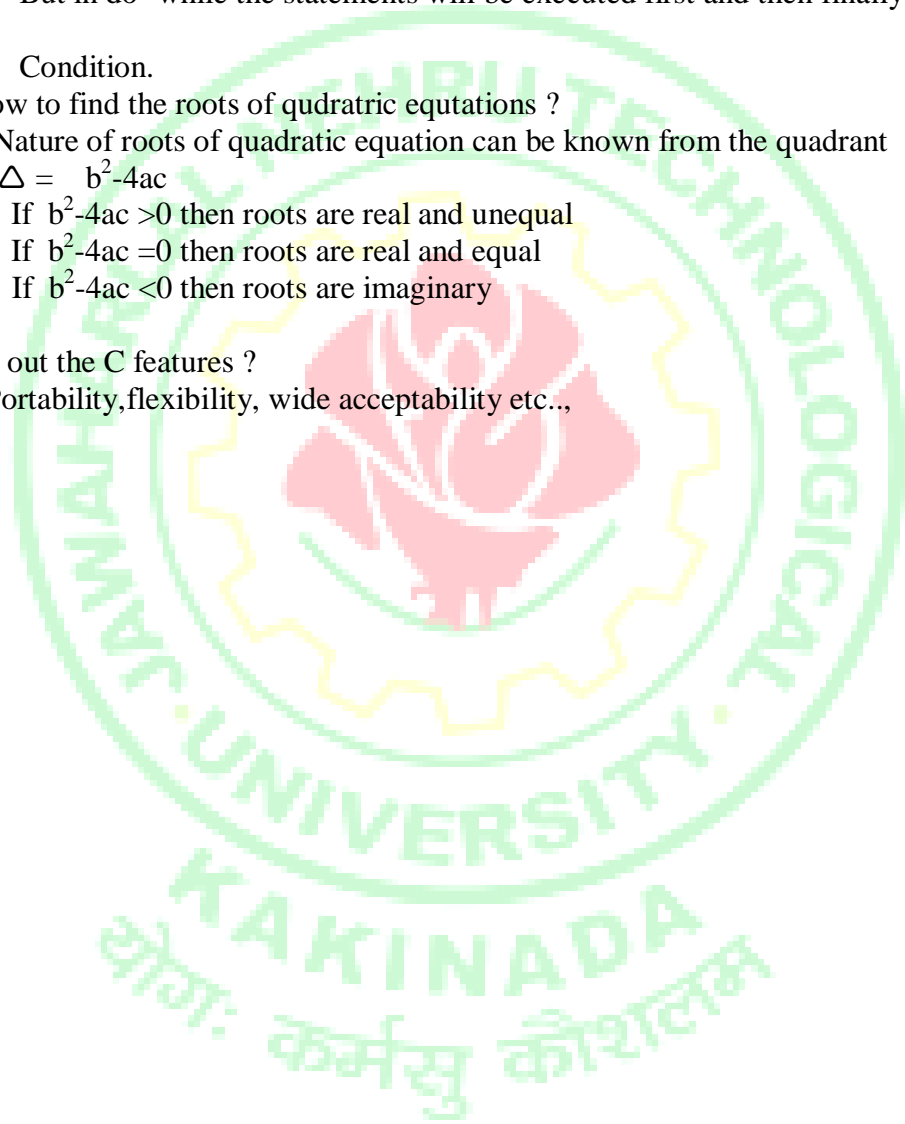
If $b^2 - 4ac > 0$ then roots are real and unequal

If $b^2 - 4ac = 0$ then roots are real and equal

If $b^2 - 4ac < 0$ then roots are imaginary

4) List out the C features ?

Ans: Portability, flexibility, wide acceptability etc.,



Exercise : 5

- a) The total distance travelled by vehicle in 't' seconds is given by distance = $ut + \frac{1}{2}at^2$ where 'u' and 'a' are the initial velocity (m/sec.) and acceleration (m/sec²). Write C program to find the distance travelled at regular intervals of time given the values of 'u' and 'a'. The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of 'u' and 'a'.

Description:

The total distance travelled by vehicle in 't' seconds is given by distance = $ut + \frac{1}{2}at^2$ where 'u' and 'a' are the initial velocity (m/sec.) and acceleration (m/sec²).

Algorithm:

Step 1: Start

Step 2 : Read t ,dt

Step 3: Set i to 1

Step 4: Set k to dt

Step 5: Read u,a

Step 6: set s to $u*k + 0.5*d*k*k$

Step 7: Write s

Step 8: If($k \leq t$) and $i=1$ then

 Begin

 Step 8.1 go to step 6

And

 Else

 Begin

 Step 8.2 : read

 Step 8.3 : if($j=0$) then

 Begin

 Step 8.3.1: Set I to 0

 End

 Else

 Begin

 Step 8.3.2: Set I to 1

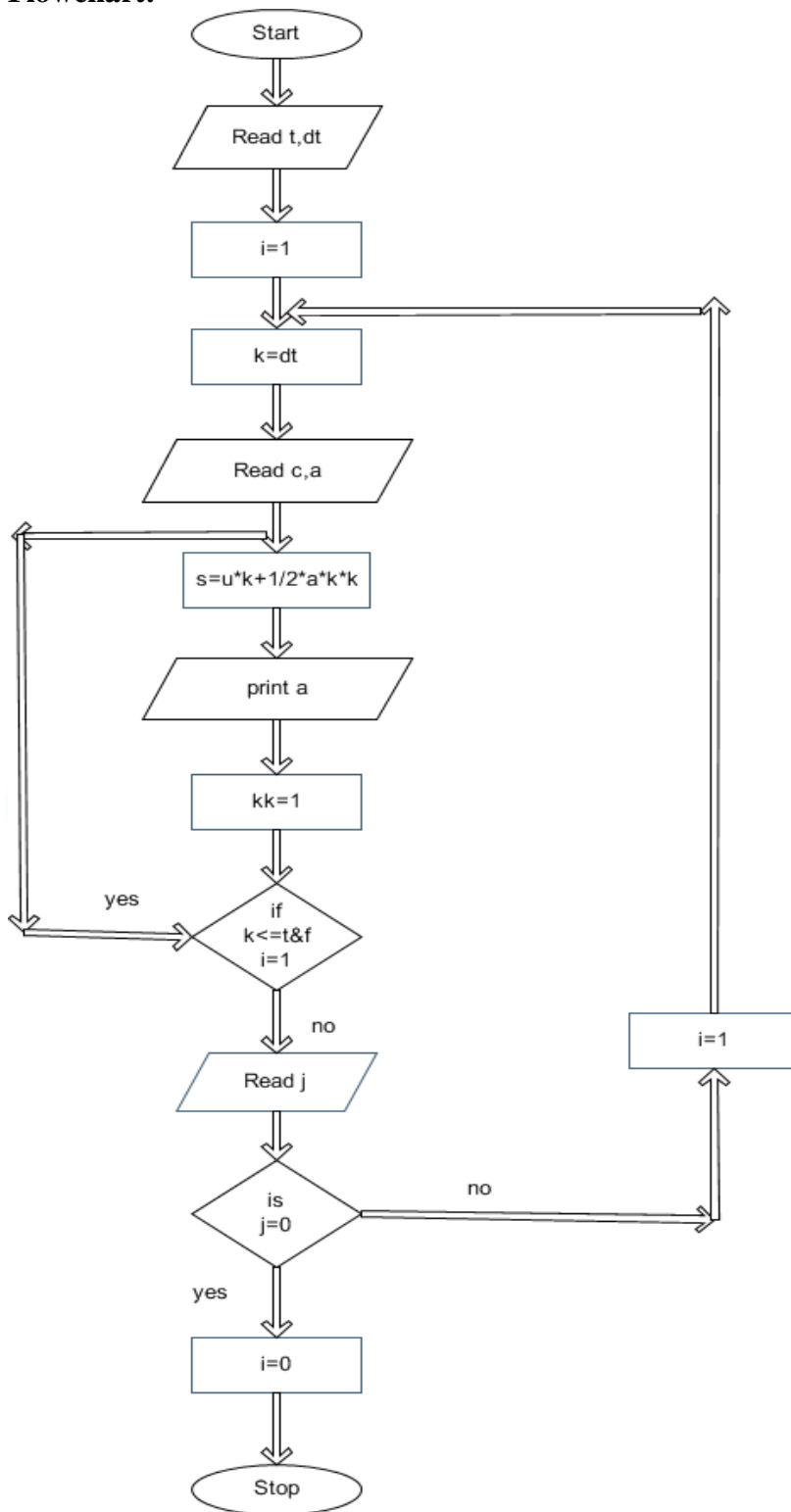
 Step 8.3.3: go to step 4

 End

Step 9: Stop

Step 10: End

Flowchart:



Program:

```

#include<stdio.h>
main()
{
    int a,u,t,t1,t2,i;
    float s;
    clrscr();
    printf("ENTER THE VALUES OF a,u,t,t1,t2:");
    scanf("%d%d%d%d%d",&a,&u,&t,&t1,&t2);
    for(i=t1;i<=t2;i=i+t) // performing the looping operation for time
intervals
    {
        s=(u*i)+(0.5*a*i*i); // calculate the total distance
        printf("\n\nthe distance travelled in %d seconds is %f ",i,s);
    }
    getch();
}

```

Input/Output:

1. ENTER THE VALUES OF a,u,t,t1,t2:1

2

3

1

5

the distance travelled in 1 seconds is 2.500000

the distance travelled in 4 seconds is 16.000000

2. ENTER THE VALUES OF a,u,t,t1,t2:0

1

2

3

4

the distance travelled in 3 seconds is 3.000000

conclusion: The program is error free

VIVA QUESATIONS:

1) How many types of arrays are there ?

Ans: Three types. They are one dimensional ,two dimensional and multi dimensional arrys

Exercise : 5

- b) Two integer operands and one operator form user, performs the operation and then prints the result.

(Consider the operators +,-,*,/, % and use Switch Statement)

Description:

To take the two integer operands and one operator from user to perform the some arithmetic operations by using the following operators like +,-,*,/, %

Ex: 2+3=5

Algorithm:

Step 1: Start

Step 2: Read the values of a,b and operator

Step 3: if the operator is '+' then

R=a+b

Go to step 8

Break

Step 4: Else if the operator is '-' then

R=a-b

Go to step 8

Step 5: Else if the operator is '*' then

R=a*b

Go to step 8

Step 6: Else if the operator is '/' then

R=a/b

Go to step 8

Step 7: Else if the operator is '%' then

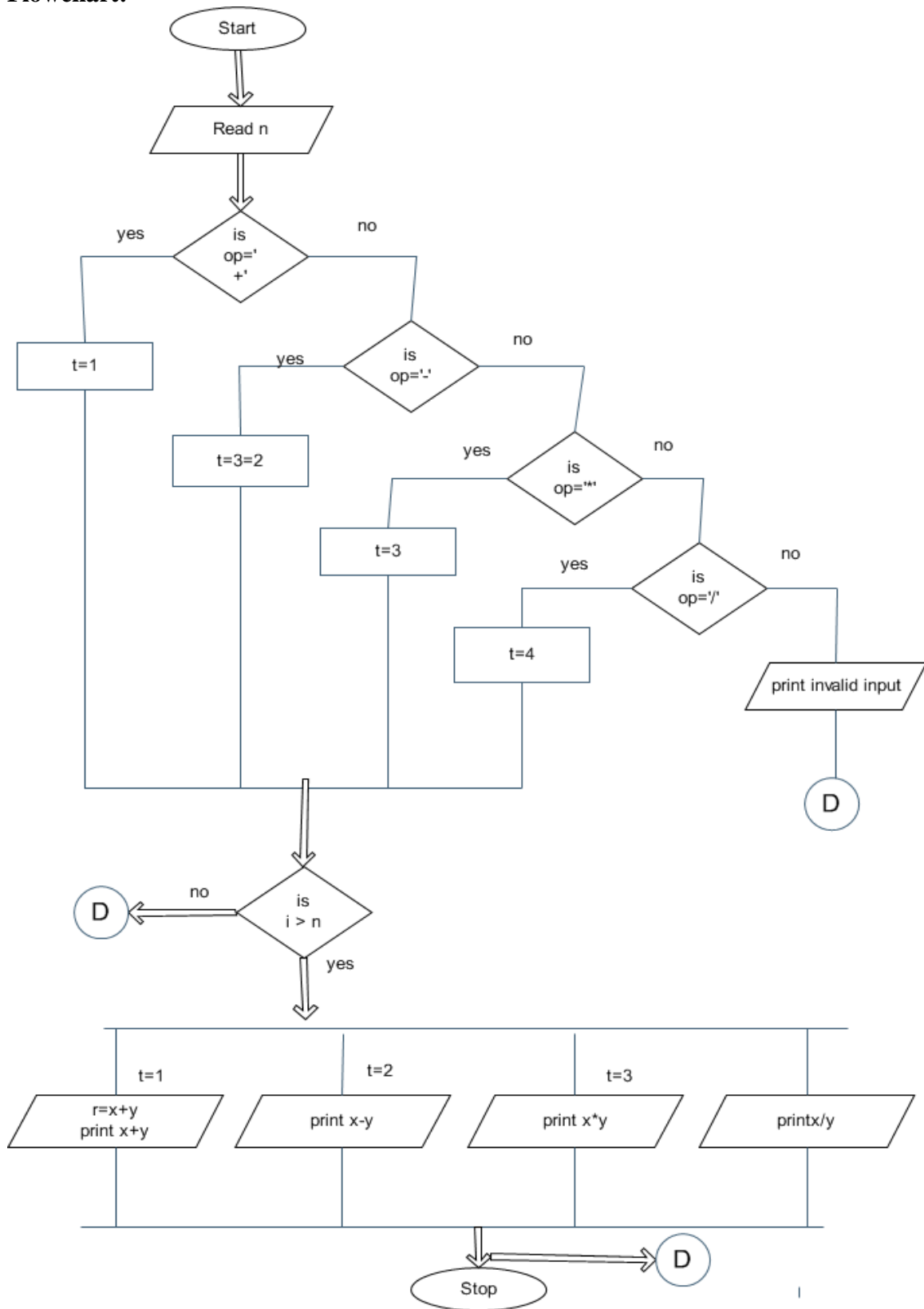
R=a%b

Go to step 8

Step 8: write R

Step 9:End

Flowchart:



Program:

```

#include<stdio.h>
main()
{
    char op;
    float a,b,c;
    clrscr();
    printf("enter two operands:");
    scanf("%d%d",&a,&b);
    printf("enter an operator:");
    scanf(" %c",&op);
    switch(op) // used to select particular case from the user
    {
        case '+':printf("sum of two numbers %2d %2d is:
%d",a,b,a+b);
            break;
        case '-':printf("subtraction of two numbers %2d %2d is:
%d",a,b,a-b);
            break;
        case '*':printf("product of two numbers %2d %2d is:
%d",a,b,a*b);
            break;
        case '/':printf("quotient of two numbers %2d %2d is:
%d",a,b,a/b);
            break;
        case '%':printf("remainder of two numbers %2d %2d is:
%d",a,b,c);
            break;
        default:printf("please enter correct operator");
            break;
    }
    getch();
}

```

Input/Output:

1. enter two operands:2 3
enter an operator:+
sum of two numbers 2 3 is: 5
2. enter two operands:3 4
enter an operator: -
subtraction of two numbers 3 4 is: -1
3. enter two operands:3 5
enter an operator:*
product of two numbers 3 5 is: 15
4. enter two operands:5 2
enter an operator: /

quotient of two numbers 5 2 is: 2
5. enter two operands:5 2
enter an operator:%
remainder of two numbers 5 2 is: 1

conclusion: The program is error free

VIVA QUESATIONS:

1) What are the various types of arithmetic operators ?

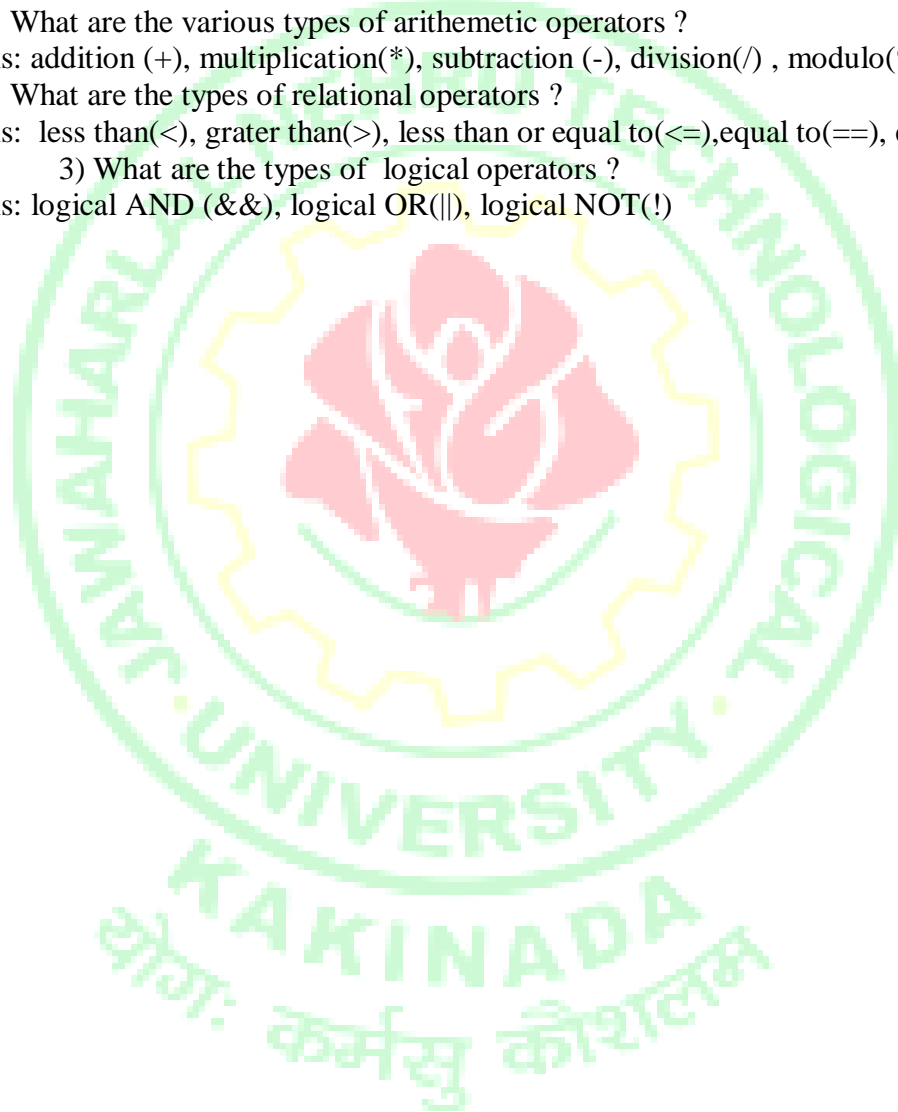
Ans: addition (+), multiplication(*), subtraction (-), division(/) , modulo(%).

2) What are the types of relational operators ?

Ans: less than(<), grater than(>), less than or equal to(<=),equal to(==), etc..,

3) 3) What are the types of logical operators ?

Ans: logical AND (&&), logical OR(||), logical NOT(!)



Exercise: 6

a) String functions (predefined) examples.

Description:

Apply some of the predefined functions on given strings. These are included in string.h header file.

Algorithm

Step 1 : start
 Step 2: read s1,s2,s3
 Step 3: l1= strlen(s1)
 Step 4 : display “Length of a string “ s1
 Step 5 : strcpy(s3,s1)
 Step 6 : display “after copying string 3 is “,s3
 Step 7 : if(!strcmp(s1,s2)) go to step 8 otherwise go to step 9
 Step 8 : display “strings are equal”
 Step 9 : display “String are not equal “
 Step 10 : display “reverse of 1st string is “, strrev(s1)
 Step 11 : display “after adding two string s1 is “ strcat(s1,s2)
 Step 12 : stop

Program

```
#include<string.h>
#include<stdlib.h>
Main()
{
  char s1[25],s2[25],s3[25];
  int l1,l2;
  printf(“Enter three strings “);
  gets(s1);
  fflush();
  gets(s2);
  fflush()
  gets(s3);
  l1 = strlen(s1);
  printf(“\nlength of s1 is “,l1);
  printf(“\nAfter copying 1st string into 3rd string is”,strcpy(s3,s1));
  if(! Strcmp(s1,s2))
  printf(“\nthe two strings are equal “);
  else
  printf(“\nString are not equal “);
  printf(“\nreverse of first string is “,strrev(s1));
  printf(“\nafter adding first two strings “,strcat(s1,s2));
}
```

Exercise : 6

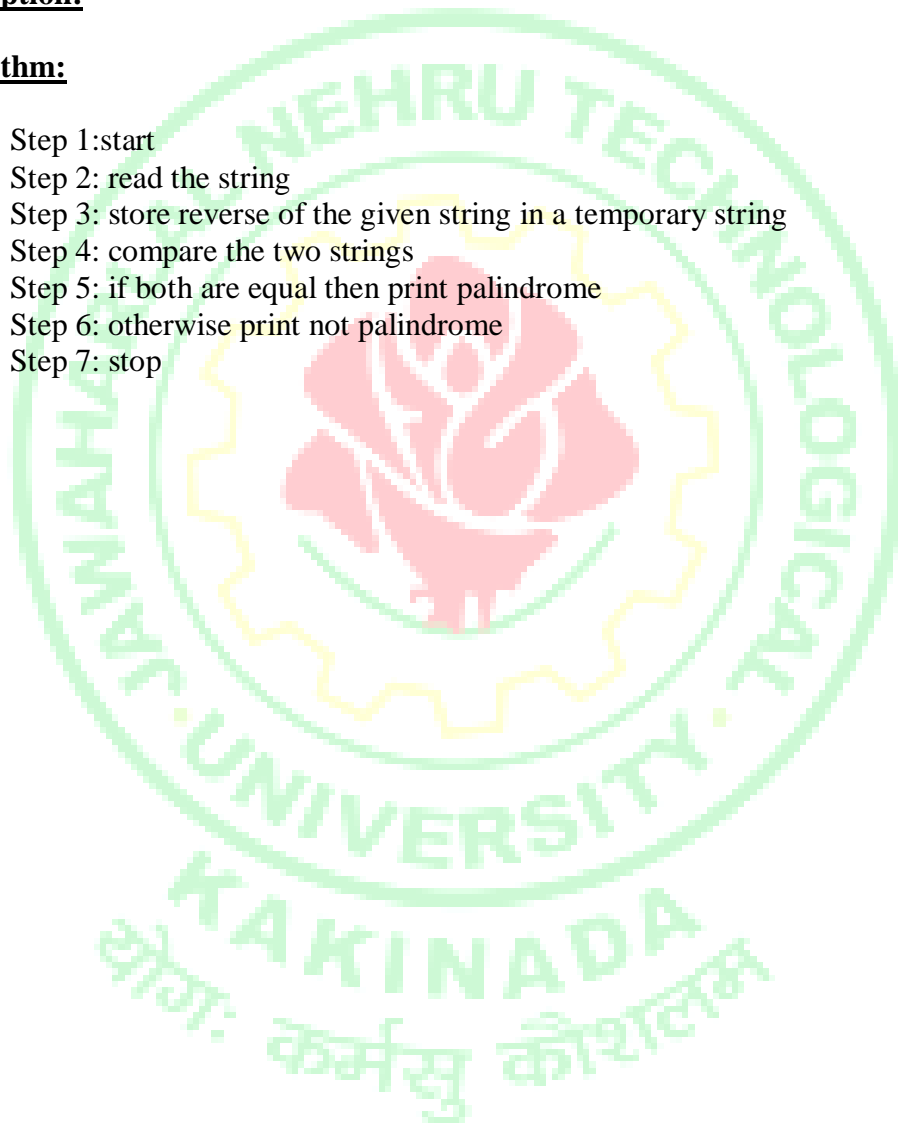
- a) Verifying a string for its palindrome property.

Description:

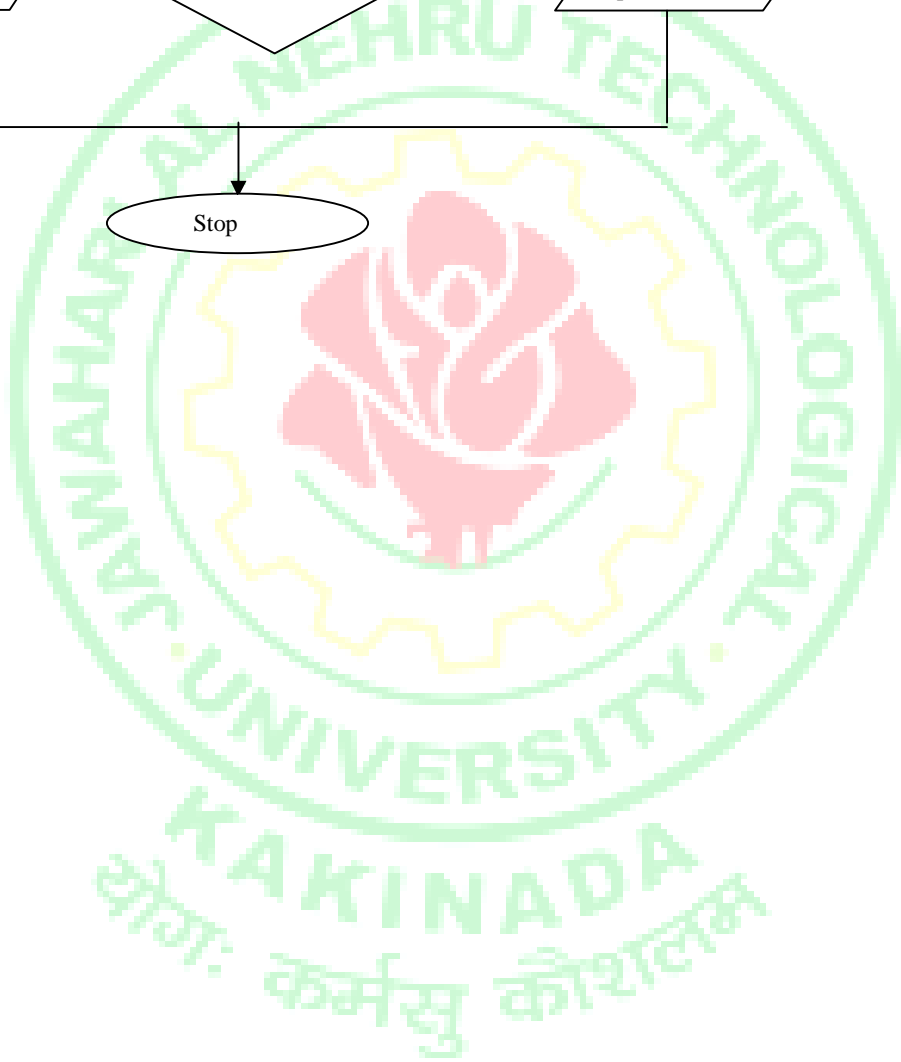
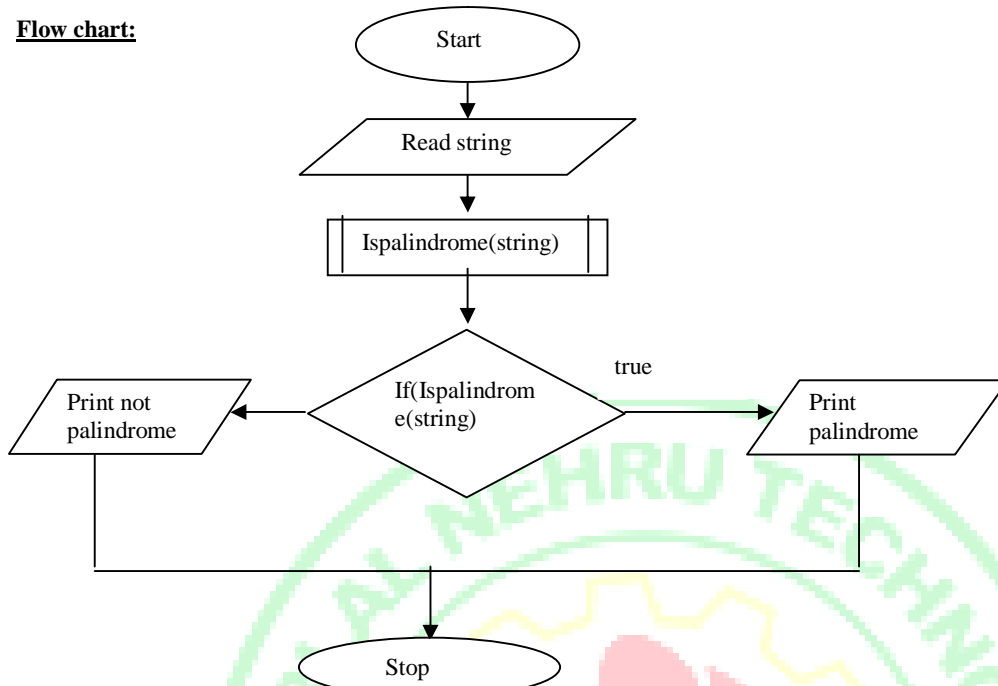
Read a string, compare first and last characters of a string, if equal continue up to middle of the string. If the comparison fails at any character the string is not a palindrome otherwise palindrome property satisfies.

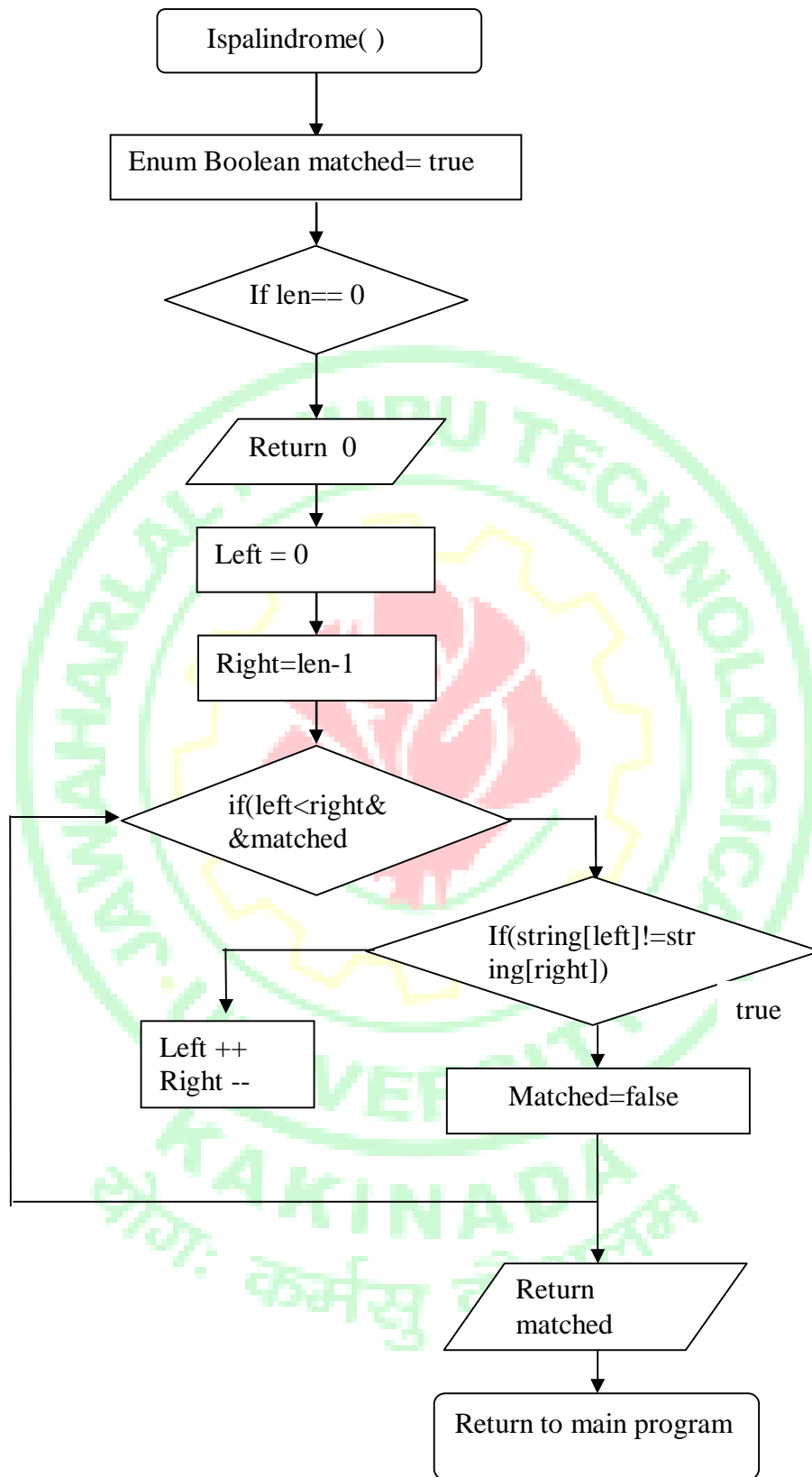
Description:**Algorithm:**

- Step 1: start
- Step 2: read the string
- Step 3: store reverse of the given string in a temporary string
- Step 4: compare the two strings
- Step 5: if both are equal then print palindrome
- Step 6: otherwise print not palindrome
- Step 7: stop



Flow chart:





Program:

```

enum Boolean{ false,true };
enum Boolean IsPalindrome(char string[])
{
    int left,right,len=strlen(string);
    enum Boolean matched=true;
    if(len==0)
        return 0;
    left=0;           right=len-1;
    /* Compare the first and last letter,second & second last & so on */
    while(left<right&&matched)
    {
        if(string[left]!=string[right])
            matched=false;
        else
        {
            left++;    right--;
        }
    }
    return matched;
}
int main()
{
    char string[40];
    clrscr();
    printf("****Program to test if the given string is a palindrome****\n");
    printf("Enter a string:");
    scanf("%s",string);
    if(IsPalindrome(string))
        printf("The given string %s is a palindrome\n",string);
    else
        printf("The given string %s is not a palindrome\n",string);
    getch();
}

```

Output:

1. Enter the string:malayalam
The given string malayalam is a palindrome
2. Enter the string:india
The given string india is not a palindrome

Conclusion: The program is error free

VIVA QUESATIONS:

- 1) What is meant by palindrome ?
Ans: If the reverse of a string/number is equal to original string/ number then it is called palindrome.
- 2) What is the use of gets() function ?
Ans: To read the string at a time
- 3) What is the use of puts() function ?
Ans: To write the string at a time

Exercise : 6

- b) To replace a character of string either from beginning or ending or at a specified location.

Description:

Replace a character of string either from beginning or ending or at a specified location.

Algorithm :

Step 1: start
 Step 2 : read s1
 Step 3 : l=length(s1)
 Step 4 : read ch
 Step 5 : display "Enter choice to replace "
 Step 6 : display if (opt= 'b') then str[0] = ch goto step 12
 Step 7 : if opt = 'e' then str[l-1]=ch go to step 12
 Step 8 : if opt = 'p' read position
 Step 9 : if pos >=0 and pos <=l-1 then goto step 10 otherwise goto step 11
 Step 10 : str[pos] = ch
 Step 11 : display "position not matched "
 Step 12 : display "after replacing string is "s1
 Step 13 : stop

Program :

```
#include<stdio.h>
main()
{
  Char s1[50], ch,opt; int pos;
  gets(s1);
  l=strlen(s1);
  printf("\nEnter character to replace :");
  scanf("%c",&ch);
  fflush();
  printf("\nb – begin, e – ending , p – position");
  scanf("%c",&opt);
  switch(opt)
  {
    case 'b' : s1[0] = ch; break;
    case 'e' : s1[l-1] = ch; break;
    case 'p' : printf("\nEnter position to replace ");
              scanf("%d",&pos);
              if( (pos >=0) && (pos <=l-1))
                s1[pos]= ch;
              else
                printf("\nposition not matched "); break;
  }
  printf("\n after replacing string is %s", s1); }
```


Exercise : 7

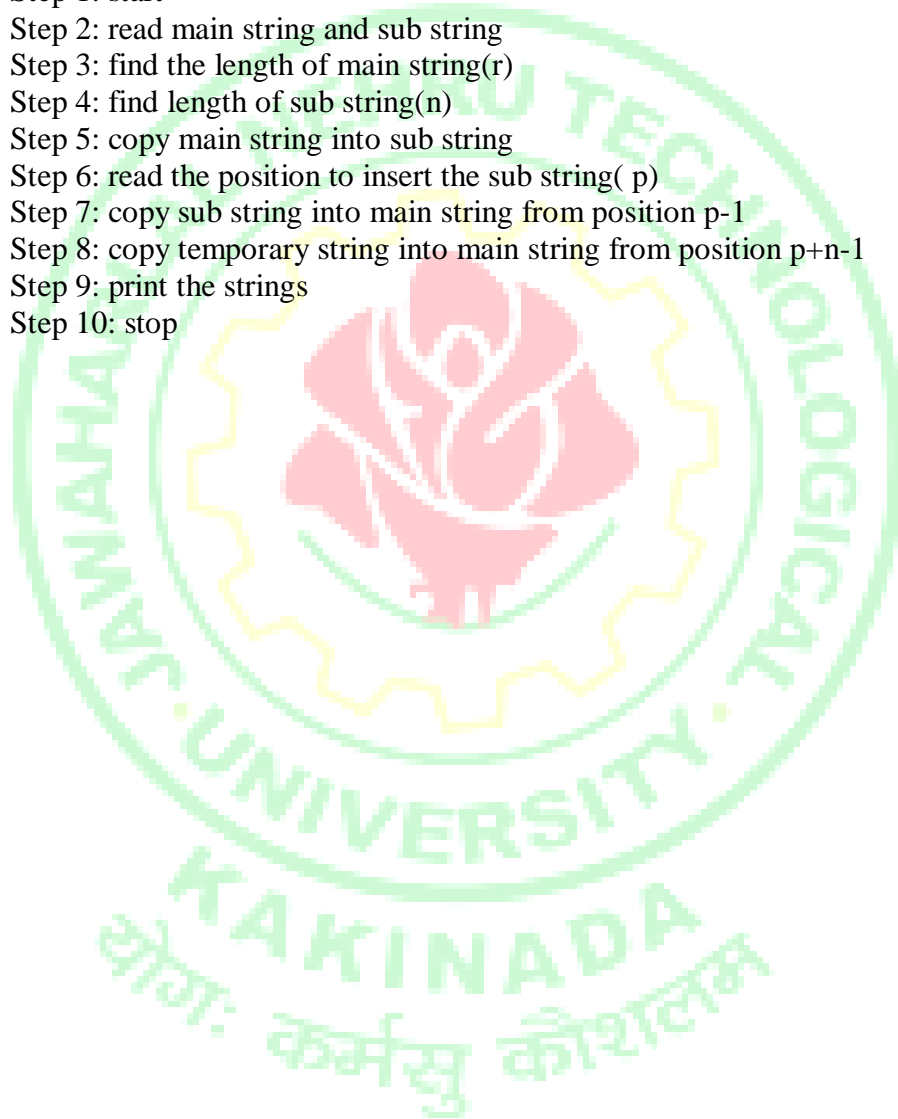
- a) Functions to insert a sub string into given main string from a given position

Description:

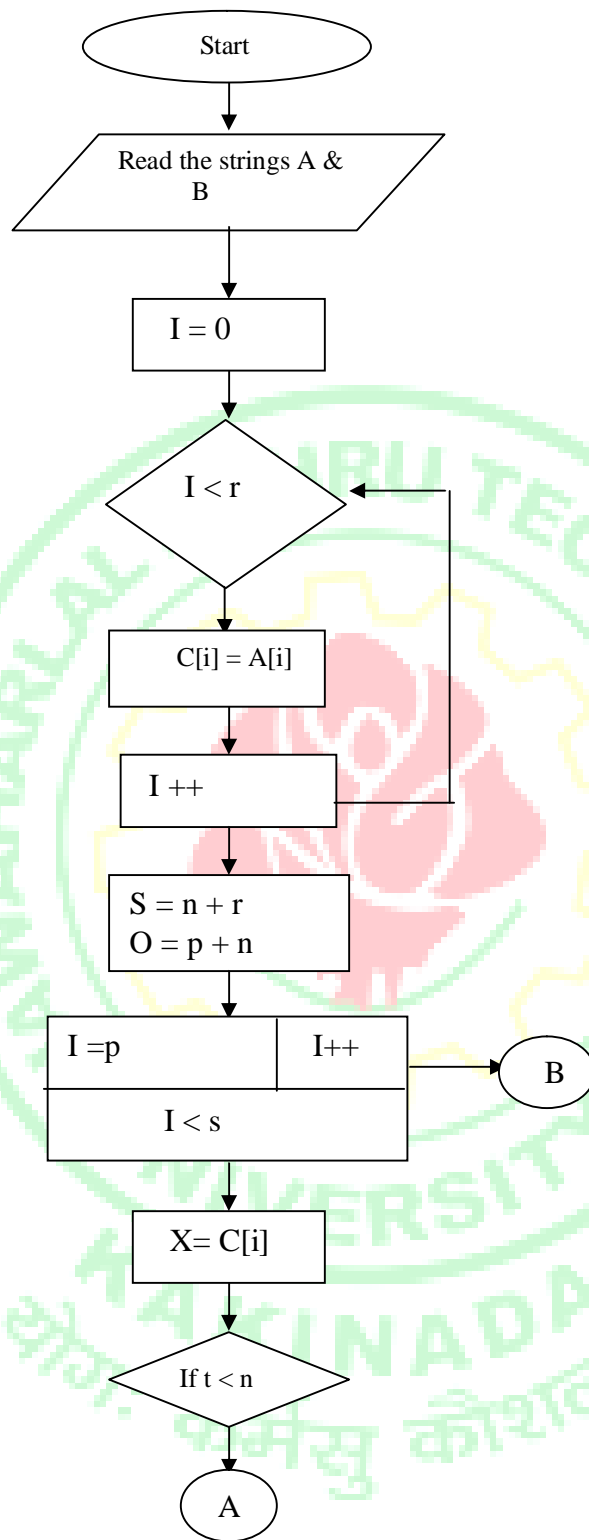
in this program we need to insert a string into another string from a specified position.

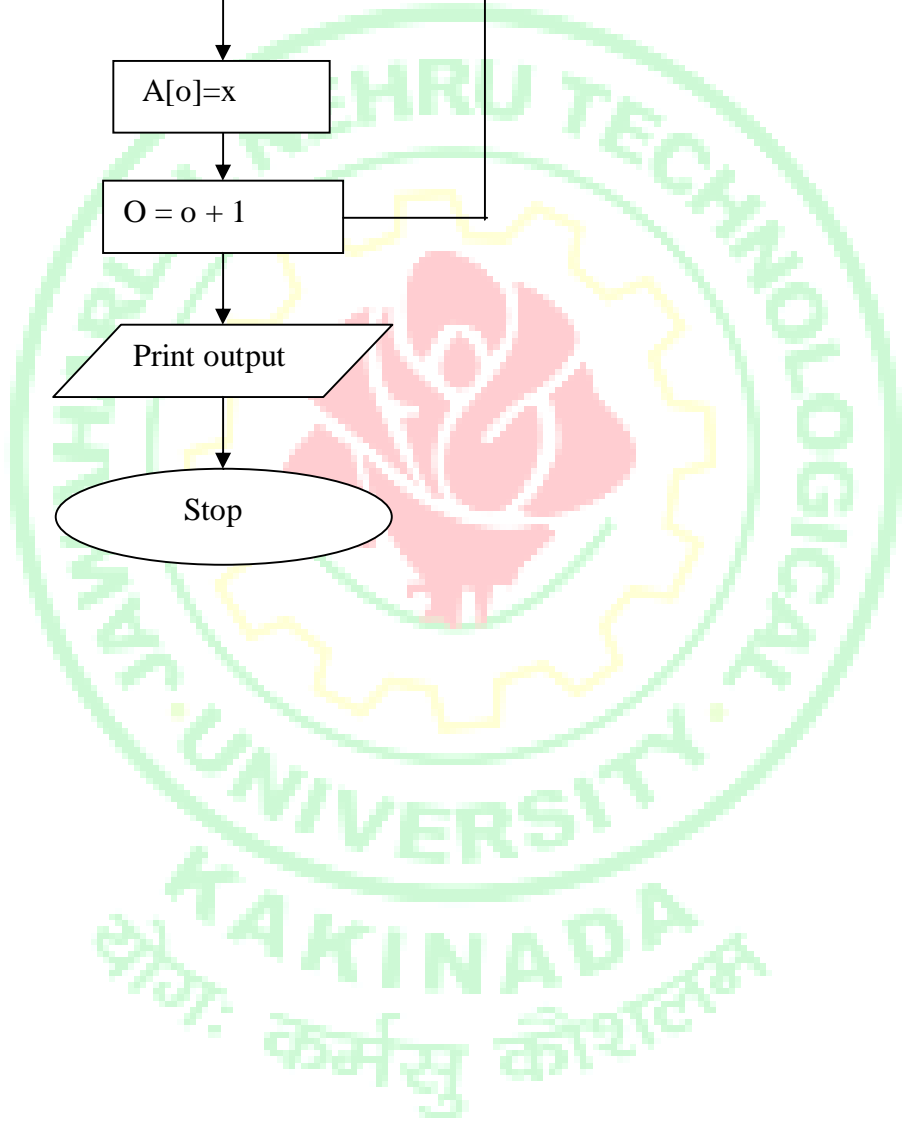
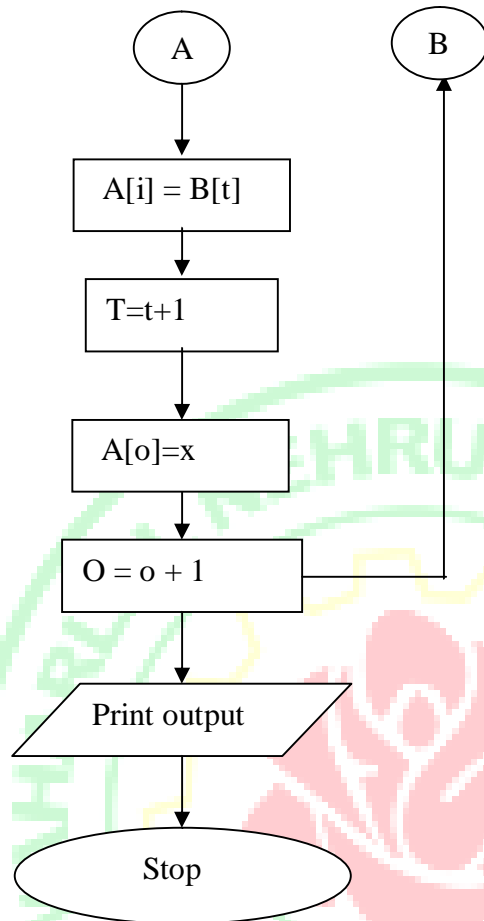
Algorithm:

- 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



Flow chart:





Program:

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

void main()
{
    char a[10];
    char b[10];
    char c[10];
    int p=0,r=0,i=0;
    int t=0;
    int x,g,s,n,o;
    clrscr();

    puts("Enter First String:");
    gets(a);
    puts("Enter Second String:");
    gets(b);
    printf("Enter the position where the item has to be inserted: ");
    scanf("%d",&p);
    r = strlen(a);
    n = strlen(b);
    i=0;

    // Copying the input string into another array
    while(i <= r)
    {
        c[i]=a[i];
        i++;
    }
    s = n+r;
    o = p+n;

    // Adding the sub-string
    for(i=p;i<s;i++)
    {
        x = c[i];
        if(t<n)
        {
            a[i] = b[t];
            t=t+1;
        }
        a[o]=x;
        o=o+1;
    }

    printf("%s", a);
    getch();
}
```

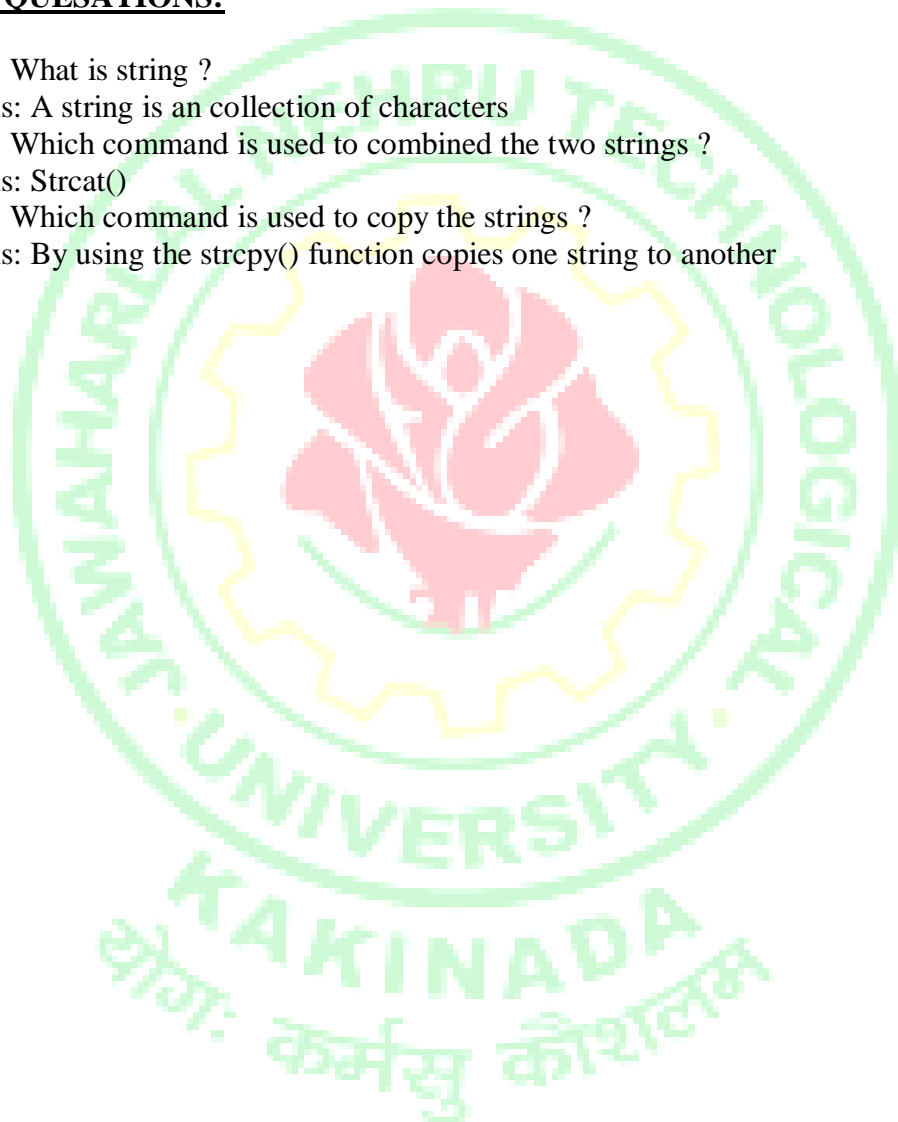
Output:

1. enter first string:
computer
2. enter second string:
gec
3. enter the position where the item has to be inserted:3
comgecputer

conclusion : the program is error free

VIVA QUESATIONS:

- 1) What is string ?
Ans: A string is an collection of characters
- 2) Which command is used to combined the two strings ?
Ans: Strcat()
- 3) Which command is used to copy the strings ?
Ans: By using the strcpy() function copies one string to another



Exercise : 7

b) To delete n characters from a given position in a given string

Description: I

In this program we need to delete a string from the given string at a specified position.

Algorithm:

Step 1: start

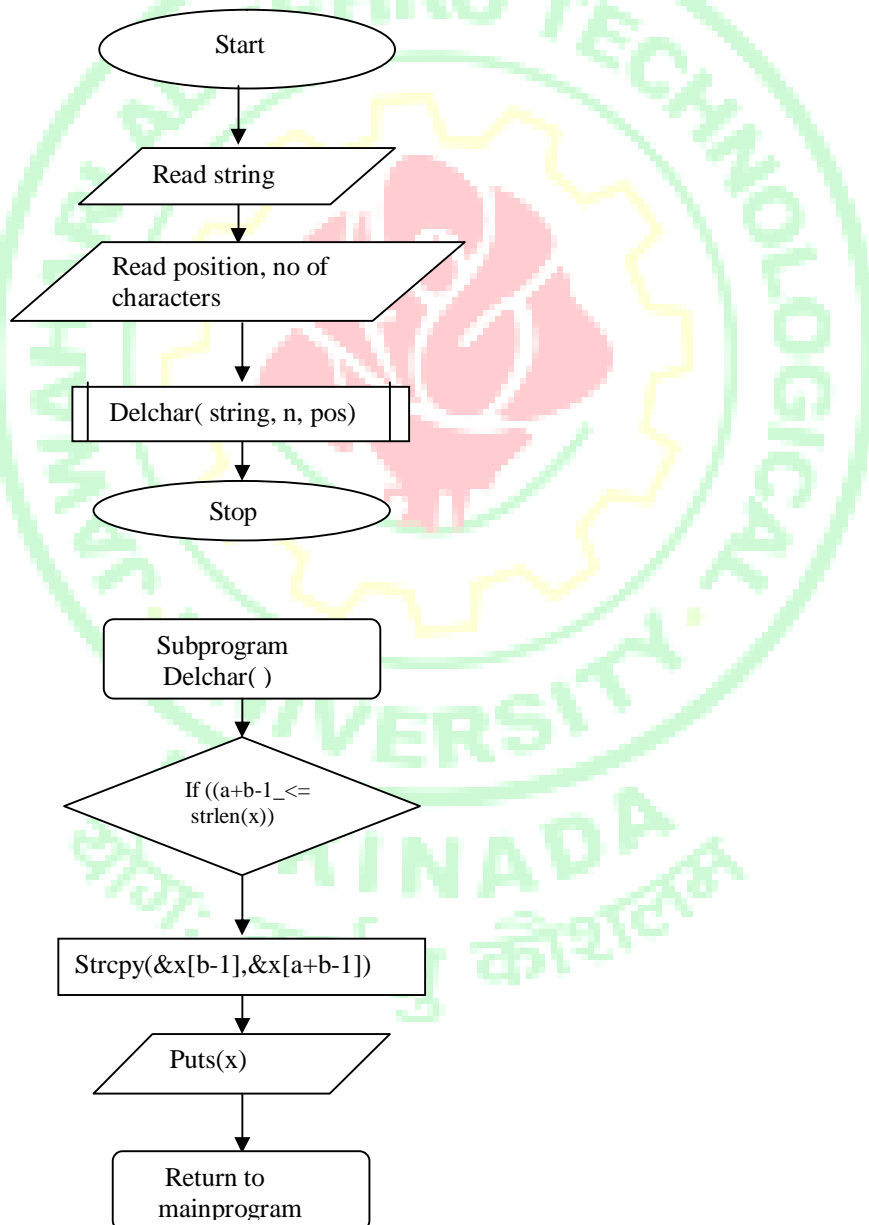
Step 2: read string

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

void delchar(char *x,int a, int b);
void main()
{
    char string[10];
    int n,pos,p;
    clrscr();

    puts("Enter the string");
    gets(string);
    printf("Enter the position from where to delete");
    scanf("%d",&pos);
    printf("Enter the number of characters to be deleted");
    scanf("%d",&n);
    delchar(string, n,pos);
    getch();
}

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

```

Output:

1. enter the string
nagaraju
Enter the position from where to delete:4
Enter the number of charcters to be deleted3
nagju
2. enter the string
kaliraju
Enter the position from where to delete:0
Enter the number of charcters to be deleted4
Raju

Conclusion: the program is error free

VIVA QUESATIONS:

- 1) Which command is used to delete the strings ?
Ans: delstr();
- 2) What are the various types of string functions ?
Ans: Strcat(), strcpy(), delstr(), substr() ,strlen()etc...

Exercise : 8

To read the two complex numbers and perform the addition and multiplication of these two numbers.

Description:

In this program the complex number means it contains the two parts . first one is real part and second one is imaginary part(2+3i).by taking these two complex numbers we can perform the addition and multiplication operation.

Algorithm:

Step 1: Start

Step 2: declare structure for complex numbers

Step 3: read the complex number

Step 4: read choice

Step 5: if choice=1 then addition operation will perform and it contains following

steps

i) $w.\text{realpart} = w1.\text{realpart} + w2.\text{realpart};$

ii) $w.\text{imgpart} = w1.\text{imgpart} + w2.\text{imgpart};$ goto step 4

Step 6: if choice=2 then multiplication operation will perform and it contains following steps

i) $w.\text{realpart} = (w1.\text{realpart} * w2.\text{realpart}) - (w1.\text{imgpart} * w2.\text{imgpart});$

ii) $w.\text{imgpart} = (w1.\text{realpart} * w2.\text{imgpart}) + (w1.\text{imgpart} * w2.\text{realpart});$

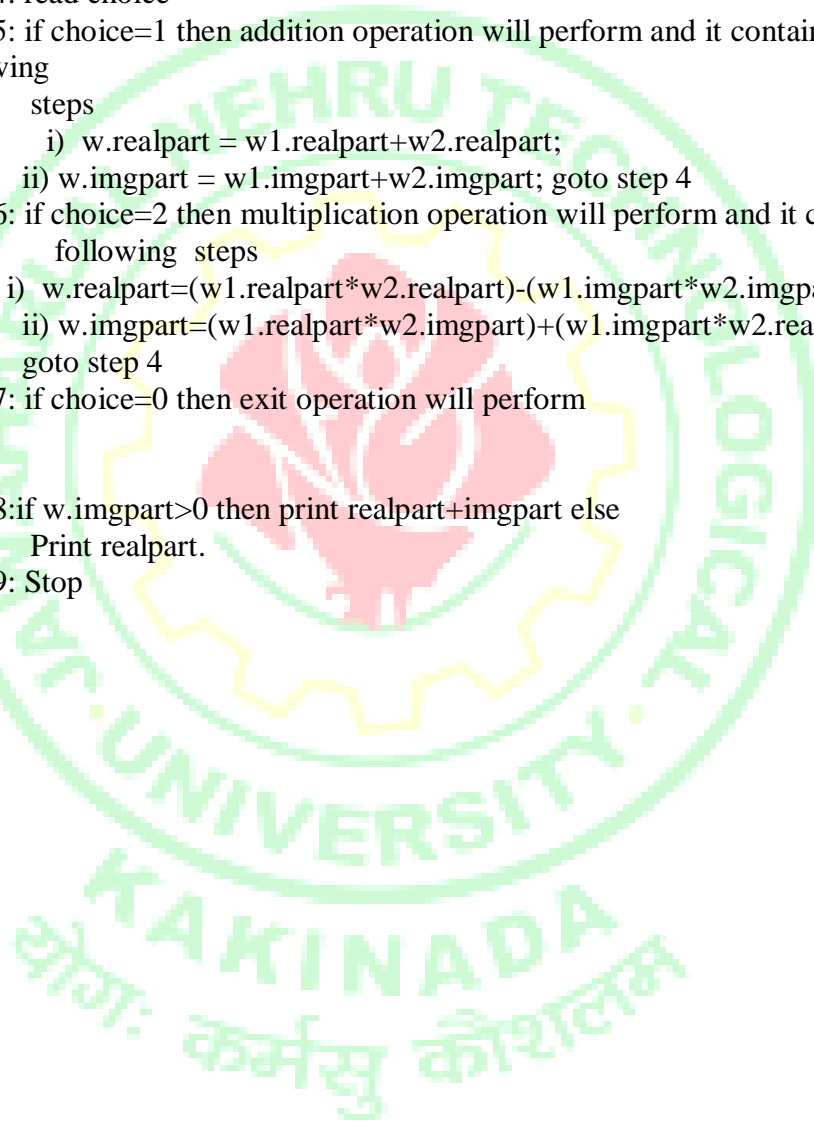
goto step 4

Step 7: if choice=0 then exit operation will perform

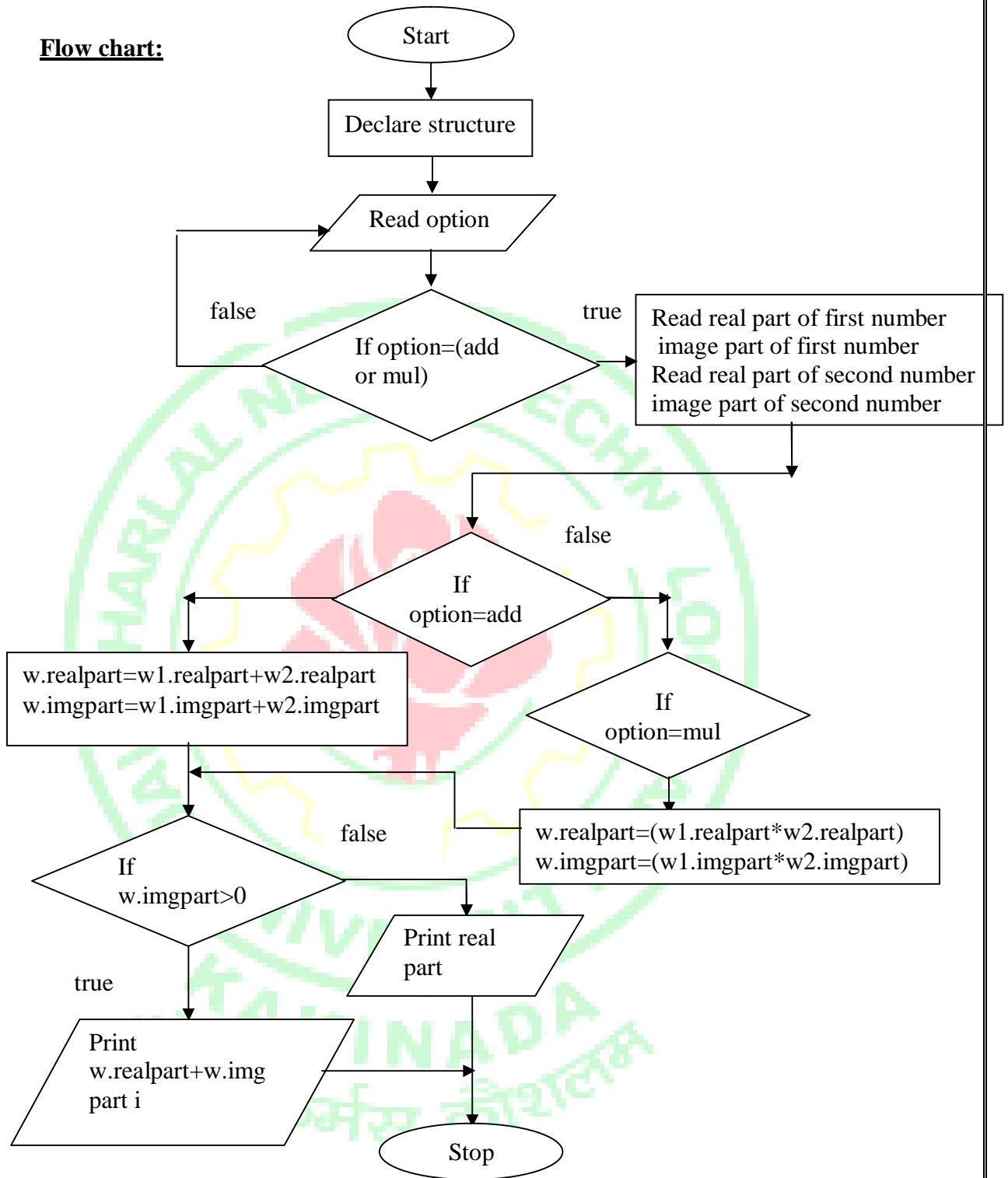
Step 8: if $w.\text{imgpart} > 0$ then print $\text{realpart} + \text{imgpart}$ else

Print $\text{realpart}.$

Step 9: Stop



Flow chart:




```

w.realpart = w1.realpart+w2.realpart;
w.imgpart = w1.imgpart+w2.imgpart;
break;
/*multiplication of complex number*/
case 2:
w.realpart=(w1.realpart*w2.realpart)-(w1.imgpart*w2.imgpart);
w.imgpart=(w1.realpart*w2.imgpart)+(w1.imgpart*w2.realpart);
break;
}
if (w.imgpart>0)
printf("\n Answer = %lf+%lfi",w.realpart,w.imgpart);
else
printf("\n Answer = %lf%lfi",w.realpart,w.imgpart);
getch();
main();
}

```

Output:

***** MAIN MENU *****

Select your option:

1 : ADD

2 : MULTIPLY

0 : EXIT

Enter your Option [1]

Enter two Complex Numbers (x+iy):

Real Part of First Number:2

Imaginary Part of First Number:2

Real Part of Second Number:2

Imaginary Part of Second Number:2

Answer = 4.000000+4.000000i

***** MAIN MENU *****

Select your option:

1 : ADD

2 : MULTIPLY

0 : EXIT

Enter your Option [2]

Enter two Complex Numbers (x+iy):

Real Part of First Number:2

Imaginary Part of First Number:2

Real Part of Second Number:2

Imaginary Part of Second Number:2

Answer = 0.000000+8.000000i

***** MAIN MENU *****

Select your option:

1 : ADD

2 : MULTIPLY

0 : EXIT

Enter your Option [3]

invalid option

***** MAIN MENU *****

Select your option:

1 : ADD

2 : MULTIPLY

0 : EXIT

Enter your Option [0]

Conclusion: The program is error free

VIVA QUESATIONS:

1) Define structure ?

Ans: Structure is a method for packing data of different types. Structure help to organize complex data in a more meaningful way.

2) What is use of <math.h> header file ?

Ans: It is used to access the mathematical functions in programs.

Exercise : 9

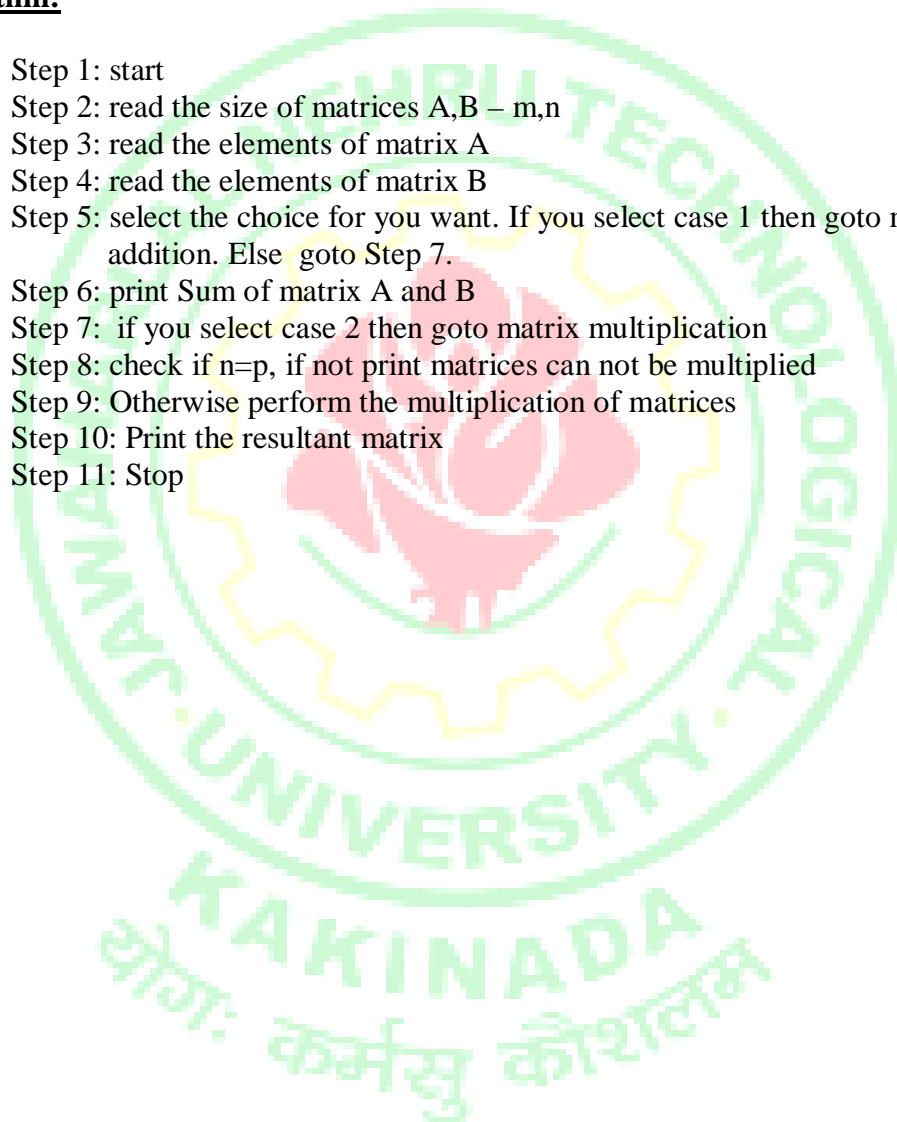
- a) To perform the addition of two matrices
- b) multiplication of two matrices by checking compatability.

Description:

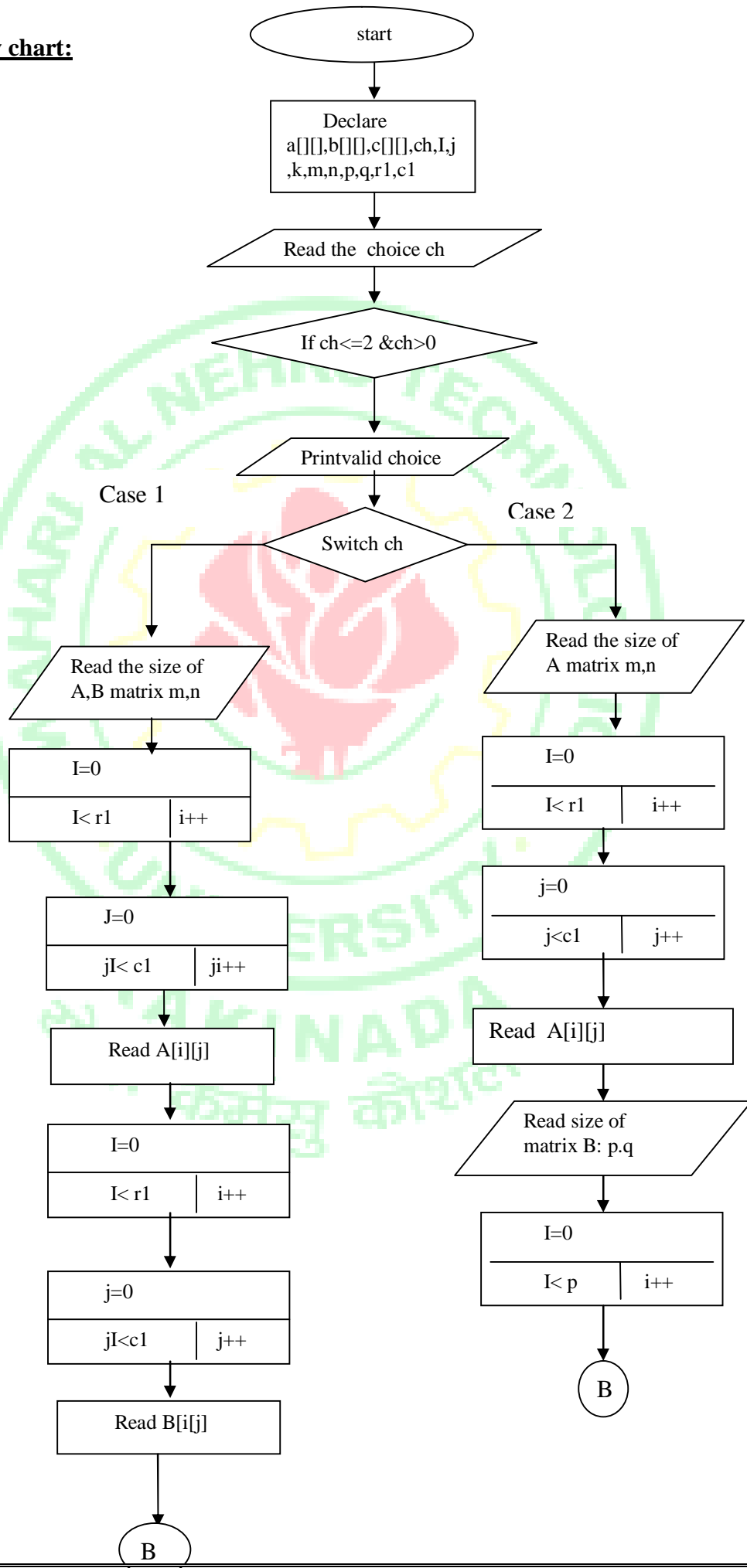
program takes the two matrixes of same size and performs the addition an also takes the two matrixes of different sizes and checks for possibility of multiplication and perform multiplication if possible.

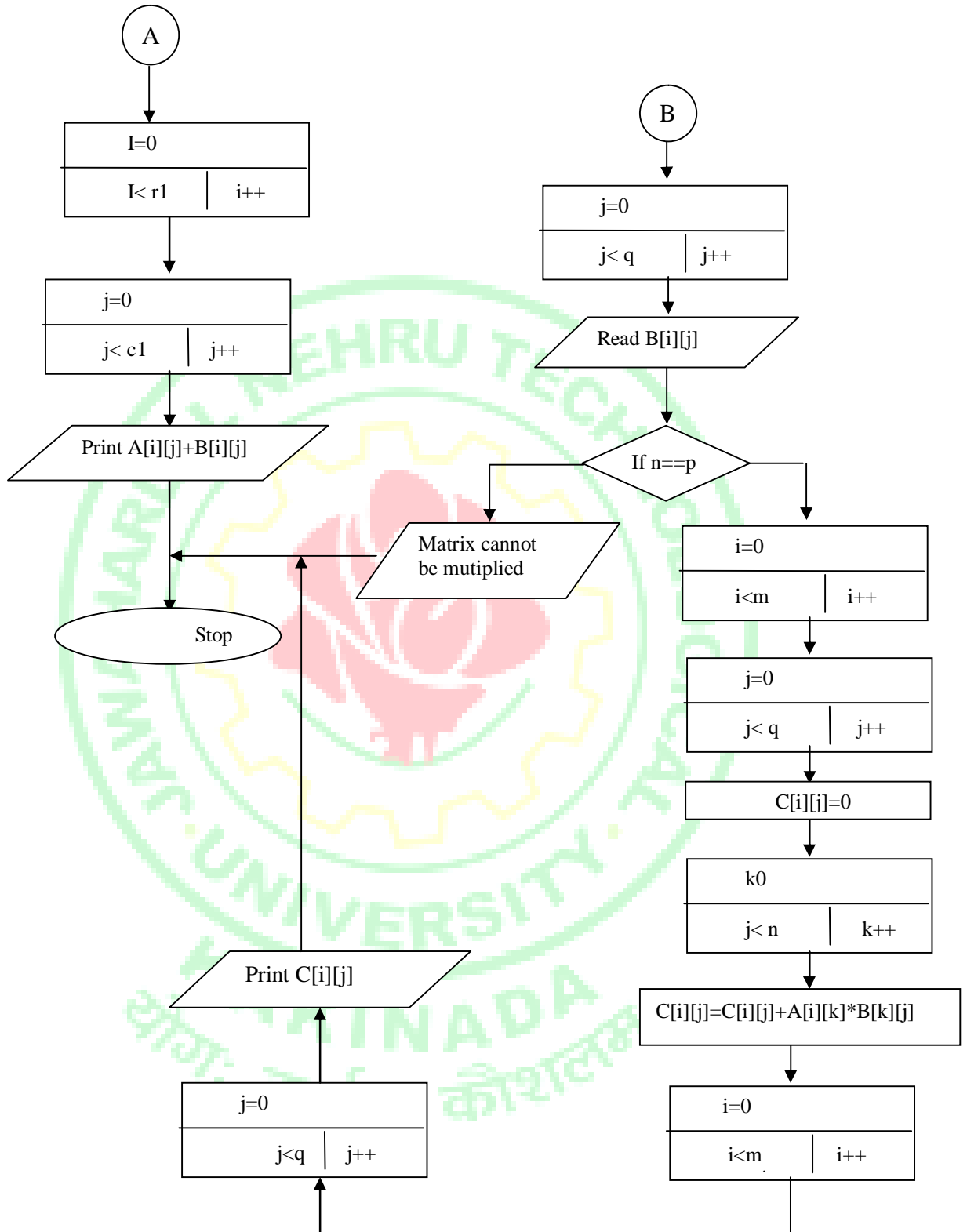
algorithm:

- Step 1: start
- Step 2: read the size of matrices A,B – m,n
- Step 3: read the elements of matrix A
- Step 4: read the elements of matrix B
- Step 5: select the choice for you want. If you select case 1 then goto matrix addition. Else goto Step 7.
- Step 6: print Sum of matrix A and B
- Step 7: if you select case 2 then goto matrix multiplication
- Step 8: check if $n=p$, if not print matrices can not be multiplied
- Step 9: Otherwise perform the multiplication of matrices
- Step 10: Print the resultant matrix
- Step 11: Stop



Flow chart:





Program:

```

#include<stdio.h>

void main()
{
    int ch,i,j,m,n,p,q,k,r1,c1,a[10][10],b[10][10],c[10][10];
    clrscr();
    printf("*****");
    printf("\n\t\tMENU");
    printf("\n*****");
    printf("\n[1]ADDITION OF TWO MATRICES");
    printf("\n[2]MULTIPLICATION OF TWO MATRICES");
    printf("\n[0]EXIT");
    printf("\n*****");
    printf("\n\tEnter your choice:\n");
    scanf("%d",&ch);

    if(ch<=2 & ch>0)
    {
        printf("Valid Choice\n");
    }
    switch(ch)
    {
        case 1:
            printf("Input rows and columns of A & B Matrix:");
            scanf("%d%d",&r1,&c1);
            printf("Enter elements of matrix A:\n");
            for(i=0;i<r1;i++)
            {
                for(j=0;j<c1;j++)
                scanf("%d",&a[i][j]);
            }
            printf("Enter elements of matrix B:\n");
            for(i=0;i<r1;i++)
            {
                for(j=0;j<c1;j++)
                scanf("%d",&b[i][j]);
            }
            printf("\n =====Matrix Addition===== \n");
            for(i=0;i<r1;i++)
            {
                For(j=0;j<c1;j++)
                printf("%5d",a[i][j]+b[i][j]);
                printf("\n");
            }
            break;

        case 2:
            printf("Input rows and columns of A matrix:");
            scanf("%d%d",&m,&n);

```



```

printf("Input rows and columns of B matrix:");
scanf("%d%d",&p,&q);
if(n==p)
{
    printf("matrices can be multiplied\n");
    printf("resultant matrix is %d*%d\n",m,q);
    printf("Input A matrix\n");
    read_matrix(a,m,n);
    printf("Input B matrix\n");
    /*Function call to read the matrix*/
    read_matrix(b,p,q);
    /*Function for Multiplication of two matrices*/
    printf("\n =====Matrix Multiplication===== \n");
    for(i=0;i<m;++i)
        for(j=0;j<q;++j)
        {
            c[i][j]=0;
            for(k=0;k<n;++k)
                c[i][j]=c[i][j]+a[i][k]*b[k][j];
        }

    printf("Resultant of two matrices:\n");
    write_matrix(c,m,q);
}
/*end if*/
else
{
    printf("Matrices cannot be multiplied.");
}
/*end else*/
break;

case 0:
    printf("\n Choice Terminated");
    exit();
    break;

    default:
        printf("\n Invalid Choice");
    }
    getch();
}

/*Function read matrix*/
int read_matrix(int a[10][10],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    return 0;
}

```

```

/*Function to write the matrix*/
int write_matrix(int a[10][10],int m,int n)
{
    int i,j;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        printf("%5d",a[i][j]);
        printf("\n");
    }
    return 0;
}

```

Output:

```

1.
*****
          MENU
*****
[1]ADDITION OF TWO MATRICES
[2]MULTIPLICATION OF TWO MATRICES
[0]EXIT
*****
Enter your choice: 1
Valid Choice
Input rows and columns of A & B Matrix:2
2
Enter elements of matrix A:
2
2
2
2
Enter elements of matrix B:
2
2
2
2
=====Matrix Addition=====
 4 4
 4 4
*****
          MENU
*****
[1]ADDITION OF TWO MATRICES
[2]MULTIPLICATION OF TWO MATRICES
[0]EXIT
*****
Enter your choice:2
Valid Choice
Input rows and columns of A matrix:2
3
Input rows and columns of B matrix:2

```

2

Matrices cannot be multiplied.

MENU

[1]ADDITION OF TWO MATRICES

[2]MULTIPLICATION OF TWO MATRICES

[0]EXIT

Enter your choice:2

Valid Choice

Input rows and columns of A matrix:2

2

Input rows and columns of B matrix:2

2

matrices can be multiplied

resultant matrix is 2*2

Input A matrix

2

2

2

2

Input B matrix

2

2

2

2

=====Matrix Multiplication=====

Resultant of two matrices:

8 8

8 8

Conclusion : The program is error free**VIVA QUESATIONS:**

1) What is condition for performing an matric addition ?

Ans: program takes the two matrixes of same size and performs the addition

2) What is condition for performing an matric addition ?

Ans: The two matrixes of different sizes and checks for possibility of multiplication and perform multiplication if possible

Exercise : 10

- a) Programs that use recursive function to find the factorial of a given integer.

Description:

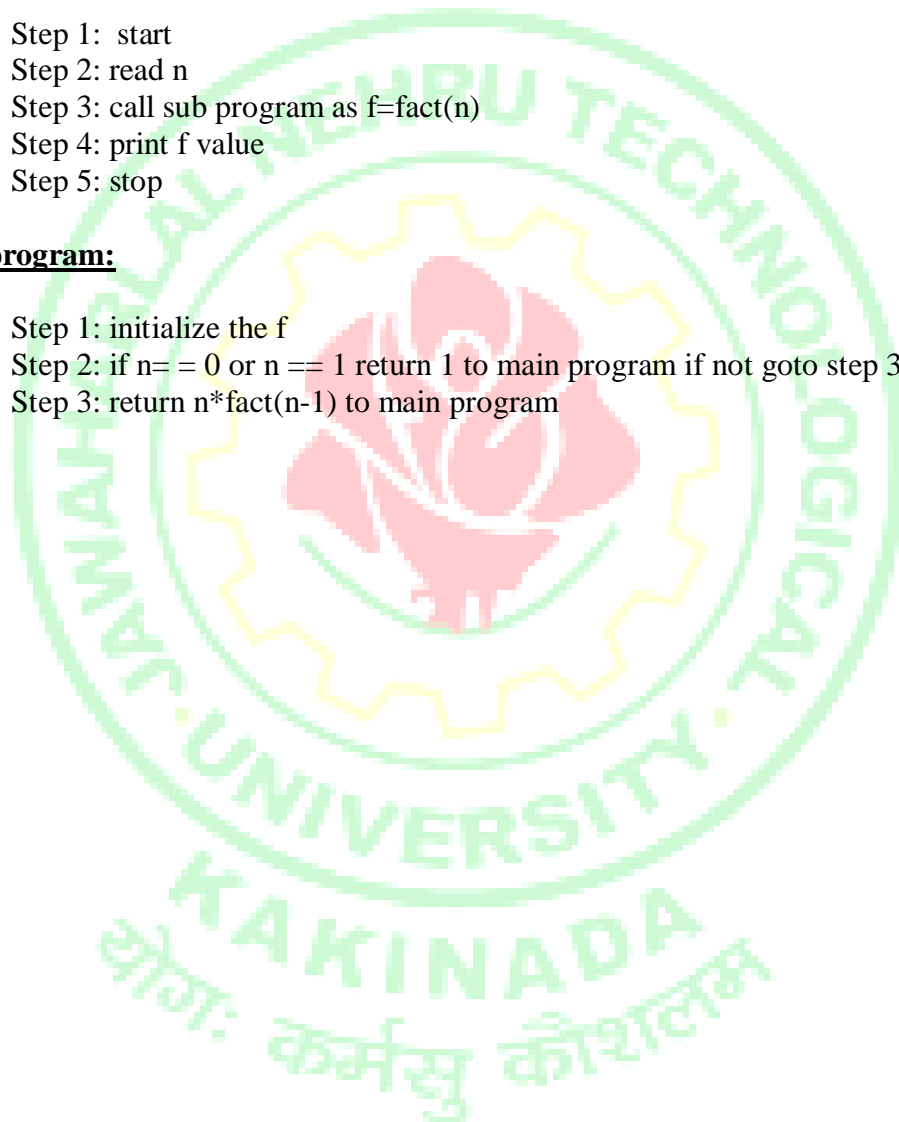
Factorial of a number is nothing but the multiplication of numbers from a given number to 1

Algorithm: main program

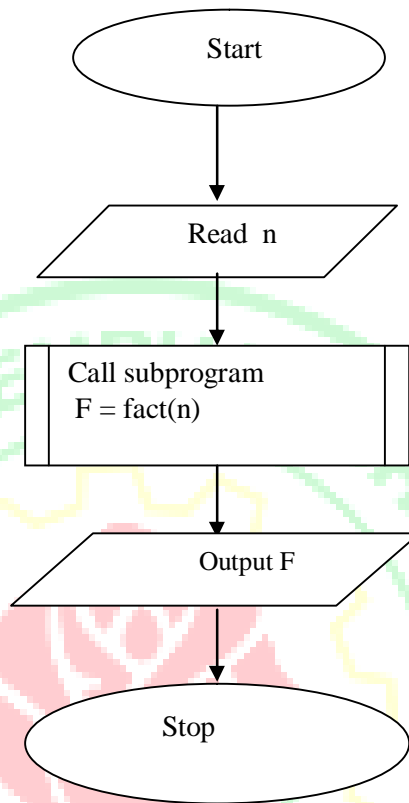
- Step 1: start
- Step 2: read n
- Step 3: call sub program as $f = \text{fact}(n)$
- Step 4: print f value
- Step 5: stop

Sub program:

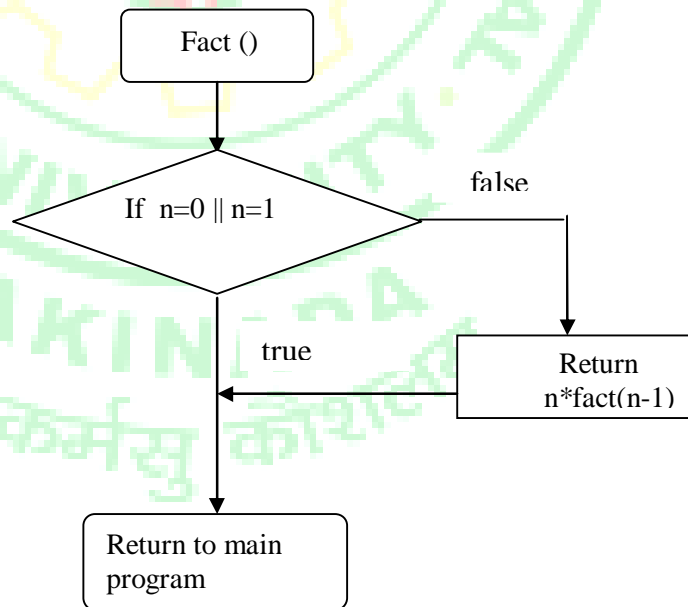
- Step 1: initialize the f
- Step 2: if $n = 0$ or $n == 1$ return 1 to main program if not goto step 3
- Step 3: return $n * \text{fact}(n-1)$ to main program



Flowchart:



Sub program



Program:

```

#include<stdio.h>
#include<conio.h>
int fact(int n)
{
    int f;
    if((n==0)||(n==1)) // check the condition for the n value
        return(n);
    else
        f=n*fact(n-1); //calculate the factorial of n
    return(f);
}
void main()
{
    int n;
    clrscr();
    printf("enter the number :");
    scanf("%d",&n);
    printf("factoria of number%d",fact(n));
    getch();
}

```

Output:

1. Enter the number : 5
Factorial of number: 120
2. Enter the number : 3
Factorial of number: 6
3. Enter the number : 9
Factorial of number: -30336

Conclusion: the program is error free

VIVA QUESATIONS:

1) What is the meaning of factorial number?

Ans : Factorial of a number is nothing but the multiplication of numbers from a given number to 1

2) What is the meaning of recursive function ?

Ans: A function call it self is called recursive function

3) define library functions ?

Ans: The functions have already been written, compiled and placed in libraries and are

called library functions.

4) Define formal parameters ?

Ans: Formal parameters are the parameters given in the function declaration and function definition.

Program that use non recursive function to find the factorial of a given integer.

Description:

Factorial of a number is nothing but the multiplication of numbers from a given number to 1

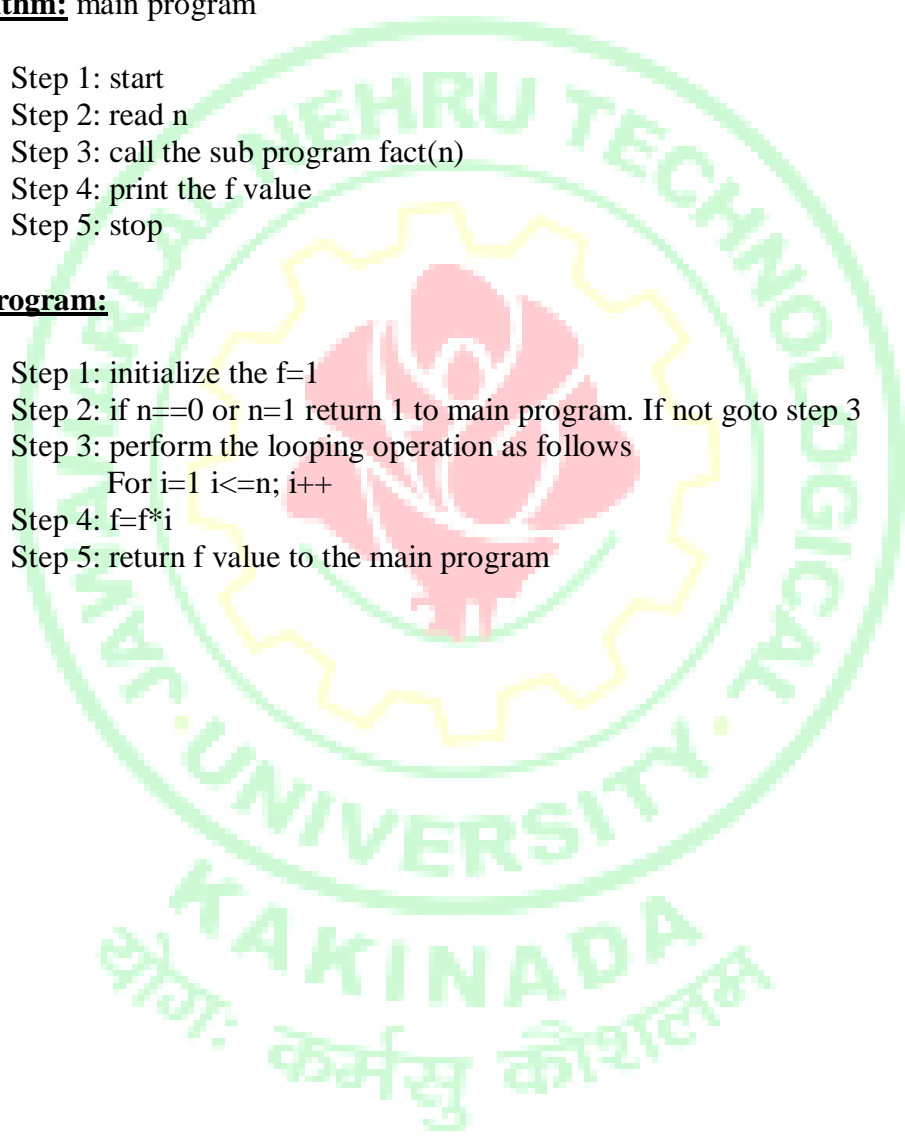
Ex: $5! = 5*4*3*2*1 = 120$

Algorithm: main program

- Step 1: start
- Step 2: read n
- Step 3: call the sub program fact(n)
- Step 4: print the f value
- Step 5: stop

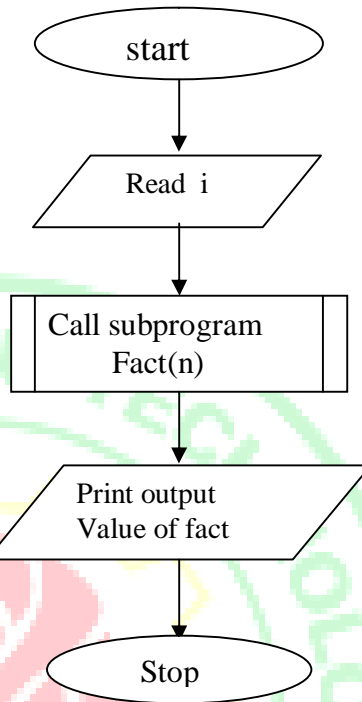
Sub program:

- Step 1: initialize the f=1
- Step 2: if n==0 or n=1 return 1 to main program. If not goto step 3
- Step 3: perform the looping operation as follows
 - For i=1 i<=n; i++
- Step 4: f=f*i
- Step 5: return f value to the main program

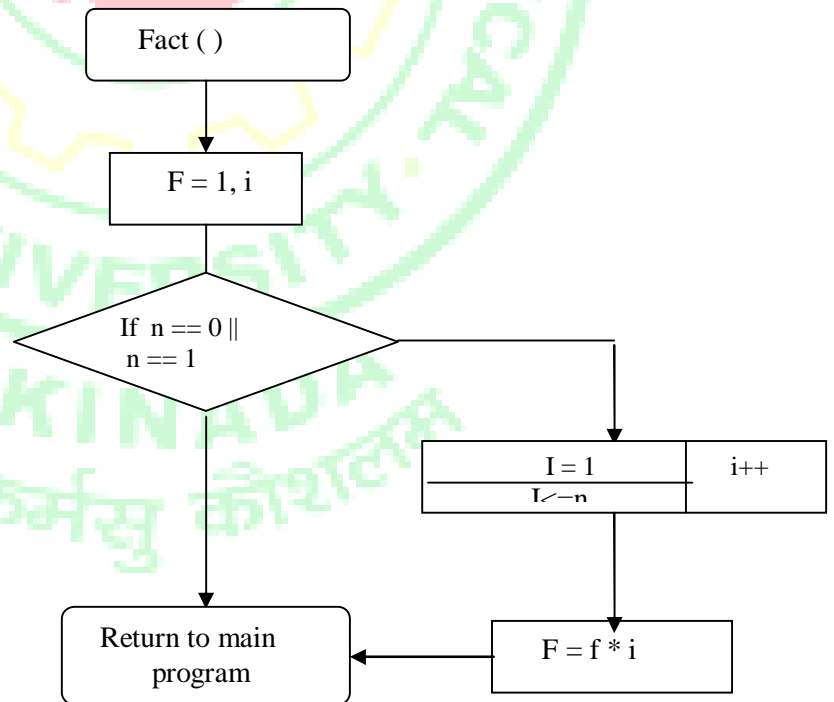


Flowchart:

Factorial nonrecursive



Sub program



Program:

```

#include<stdio.h>
#include<conio.h>
int fact(int n) //starting of the sub program
{
    int f=1,i;
    if((n==0)||(n==1)) // check the condition for n value
        return(1);
    else
        for(i=1;i<=n;i++) // perform the looping operation for calculating the factorial
            f=f*i;
    return(f);
}
void main()
{
    int n;
    clrscr();
    printf("enter the number :");
    scanf("%d",&n);
    printf("factoria of number%d",fact(n));
    getch();
}

```

Output:

1. Enter the number: 7
Factorial of number: 5040
2. Enter the number: 6
Factorial of number: 720
3. Enter the number: 8
Factorial of number: -25216

Conclusion:

The program is error free

VIVA QUESATIONS:

1) What is meant by call by value ?

Ans: passing values to the function as arguments

2) What is meant by call by reference ?

Ans: passing address to the function as arguments

3) define actual parameters ?

Ans: The actual parameters often known as arguments are specified in the function call.

Exercise : 10

b) To find the GCD of two given integers by using the 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: $GCD = \frac{\text{product of numbers}}{\text{LCM of numbers}}$

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

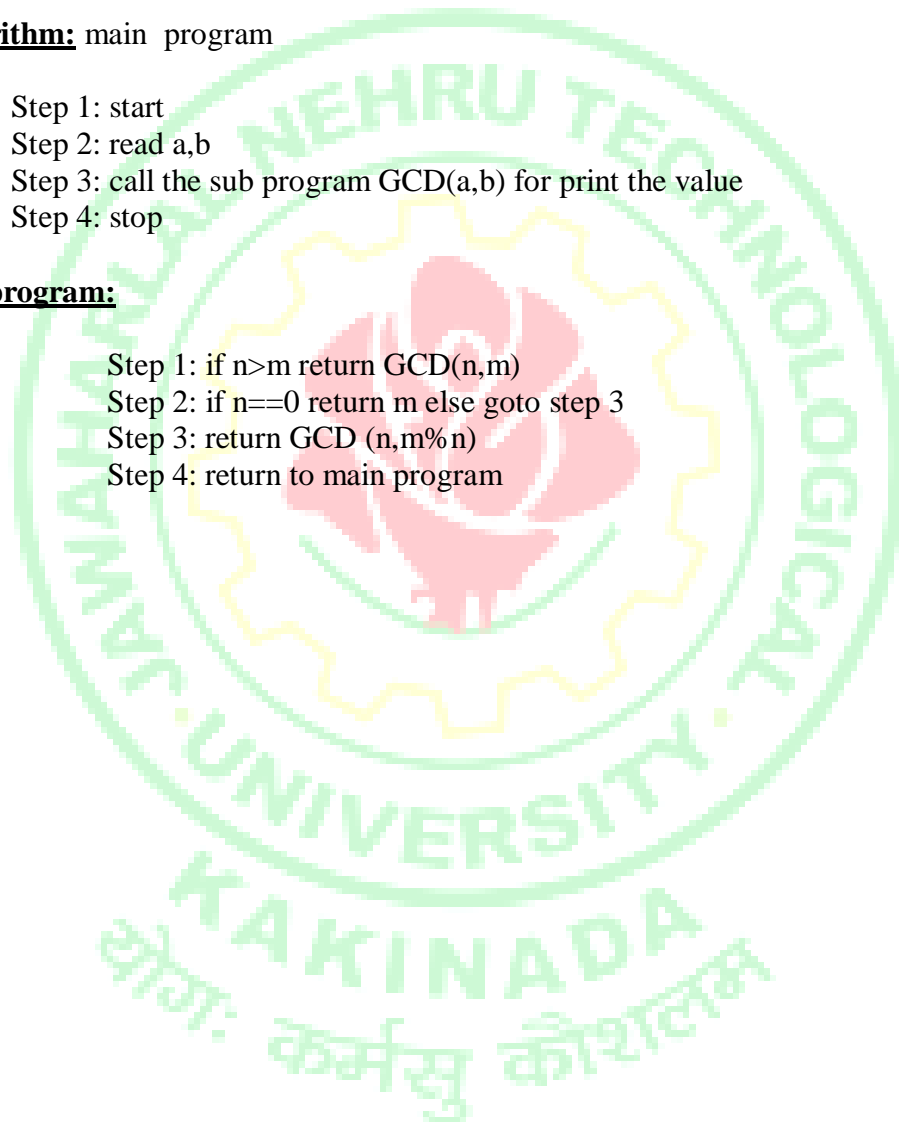
Sub program:

Step 1: if $n > m$ return GCD(n,m)

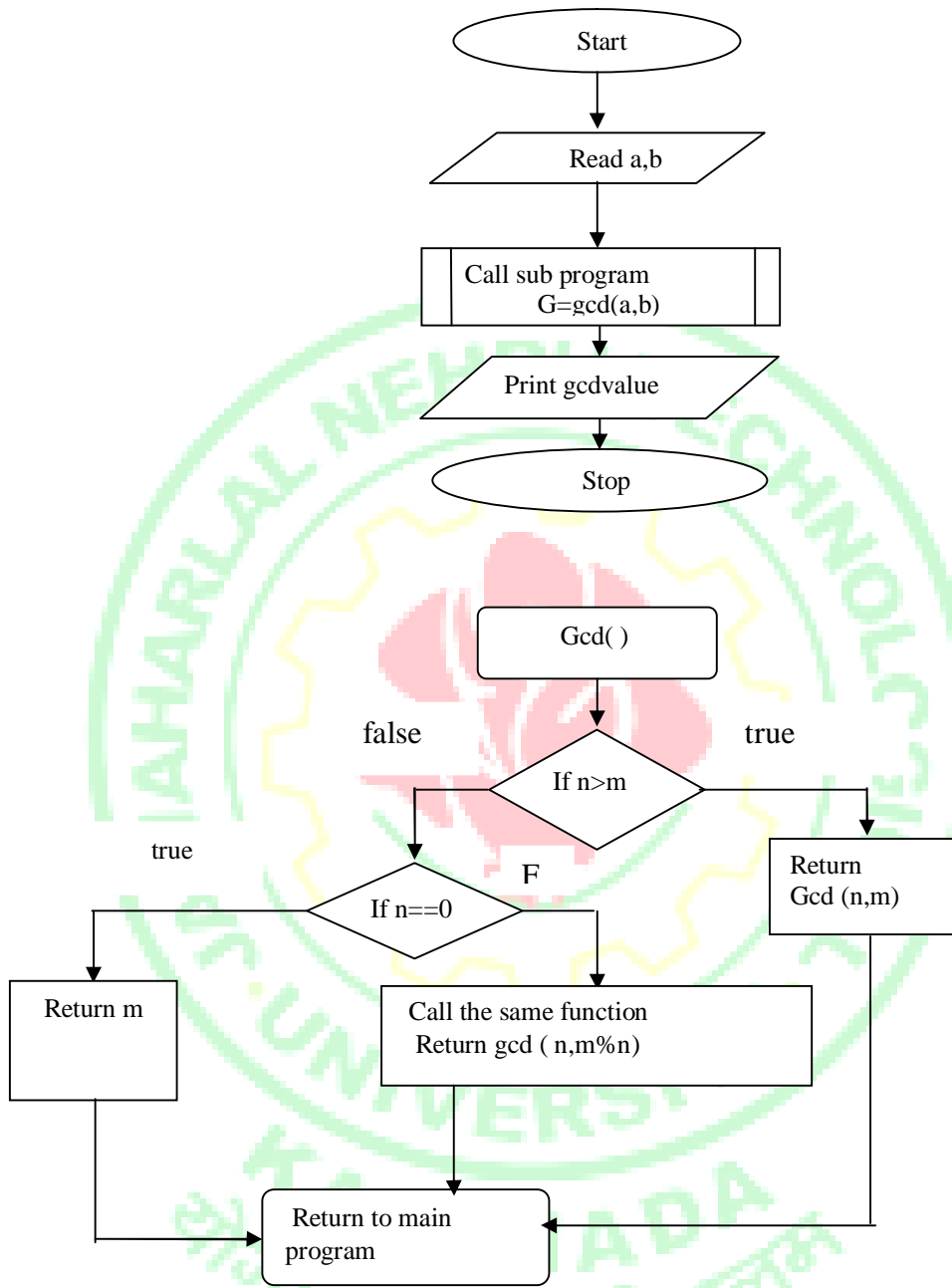
Step 2: if $n == 0$ return m else goto step 3

Step 3: return GCD (n,m%n)

Step 4: return to main program



Flowchart:



Program:

```
#include<stdio.h>
#include<conio.h>
int gcdrecursive(int m,int n) // starting of the sub program
{
    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();
}
```

Output:

1. enter the two numbers whose gcd is to be found:5,25
GCD of a,b is : 5
2. enter the two numbers whose gcd is to be found:36,54
GCD of a,b is : 18
3. enter the two numbers whose gcd is to be found:11,13
GCD of a,b is : 1

Conclusion:

The program is error free

VIVA QUESATIONS:

1) What is meaning of GCD ?

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

2) Define scope of a variable ?

Ans: The scope of a variable can be define as the region over which the variable is accessible

3) Show an scope resolution operator ?

Ans: double colon(::)

4) Define extent of a variable ?

Ans: The period of time during which memory is associated with a variable is called extent of the variable.

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

Algorithm:

Step 1: start

Step 2: read a,b

Step 3: call sub program $g = GCD(a,b)$

Step 4: print the g value

Step 5: stop

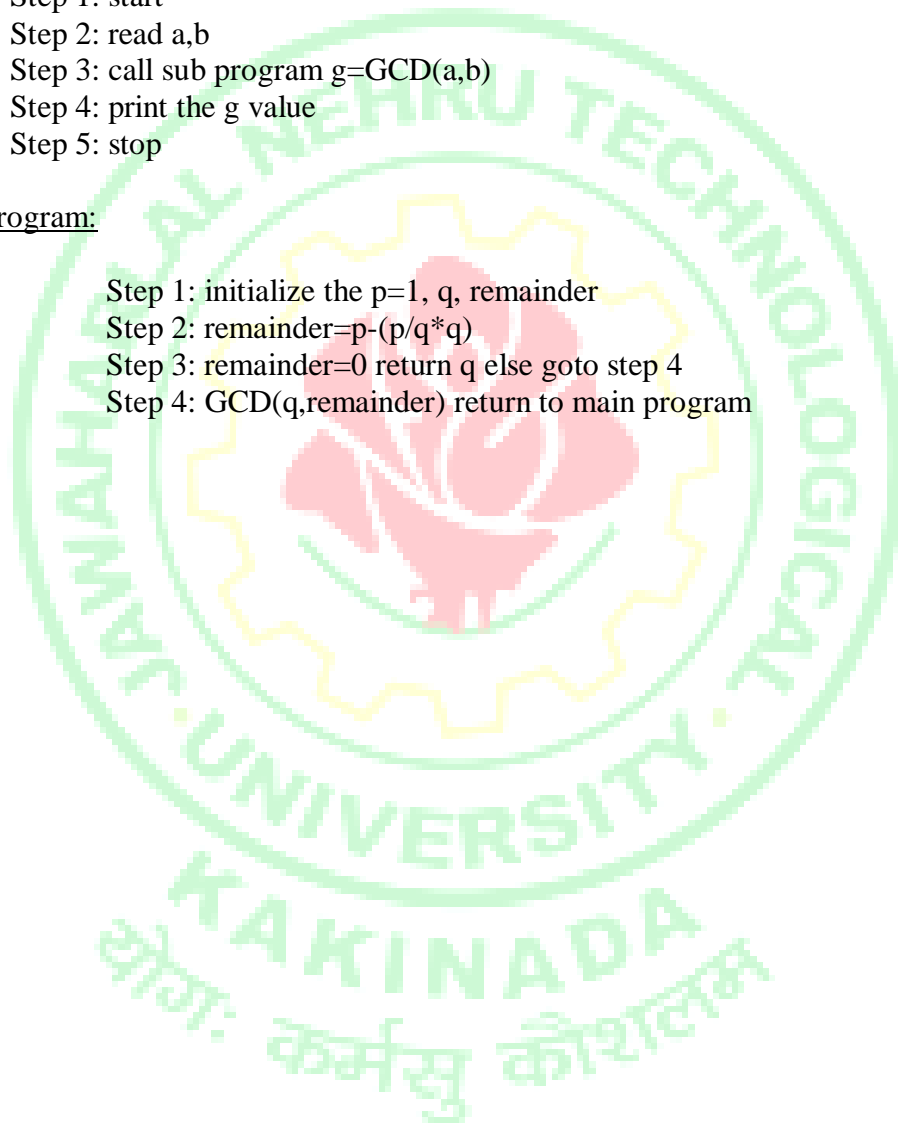
Sub program:

Step 1: initialize the p=1, q, remainder

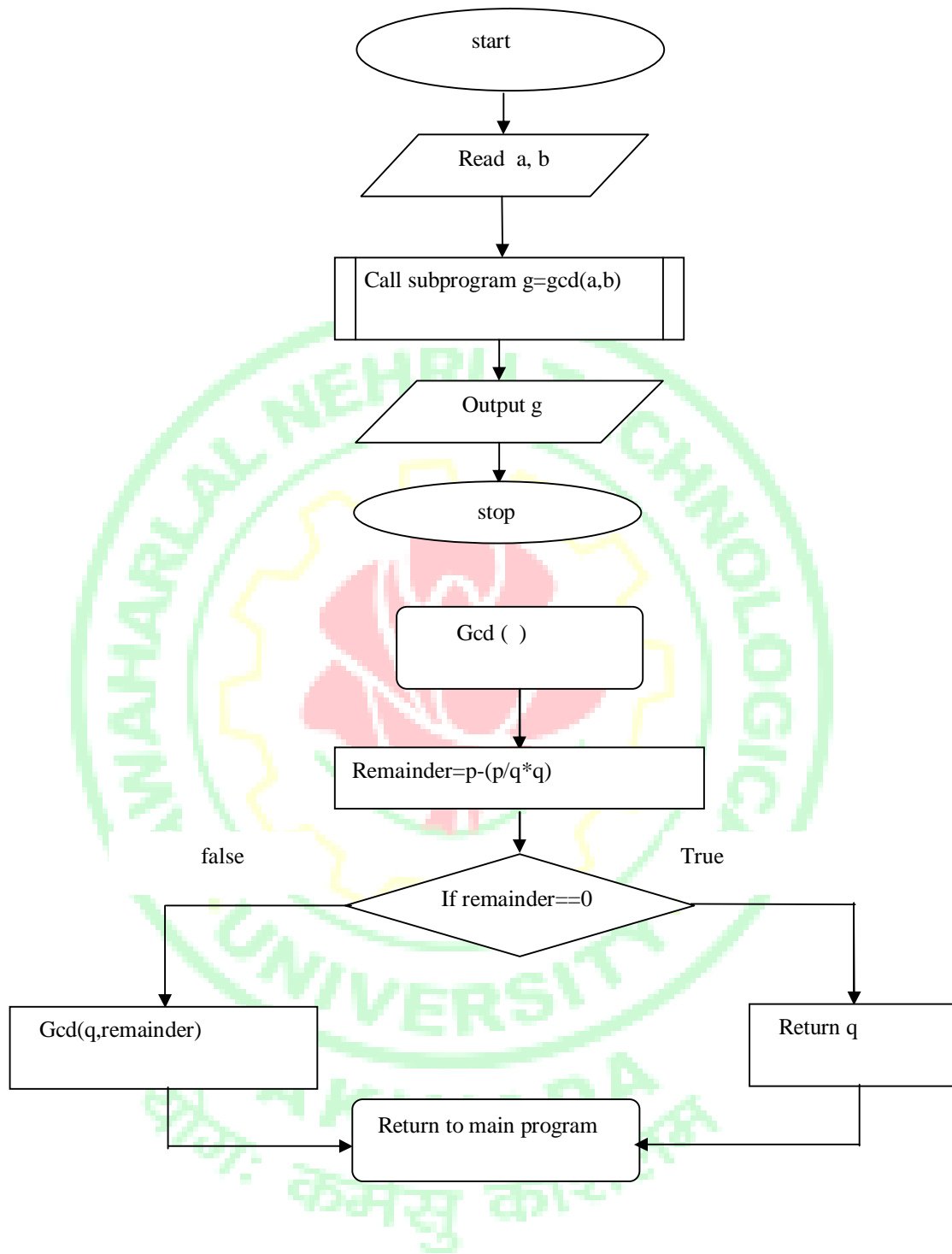
Step 2: remainder = $p - (p/q * q)$

Step 3: remainder = 0 return q else goto step 4

Step 4: GCD(q,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
2. enter the two numbers whose gcd is to be found:36,54
GCD of a,b is : 18
3. enter the two numbers whose gcd is to be found:11,13
GCD of a,b is : 1

Conclusion:

The program is error free

VIVA QUESATIONS:

1)What is meaning of GCD ?

Ans:

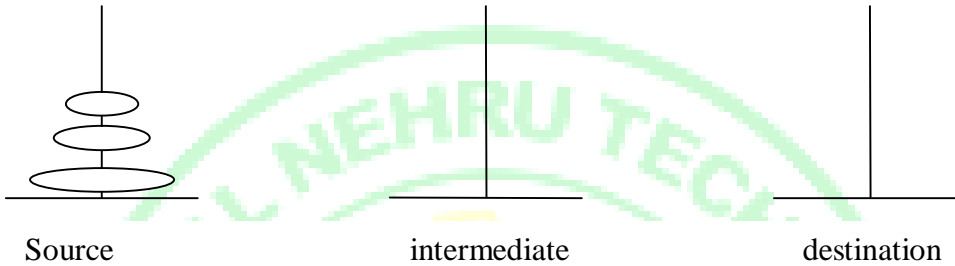
GCD means Greatest Common Divisor. i.e the highest number which divides the given number

Exercise : 10

- c) To solve the towers of Hanoi problem by using the recursive function

Description:

Towers of Hanoi problem means we have three towers



Here source ,intermediate and destination are the three towers. We have to transfer all the disks from source to destination towers. Here the restriction is not to place a big disk on smaller one . for this we use intermediate tower. Finally the arrangements in the destination tower must be as same as the disks in the source tower at first.

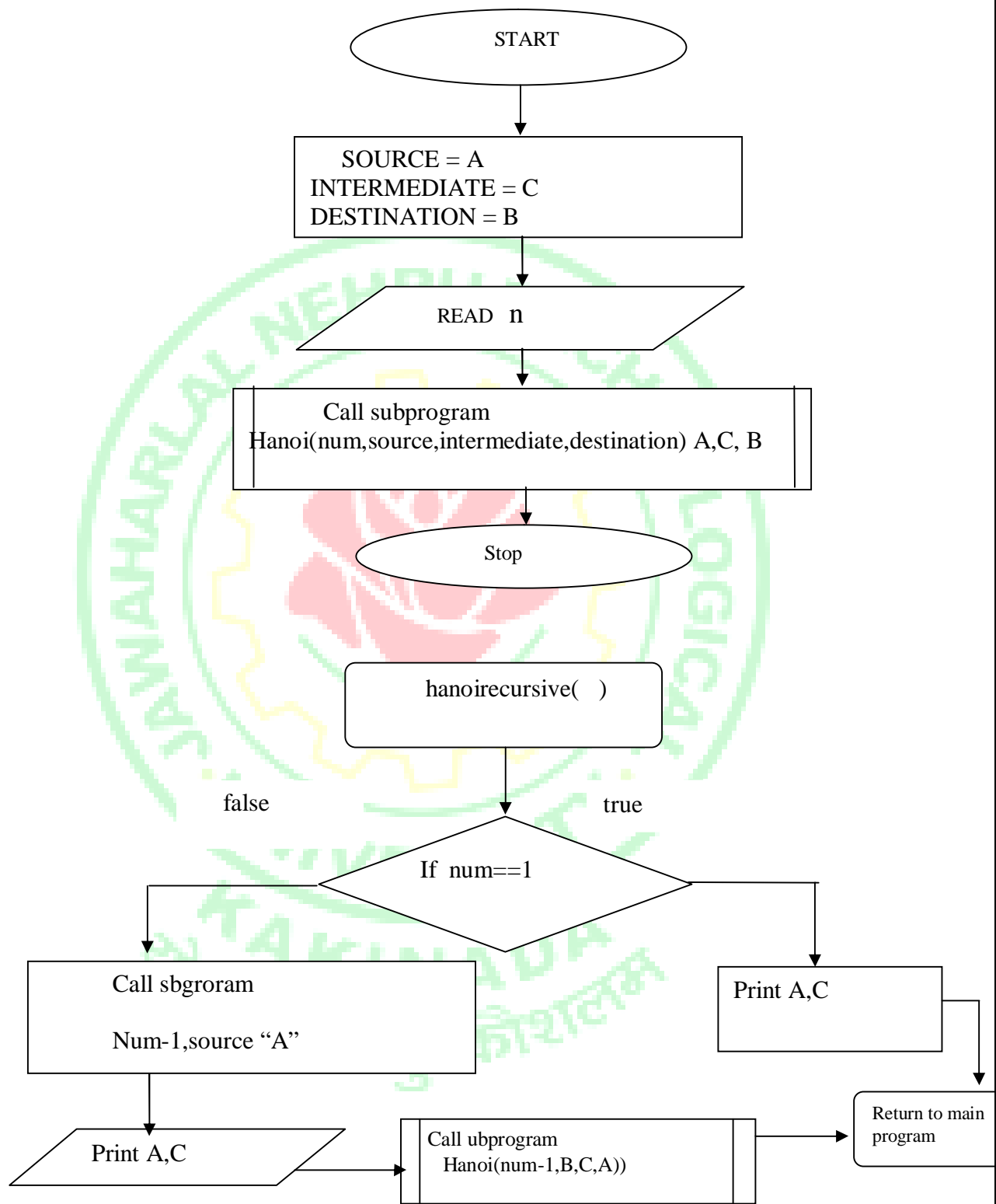
Algorithm: main program

- Step 1: start
- Step 2: initialize the source=a, intermediate=c, destination = d
- Step 3: read n
- Step 4: call the sub program Hanoi recursion (n value,a ,b, c)
- Step 5: stop

Sub program:

- Step 1: if $n == 1$ call the sub program Hanoi recursion (num-1, a, c, b)
- Step 2: print the output from a to b
- Step 3: call the sub program Hanoi recursion(num-1, b, c, a)
- Step 4: return to main program

Flowchart:



Program:

```

#include<stdio.h>
#include<conio.h>
void Hanoirecursion(int num,char ndl1,char ndl2,char ndl3)
{
    if(num==1)
    
```

```

    {
        printf("Move top disk from needle %c to needle %c",ndl1,ndl2);
        return;
    }
    Hanoirecursion(num-1,ndl1,ndl3,ndl2);
    printf("Move top dis from needle %c to needlle %c",ndl1,ndl2);
    Hanoirecursion(num-1,ndl3,ndl2,ndl1);
}

void main()
{
    int no;
    clrscr();
    printf("Enter the no. of disk to be transferred:");
    scanf("%d",&no);
    if(no<1)
        printf("\n There's nothing to move");
    else
        printf("\n recursive");
    Hanoirecursion(no,'A','B','C');
    getch();
}

```

Outputs:

1. Enter the no. of disk to be transferred :3
 Move top disk from needle a to needle b
 Move top disk from needle a to needle c
 Move top disk from needle b to needle c
 Move top disk from needle a to needle b
 Move top disk from needle c to needle a
 Move top disk from needle c to needle b
 Move top disk from needle a to needle b

Conclusion:

The program is error free

VIVA QUESATIONS:**1) What is purpose of towers of Hanoi ?**

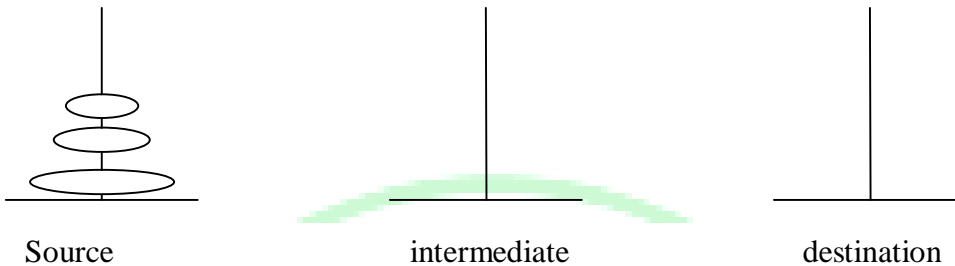
Ans: We have to transfer all the disks from source to destination towers. Here the restriction is not to place a big disk on smaller one . for this we use intermediate tower. Finally the arrangements in the destination tower must be as same as the disks in the source tower at first.

Objective :11

To solve the towers of Hanoi problem by using the non recursive function

Description:

Towers of Hanoi problem means we have three towers



Here source ,intermediate and destination are the three towers. We have to transfer all the disks from source to destination towers. Here the restriction is not to place a big disk on smaller one . for this we use intermediate tower. Finally the arrangements in the destination tower must be as same as the disks in the source tower at first.

Algorithm:

Step 1: start
 Step 2: declare the no
 Step 3: read the no value
 Step 4: if (no<1) Print nothing to move
 Else Print nonrecursion
 Step 5:Hanoi non recursion(no, 'A', 'B', 'C')
 Step 6:stop
 Sub program:
 Step 1: Declare num,sndl,indl,dndl,
 stkn[],stksndl[],stkndnl [],stkadd[],
 temp,top,add
 Step 2: declare the top=NULL
 Step 3: one:
 If(num==1)then
 Print the out put value
 Goto four
 Step 4: two:
 Top=top+1
 Stkn[top]=num
 Stksndl[top]=sndl
 Stkindl[top]=indl
 Stkdndl[top]=dndl
 Stkadd[top]=3
 Num=num-1
 Sndl=sndl
 Temp=indl
 Indl=dndl
 Dndl=temp
 Goto one. Goto step 3
 Step 5:

Three:

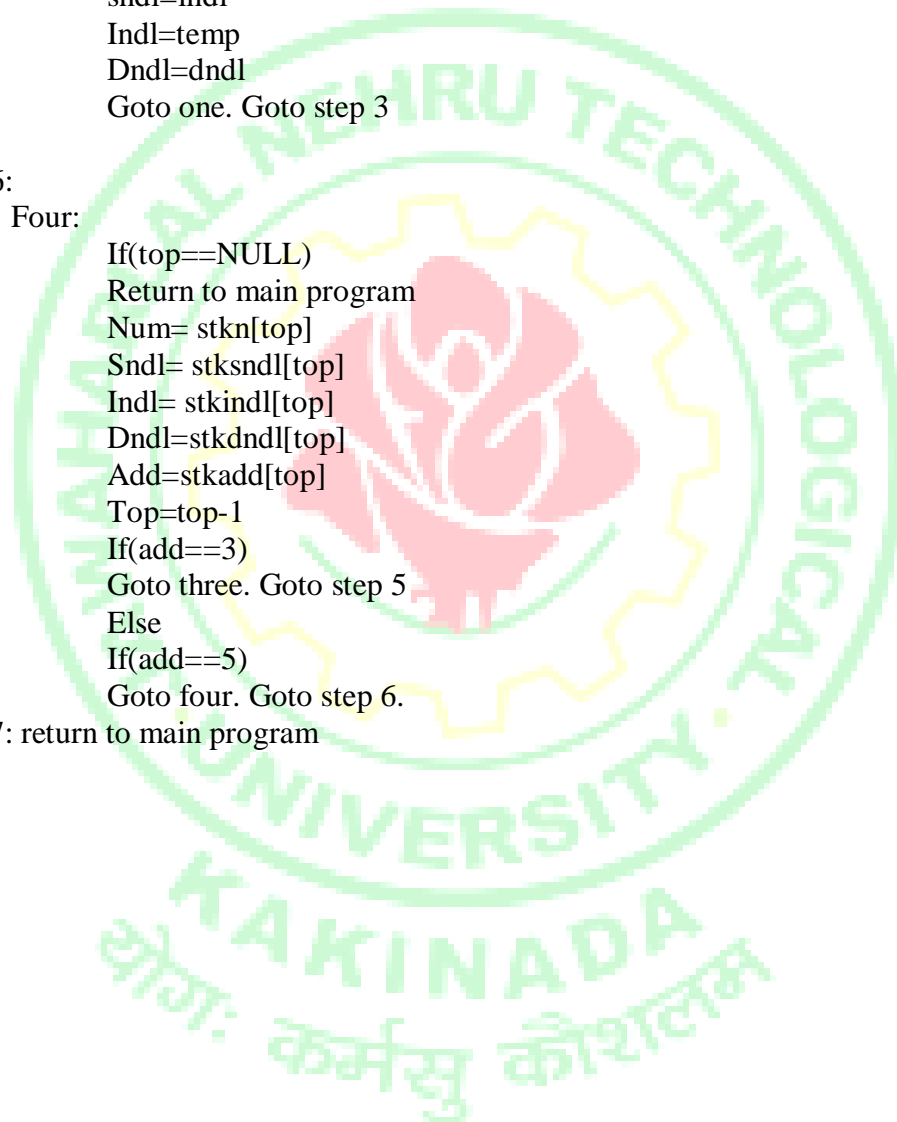
```
Print the output
Top=top+1
Stkn[top]=num
Stksndl[top]=sndl
Stkindl[top]=indl
Stkdndl[top]=dndl
Stkadd[top]=5
Num=num-1
temp=sndl
sndl=indl
Indl=temp
Dndl=dndl
Goto one. Goto step 3
```

Step 6:

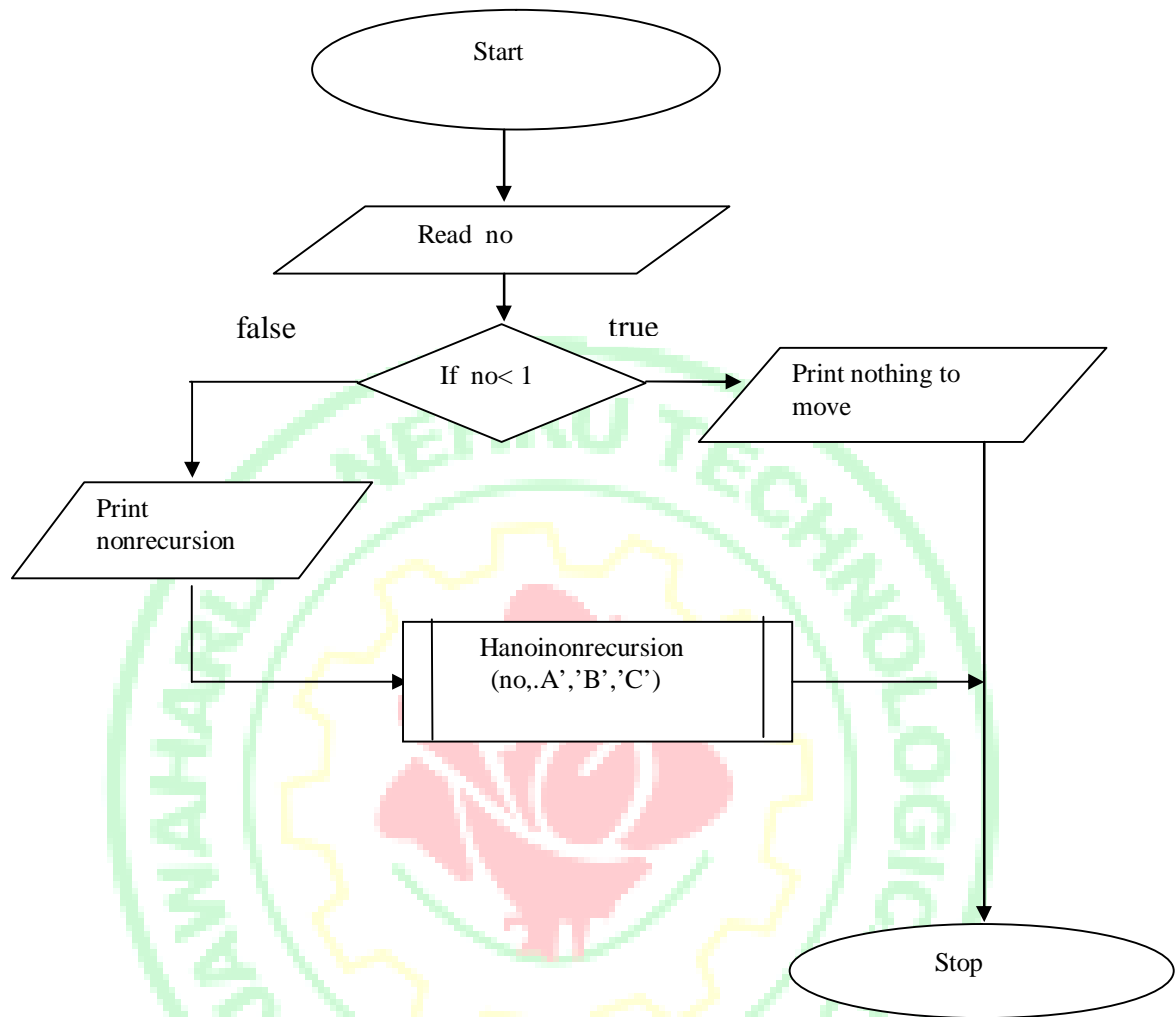
Four:

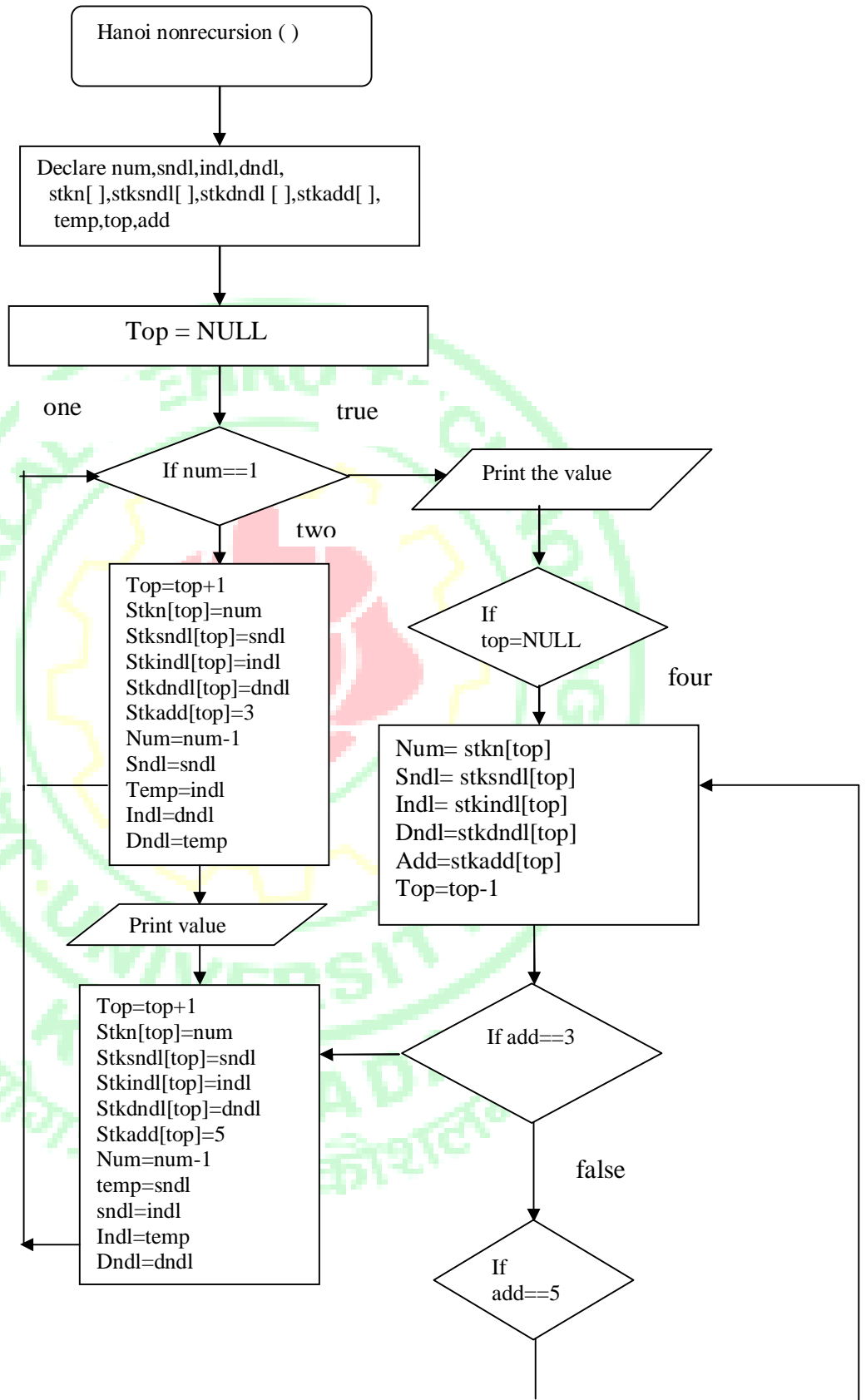
```
If(top==NULL)
Return to main program
Num= stkn[top]
Sndl= stksndl[top]
Indl= stkindl[top]
Dndl=stkdndl[top]
Add=stkadd[top]
Top=top-1
If(add==3)
Goto three. Goto step 5
Else
If(add==5)
Goto four. Goto step 6.
```

Step 7: return to main program



Flow chart:





Program:

```
#include<stdio.h>
#include<conio.h>
void Hanoi(int num,char sndl,char indl,char dndl)
{
char stkn[100],stksndl[100],stkindl[100],stk dndl[100],stkadd[100],temp;
int top,add;
top=NULL;
one:
if(num==1)
{
printf("\n Move top disk from needle %c to needle %c",sndl,dndl);
goto four;
}
two:
top=top+1;
stkn[top]=num;
stksndl[top]=sndl;
stkindl[top]=indl;
stk dndl[top]=dndl;
stkadd[top]=3;
num=num-1;
sndl=sndl;
temp=indl;
indl=dndl;
dndl=temp;
goto one;
three:
printf("\n Move top disk from needle %c to needle %c",sndl,dndl);
top=top+1;
stkn[top]=num;
stksndl[top]=sndl;
stkindl[top]=indl;
stk dndl[top]=dndl;
stkadd[top]=5;
num=num-1;
temp=sndl;
sndl=indl;
indl=temp;
dndl=dndl;
goto one;
four:
if(top==NULL)
return;
num=stkn[top];
sndl=stksndl[top];
indl=stkindl[top];
dndl=stk dndl[top];
add=stkadd[top];
top=top-1;
```

```
    if(add==3)
        goto three;
    else if(add==5)
        goto four;
    }
void main()
{
    int no;
    clrscr();
    printf("Enter the no. of diss to be transferred:");
    scanf("%d",&no);
    if(no<1)
        printf("\n There's nothing to move");
    else
        printf("\n nonrecursive");
        Hanoiinrecursion(no,'A','B','C');
    getch();
}
```

Output:

1. Enter the no. of diss to be transferred:3
nonrecursive
Move top disk from needle A to needle C
Move top disk from needle A to needle B
Move top disk from needle C to needle B
Move top disk from needle A to needle C
Move top disk from needle B to needle A
Move top disk from needle B to needle C
Move top disk from needle A to needle C

Conclusion: The program is error freed

VIVA QUESATIONS:

1) **What is purpose of towers of Hanoi ?**

Ans: We have to transfer all the disks from source to destination towers. Here the restriction is not to place a big disk on smaller one . for this we use intermediate tower. Finally the arrangements in the destination tower must be as same as the disks in the source tower at first.

2) **What is an array ?**

Ans: An array is a sequence of memory location of same data type.

Exercise : 11

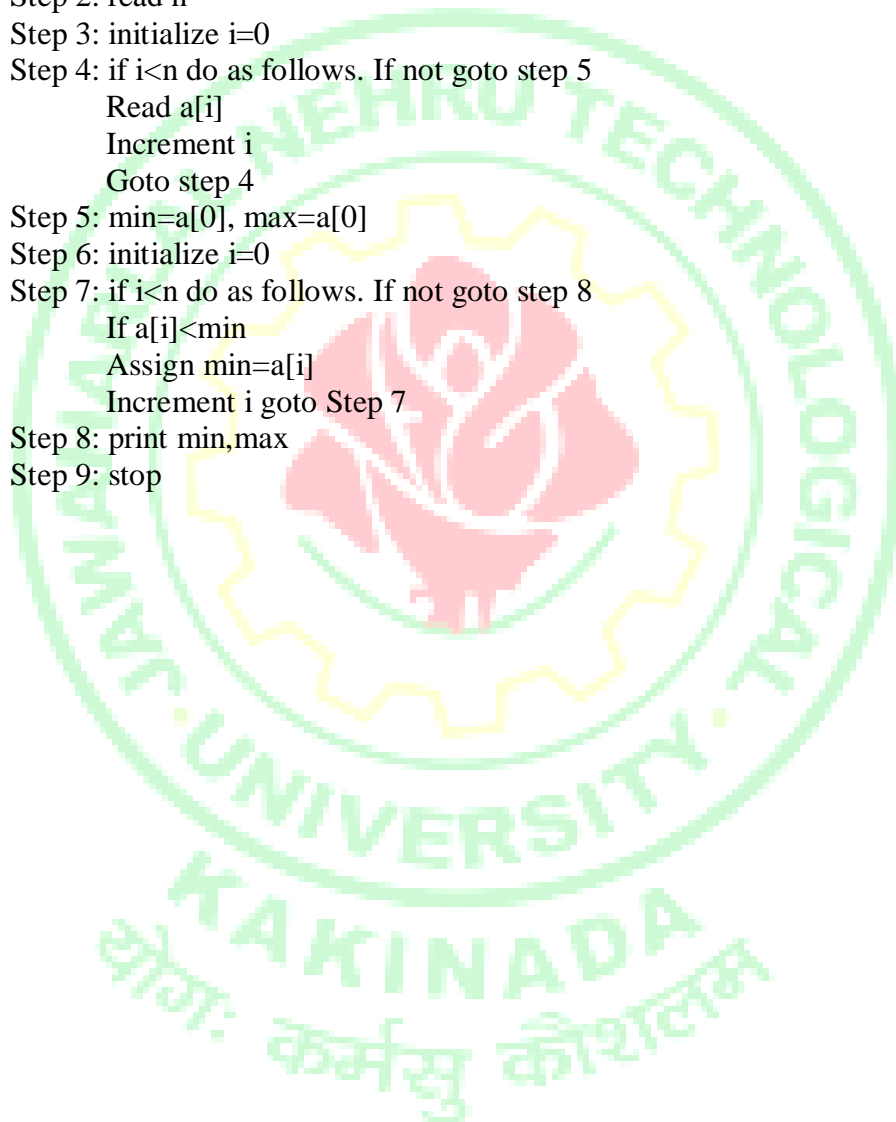
- a) To find both the largest and smallest number in a list of integers

Description:

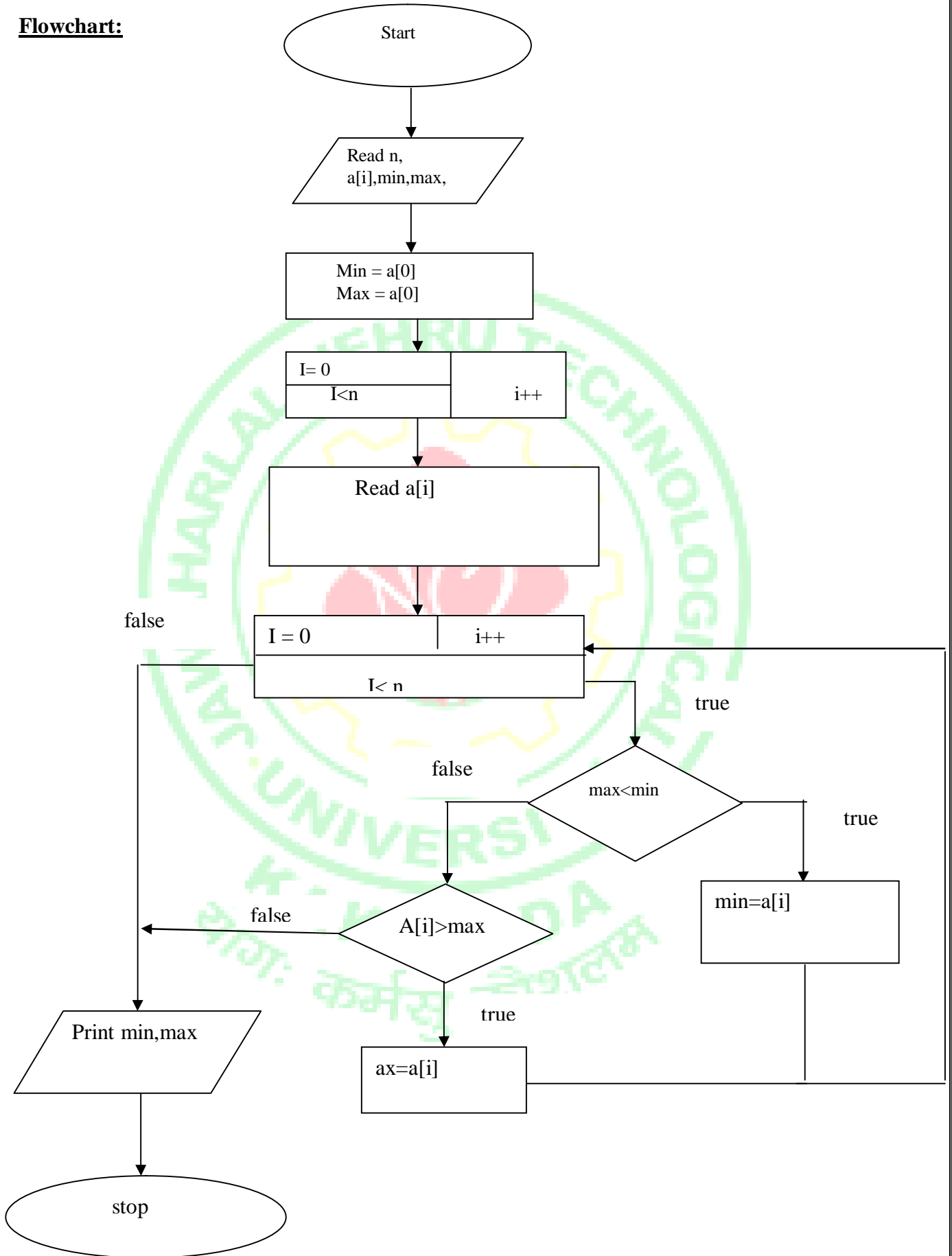
This program contains n number of elements, in these elements we can find the largest and smallest numbers and display these two numbers

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: min=a[0], max=a[0]
Step 6: initialize i=0
Step 7: if i<n do as follows. If not goto step 8
 If a[i]<min
 Assign min=a[i]
 Increment i goto Step 7
Step 8: print min,max
Step 9: stop



Flowchart:



Program:

```
#include<stdio.h>
void main()
{
    int a[10],i,n,min,max;
    clrscr();
    printf("enter the array size:");
    scanf("%d",&n);
    printf("Enter the elements of array");
    for(i=0;i<n;i++) // read the elements of an array
    scanf("%d",&a[i]);
    min=a[0];
    max=a[0];
    for(i=0;i<n;i++)// read the elements of an array
    {
        if(a[i]<min)// check the condition for minimum value
        min=a[i];
        if(a[i]>max)//check the condition for maximum value
        max=a[i];
    }
    printf("maximum value is:%d\n",max);
    printf("minimum value is:%d\n",min);
    getch();
}
```

Output:

1. enter the array size:4
 Enter the elements of array 36 13 2 45
 maximum value is:45
 minimum value is:2
2. enter the array size:5
 Enter the elements of array 6 2 1 3 8
 maximum value is:8
 minimum value is:1
3. enter the array size:5
 Enter the elements of array-6 9 -9 2 5
 maximum value is:9
 minimum value is:-9

conclusion: the program is error free

VIVA QUESATIONS:

1) What is an array ?

Ans: The collection of similar elements is called array

2) How many types of arrays are there ?

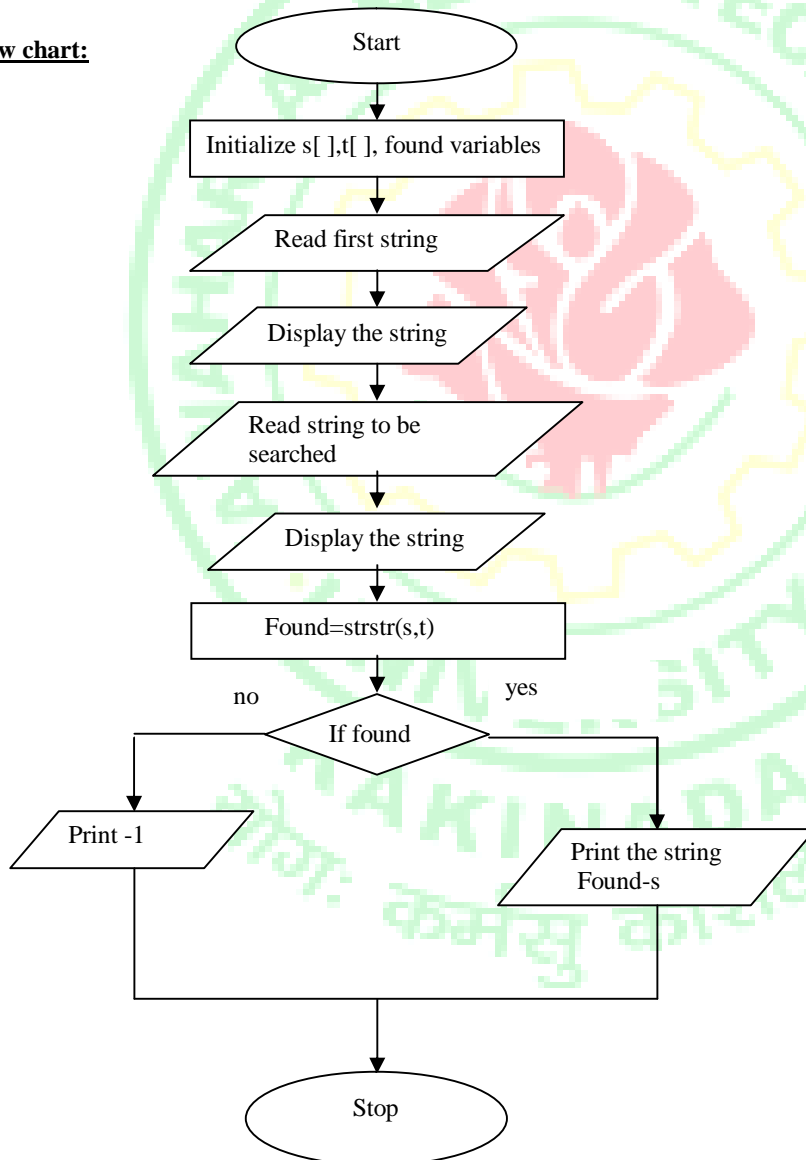
Ans: Three types. They are one dimensional ,two dimensional and multi dimensional arrys

Exercise : 11

- b) Program that displays 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();

    puts("Enter the first string: ");
    gets(s);
    puts("Enter the string to be searched: ");
    gets(t);

    found=strstr(s,t);
    if(found)
        printf("Second String is found in the First String at %d position.\n",found-s);
    else
        printf("-1");
    getch(); }

```

Output:

```

1.enter the first string:
kali
Enter the string to be searched:
li
second string is found in the first string at2position
2.enter the first string:
nagaraju
Enter the string to be searched:
raju
second string is found in the first string at4position
3.enter the first string:
nagarjuna
Enter the string to be searched:
ma
-1

```

Conclusion: The program is error free

VIVA QUESATIONS:

- 1) What is the difference between printf() and puts() ?
 Ans: puts() is used to display the string at a time and it doesn't take any integers values but printf() takes any values as defined by the user
- 2) define pointer variable ?
 Ans: pointer variables are defined as variable that contain the memory addresses of data or executable code.
- 3) What is use of the strcmp() function ?
 Ans: This function compares two strings character by character and returns a value 0 if both strings are equal and non zero value if the strings are different.

Exercise : 11

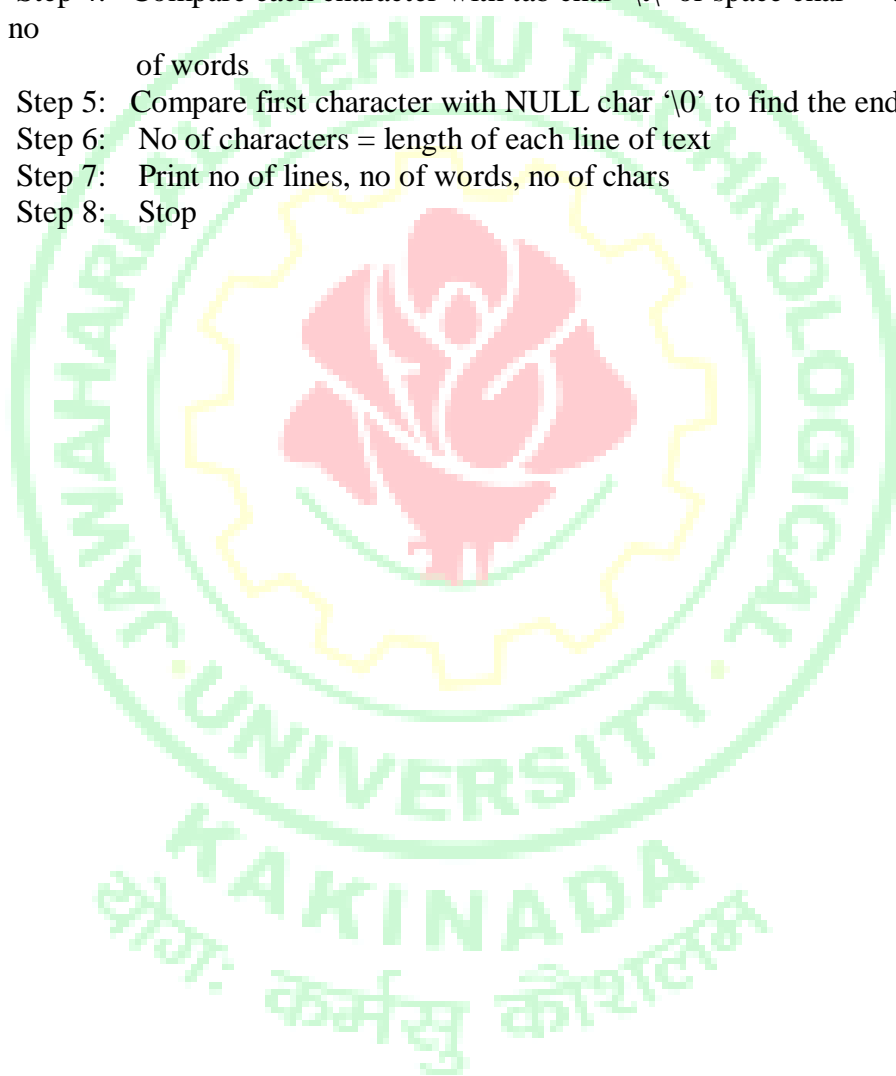
- c) To count the lines, words & characters in a given text

Description:

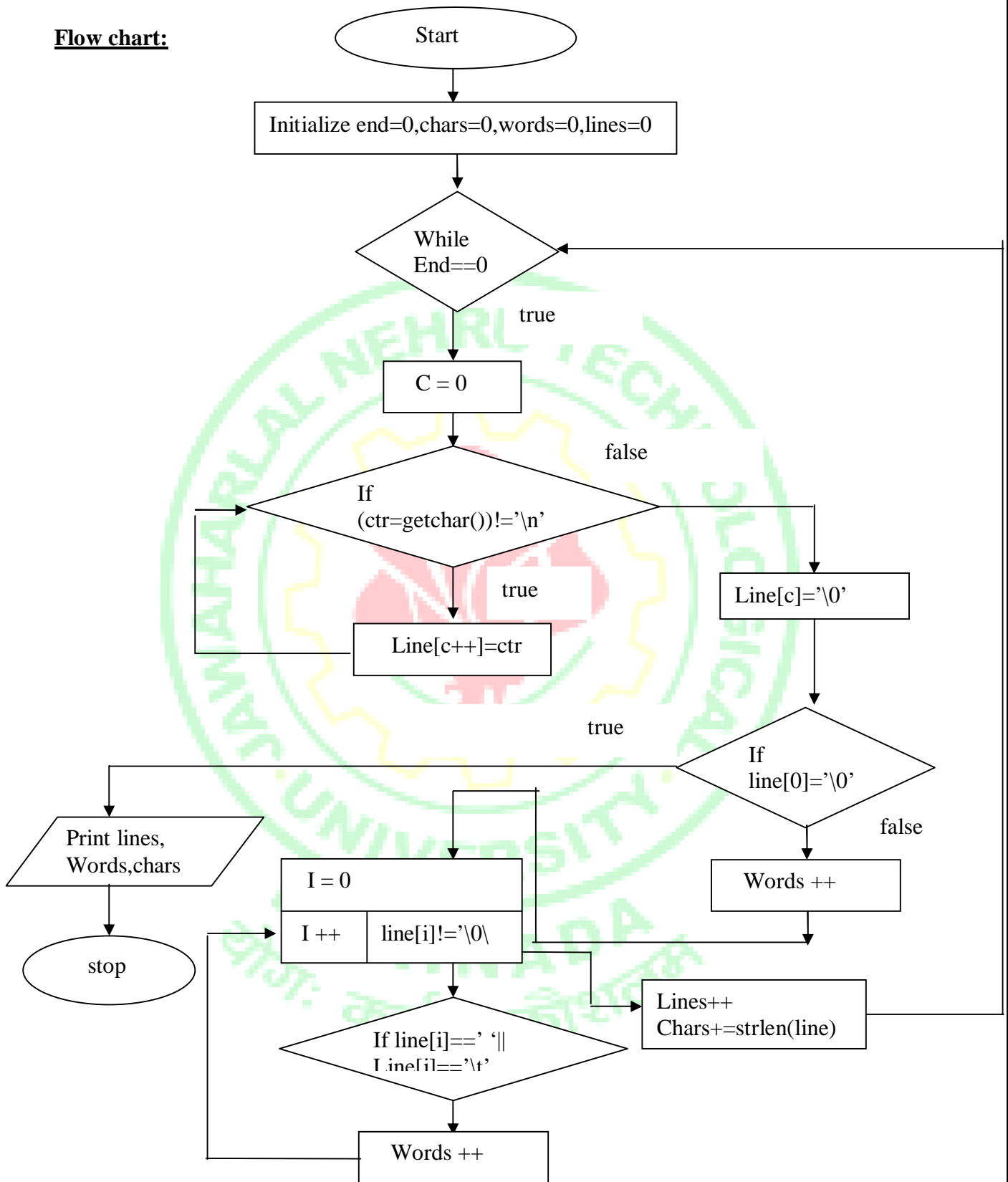
In this program we have to count the no of lines, no of words and no of characters in a given program or given text by using the string function

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>
main()
{
    char line[81], ctr;
    int i,c,
        end = 0,
        characters = 0,
        words = 0,
        lines = 0;
    printf("KEY IN THE TEXT.\n");
    printf("GIVE ONE SPACE AFTER EACH WORD.\n");
    printf("WHEN COMPLETED, PRESS 'RETURN'.\n\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);
}

```

Output

1.KEY IN THE TEXT.

GIVE ONE SPACE AFTER EACH WORD.

WHEN COMPLETED, PRESS 'RETURN'.

Admiration is a very short-lived passion.

Admiration involves a glorious obliquity of vision.

Always we like those who admire us but we do not

like those whom we admire.

Fools admire, but men of sense approve.

Number of lines = 5

Number of words = 36

Number of characters = 205

Conclusion: The program is error free

VIVA QUESATIONS:

1) What is use of strlen() ?

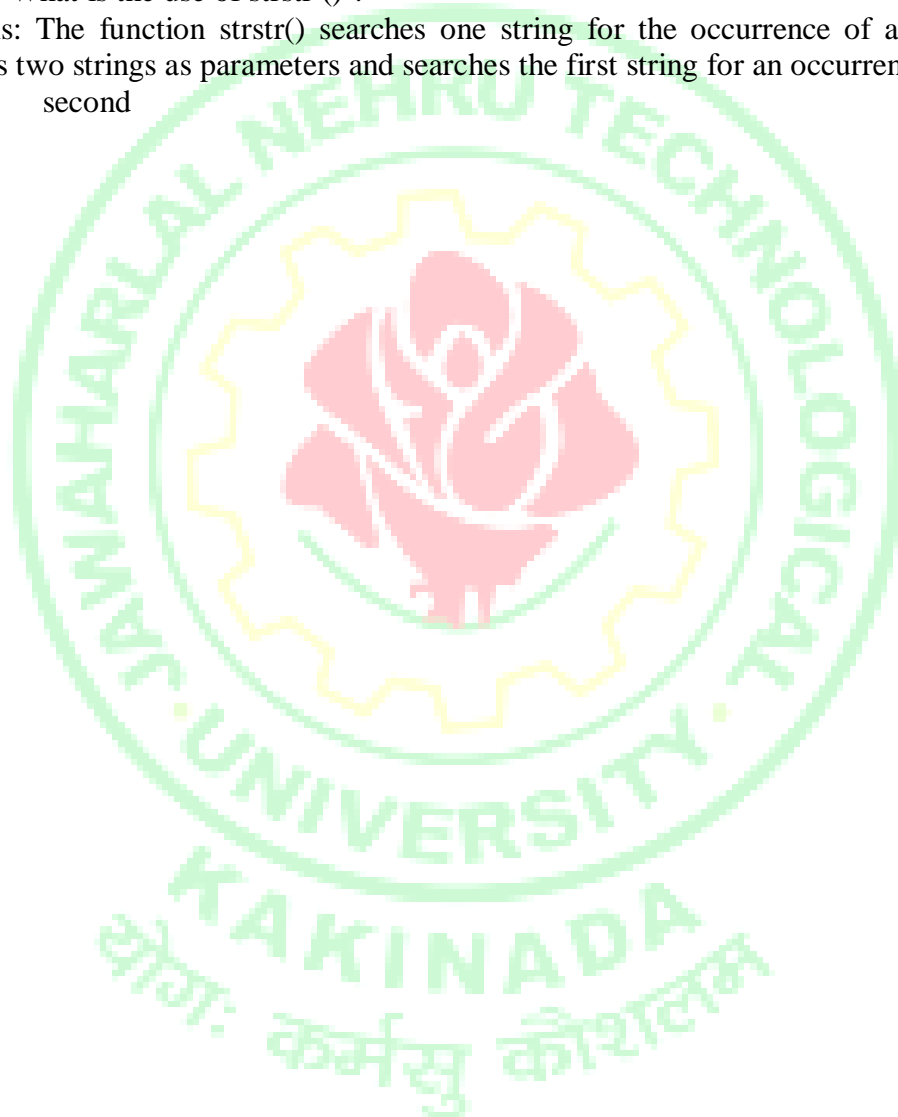
Ans: to read a string length

2) what is the use of getc() function ?

Ans: To read the character one by one.

3) What is the use of strstr () ?

Ans: The function strstr() searches one string for the occurrence of another. It accepts two strings as parameters and searches the first string for an occurrence of the second



Exercise : 12

- a) To generate Pascal's triangle

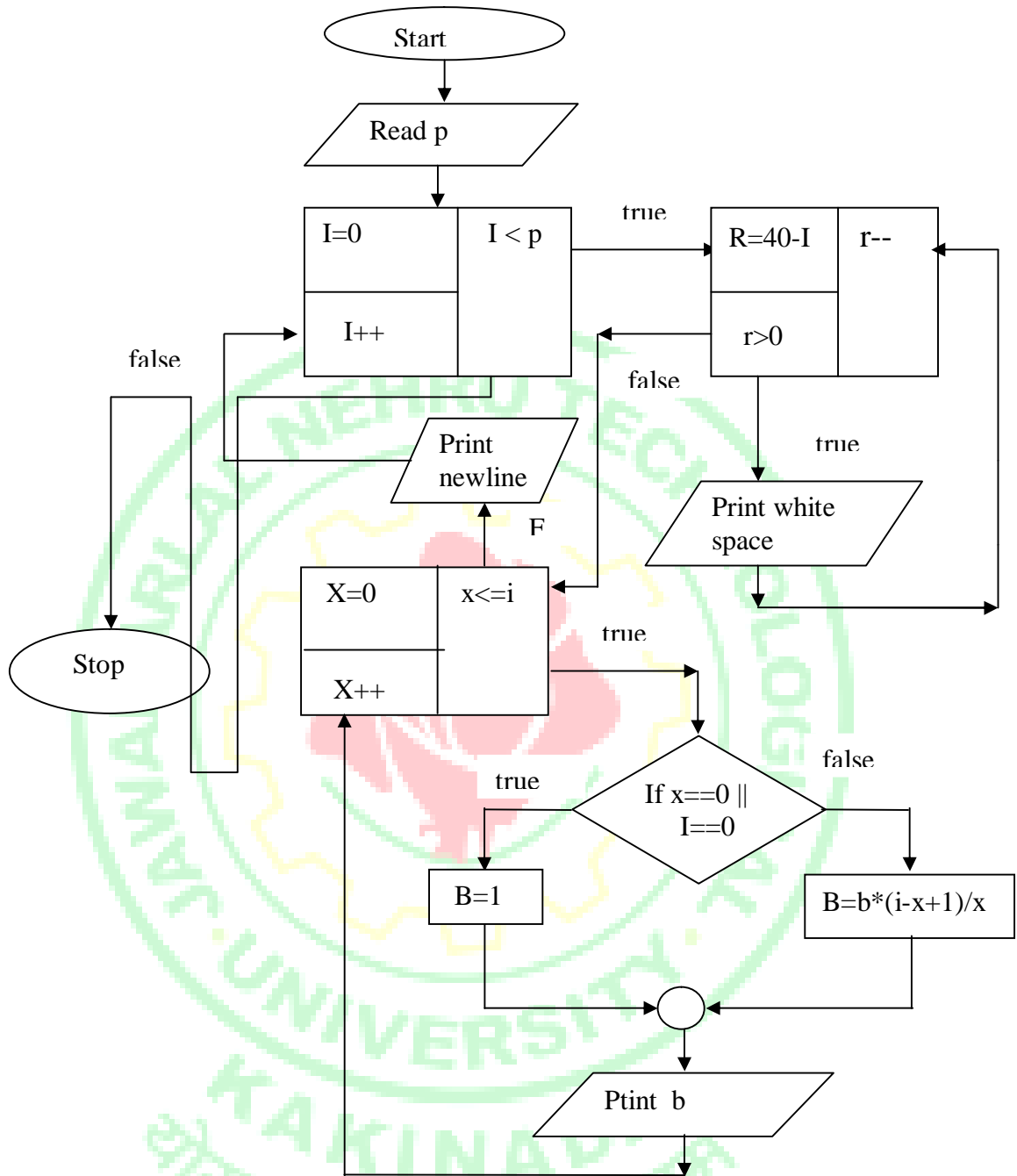
Description:

Pascal's triangle which is used for a coefficient in the equation in polynominals.

Alogrithm:

- Step 1: Start
Step 2: Initialize m=0
Step 3: Read n
Step 4: If m<n goto step 5.if not goto step 12
Step 5: initialize i=40-m
Step 6: If i>0 is true do as follows. If not goto step 7
 i. print white space
 ii. decrement i
 iii. goto Step 6
Step 7: Initialize j=0
Step 8: If j=m do as follows. If not goto Step 10
 i) if(j==0||m==0)
 ii) Initialize b=1 if not b=b*(m-j+1)/j
 iii) Print white space, b .
 iv) Goto Step 9
Step 9: increment j, goto Step 8
Step 10: print new line control
Step 11: increment m, goto step 4
Step 12: Stop

Flow chart:



Program:

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i,p,r,x,binom=1;
    clrscr();
    printf("enter the how many lines to print");
    scanf("%d",&p);
    i=0;
    while(i<p) // check the condition
    {
        for(r=40-i;r>0;r--) // perform the looping operation until 0
        printf(" ");
        for(x=0;x<=i;x++)
        {
            if((x==0)||(i==0)) // check the condition
            binom=1;
            else
            binom=binom*(i-x+1)/x;
            printf("%d",binom);
            printf(" ");
        }
        printf("\n");
        i++;
    }
    getch();
}

```

Output:

1. enter the how many lines to print 5

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

```

2. enter the how many lines to print 3

```

1
1 1
1 2 1

```

Conclusion: the program is error free

VIVA QUESATIONS:

1) What is meant by Pascal's triangle ?

Ans: Pascal's triangle which is used for a coefficient in the equation in polynomials

2) define structure ?

Ans: A structure in c is a heterogenous user defined data type. A structure may contain different data types. It groups variables into a single entity.

Exercise : 12

Program:

```

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

void main()
{
    int num,i,y,x=35;
    clrscr();
    printf("\nEnter the number to generate the pyramid:\n");
    scanf("%d",&num);

    for(y=0;y<=num;y++)
    {
        /*(x-coordinate,y-coordinate)*/
        gotoxy(x,y+1);

        /*for displaying digits towards the left and right of zero*/
        for(i=0-y;i<=y;i++)
            printf("%3d",abs(i));
        x=x-3;
    }
    getch();
}

```

Output:

```

1.enter the number:    0
4                      1 0 1
                      2 1 0 1 2
                      3 2 1 0 1 2 3
                      4 3 2 1 0 1 2 3 4

```

```

2.enter the number:    0
3                      1 0 1
                      2 1 0 1 2
                      3 2 1 0 1 2 3

```

Conclusion: The program is error free

VIVA QUESATIONS:

1) What is the use of dot operator in structures ?

Ans: The use of dot(.) operator to access the members of a structure independently. The dot operator connects a member with the structure variable.

2) Define unions ?

Ans: A union is a data type in c which allows the overlay of more than one variable in the same memory area.

Exercise : 13

- a) To read in two numbers x and n and then compute the sum of this geometric progression $1+x+x^2+x^3+\dots+x^n$

Description:

In this program we have to read the two numbers and the calculate the sum of this geometric progression in above mention .

Algorithm:

Step 1: Start

Step 2: read values of x and n, sum=1, i=1

Step 3: check for n & X

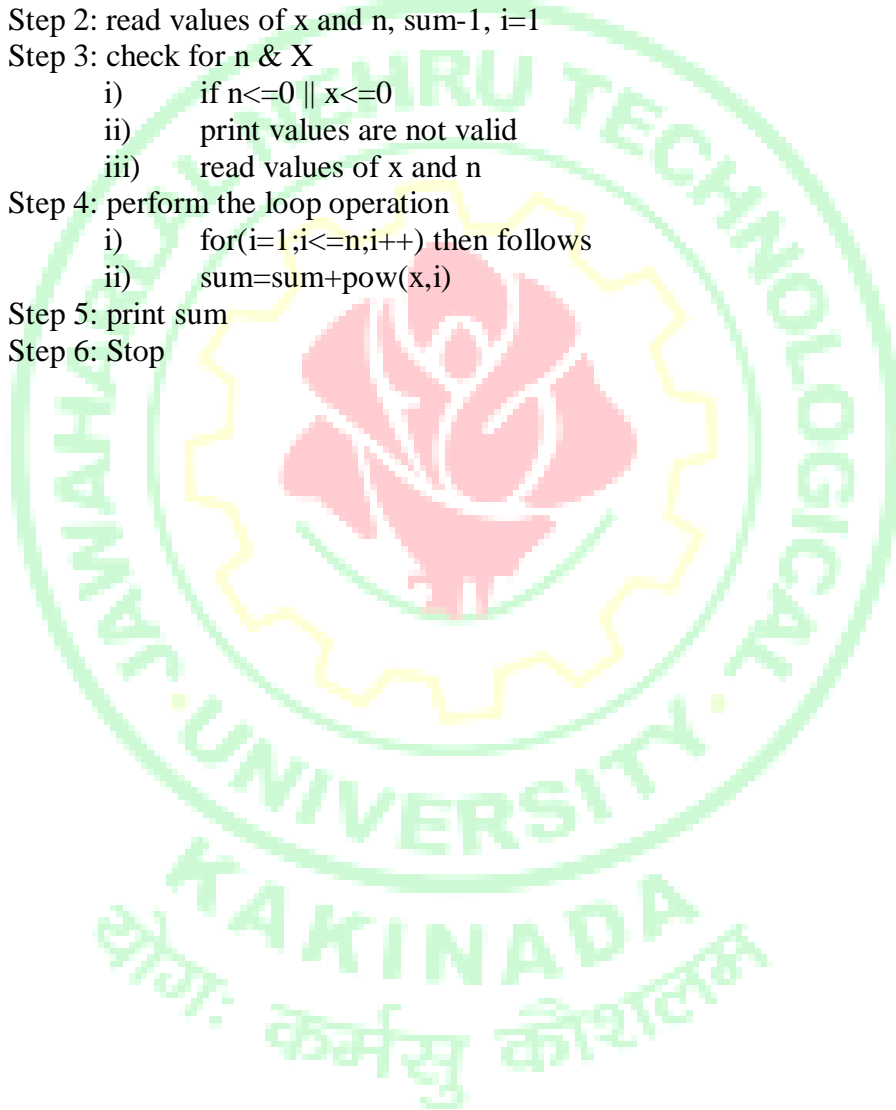
- i) if $n \leq 0$ || $x \leq 0$
- ii) print values are not valid
- iii) read values of x and n

Step 4: perform the loop operation

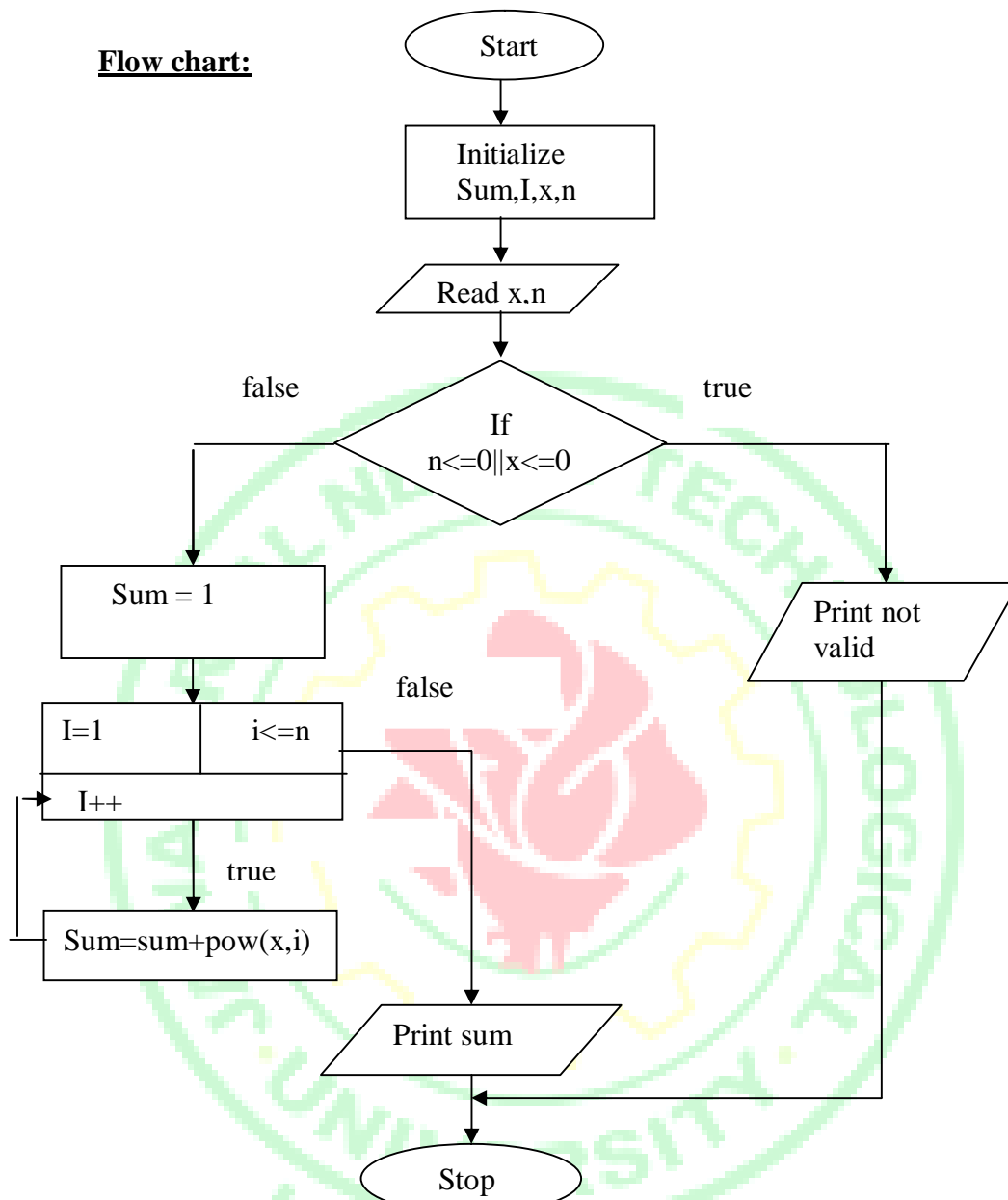
- i) for($i=1; i \leq n; i++$) then follows
- ii) $sum = sum + pow(x, i)$

Step 5: print sum

Step 6: Stop



Flow chart:



Program:

```

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

void main()
{
    int s_sum,i,x,n;

    clrscr();
    printf("Enter the values for x and n:");
    scanf("%d %d",&x,&n);

    if(n<=0 || x<=0)
    {
        printf("Value is not valid\n");
    }
    else
    {
        printf("Value is valid\n");
        s_sum=1;
        for(i=1;i<=n;i++)
        {
            s_sum=s_sum+pow(x,i);
        }
        printf("Sum of series=%d\n",s_sum);
    }
    getch();
}

```

Output:

1. Enter the values for x and n:2
3
Value is valid
Sum of series=15
2. Enter the values for x and n:4
9
Value is valid
Sum of series=21845
3. Enter the values for x and n:0
1
Value is not valid

Conclusion: the program is error free

VIVA QUESATIONS:

1) what are the difference between structures and unions ?

Ans: Here the major difference is with in the structure all elements must be allocated memory. But in union highest memory allocation must be allocated the all these elements.

Exercise : 13

- b) Write a C function to read in two numbers, x and n(no. of terms), and then compute sin(x) and cos(x).

Description

In this we calculate sin and cos values by reading x value and no.of terms 'n'.

Program**Sin series**

```
main()
{
    float x,x1,sum=0.0;
    long int fact(int);
    int i,n;
    printf("\nEnter any number : ");
    scanf("%f",&x);
    printf("\nEnter no.of terms ");
    scanf("%d",&n)
    x1 = x*3.1412/180;
    for(i=1;i<n;i+=2)
    {
        sum = sum +(sign*pow(x1,i)/fact(i));
        sign * = -1;
    }
    printf("\nThe sin(%f) is %f",x,sum);
}
```

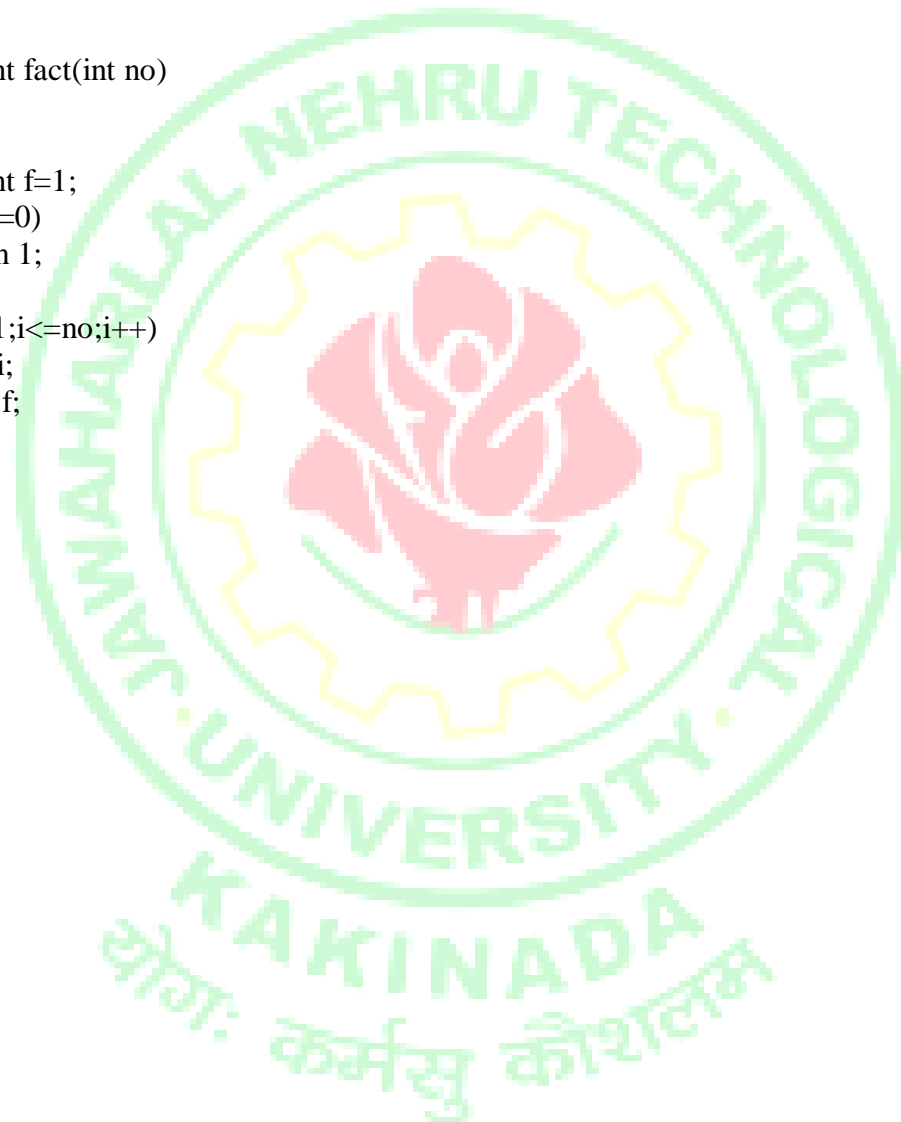
```
long int fact(int no)
{
    int i;
    long int f=1;
    if(no==0)
        return 1;
    else
        for(i=1;i<=no;i++)
            f=f*i;
    return f;
}
```

cos series

```
main()
{
    float x,x1,sum=0.0;
    long int fact(int);
    int i,n;
    printf("\nEnter any number : ");
```

```
scanf("%f",&x);
printf("\nEnter no.of terms ");
scanf("%d",&n);
x1 = x*3.1412/180;
for(i=0;i<n;i+=2)
{
    sum = sum +(sign*pow(x1,i)/fact(i));
    sign * = -1;
}
printf("\nThe cos(%f) is %f",x,sum);
}
```

```
long int fact(int no)
{
    int i;
    long int f=1;
    if(no==0)
        return 1;
    else
        for(i=1;i<=no;i++)
            f=f*i;
    return f;
}
```



Exercise : 14

- a) Function to exchange value of two integers using passing by address.

Description

This program interchange of values of two variable using functions and sending addresses.

Algorithm

- Step 1 : start
Step 2 : read a,b
Step 3 : display “before calling function a,b are” a,b
Step 4 : call exchange
Step 5: display “after calling function a,b are “ ,a,b
Step 6 : stop

Algorithm for exchange :

- Step 1 : start
Step 2 : assign t = *x;
Step 3 : assign *x =*y
Step 4 : assign *y = t
Step 5 : display “in the function a,b “ x,y
Step 6: stop

Program

```
main()
{
int a,b;
void swap(int *, int *);
printf("\nEnter a,b values :");
scanf("%d%d",&a,&b);
printf("\nbefore calling function a=%d, b=%d",a,b);
swap(&a,&b);
printf("\nAfter calling function a=%d, b=%d",a,b);
}
void swap(int *x, int *y)
{
int t;
t=*x;
*x=*y;
*y = t;
printf("\nIn the function a=%d , b=%d",a,b);
}
```

Exercise : 14

b) Program which explains the use of dynamic arrays.

Algorithm

Step 1 : start
Step 2 : read n
Step 3 : allocate memory for n values of given type
Step 4 : assign 0 to i
Step 5 : if(i<n) go to step 6 else go to step 7
Step 6 : read a+i values
Step 7 : compute i = i+1 goto step 5
Step 8 : assign 0 to i
Step 9 : display given element *(a+i)
step 10 : i = i+1
step 11 : if (i<n) go to step 9
step 12 : stop

Program

```
#include<alloc.h>

main()
{

int *a,n;
printf("\nEnter no.of values :");
scanf("%d",&n);
a=(int *) malloc(sizeof(int));
printf("\nEnter %d elements :");
for(i=0;i<n;i++)
scanf("%d",(a+i));
printf("\nThe given elements are :");
for(i=0;i<n;i++)
printf("%d\t",*(a+i));
getch();
}
```

Exercise : 15

a) Program to display students information using structures.

Description

This program demonstrates about structures.

Algorithm

Step 1 : start
 Step 2 : read n
 Step 3 : initialize I =0
 Step 4 : if(i<n) go to step 5 otherwise go to step 10
 Step 5 : initialize total to 0
 Step 6 : display “enter student details”
 Step 7 : read student rollno, name, class, marks in 6 subjects
 Step 8 : calculate total = sum of 6 subjects
 Step 9 : compute i = i+1 goto step 4
 Step 10 : initialize i =0
 Step 11 : if(i<n) go to step 12 otherwise go to step 15
 Step 12 : display “Name “ student name
 Display “Roll no “ student rollno
 Display “class “ student class
 Display “total “ student total
 Step 13 : compute i = i+1
 Step 14 : go to step 11
 Step 15 : stop

Program

```

struct student
{
int rno,m[6],total;
char name[25],class[25];
};
struct student s[25];
main()
{
int i,n,j;
printf("\nEnter n value :");
scanf("%d",&n);
for(i=0;i<n;i++)
{
s[i].total =0;
printf("\nEnter %d student details :");
printf("\nEnter name, rollno, class marks in 6 subjects ");
scanf("%",&s[i].rno);
flushall();
gets(s[i].name);
flushall();
gets(s[i].class);
for(j=0;j<6;j++)
scanf("%d",&s[i].m[j]);
s[i].total += s[i].m[j]
}
}

```

```

}
printf("\nthe students details :");
for(i=0;i<n;i++)
{
printf("\nName ",s[i].name);
printf("\nRoll number ",s[i].rollno);
printf("\nclass ",s[i].class)
printf("\ntotal marks :";s[i].total);

}

```

Viva Questions

1. What is a structure?

Structures in C defines the group of contiguous (adjacent) fields, such as records or control blocks. A structure is a collection of variables grouped together under a single name. It provides an elegant and powerful way for keeping related data together.

2. Differentiate between structure and union.

Structure	Union
i. Access Members	
We can access all the members of structure at anytime.	Only one member of union can be accessed at anytime.
ii. Memory Allocation	
Memory is allocated for all variables.	Allocates memory for variable which variable require more memory.
iii. Initialization	
All members of structure can be initialized	Only the first member of a union can be initialized.
iv. Keyword	
'struct' keyword is used to declare structure.	'union' keyword is used to declare union.

v. Syntax	
<pre>struct struct_name { structure element 1; structure element 2; ----- ----- structure element n; }struct_var_nm;</pre>	<pre>union union_name { union element 1; union element 2; ----- ----- union element n; }union_var_nm;</pre>
vi. Example	
<pre>struct item_mst { int rno; char nm[50]; }it;</pre>	<pre>union item_mst { int rno; char nm[50]; }it;</pre>

3. What are nested structures?

Structures within Structures (Nested Structures) :

Structures can be used as structures within structures. It is also called as 'nesting of structures'.

Syntax:

```
struct structure_nm
{
    <data-type> element 1;
    <data-type> element 2;
    -----
    -----
    <data-type> element n;
```



```
struct structure_nm
{
    <data-type> element 1;
    <data-type> element 2;
    -----
    -----
    <data-type> element n;
}inner_struct_var;
}outer_struct_var;
```

Example :

```
struct stud_Res
{
    int rno;
    char nm[50];
    char std[10];

    struct stud_subj
    {
        char subjnm[30];
        int marks;
    }subj;
}result;
```

In above example, the structure stud_Res consists of stud_subj which itself is a structure with two members. Structure stud_Res is called as 'outer structure' while stud_subj is called as 'inner structure.' The members which are inside the inner structure can be accessed as follow : result.subj.subjnm result.subj.marks

Exercise : 15

b) Program to display student details using unions.

Program:

```

union student
{
int rno,m[6],total;
char name[25],class[25];
};

union student s[25];
main()
{
int i,n,j;
printf("\nEnter n value :");
scanf("%d",&n);
for(i=0;i<n;i++)
{
s[i].total =0;
printf("\nEnter %d student details :");
printf("\nEnter name, rollno, class marks in 6 subjects ");
scanf("%",&s[i].rno);
flushall();
gets(s[i].name);
flushall();
gets(s[i].class);
for(j=0;j<6;j++)
scanf("%d",&s[i].m[j]);
s[i].total += s[i].m[j]
}
}
printf("\nthe students details :");
for(i=0;i<n;i++)
{
printf("\nName ",s[i].name);
printf("\nRoll number ",s[i].rollno);
printf("\nclass ",s[i].class)
printf("\ntotal marks :",s[i].total);
}
}

```

Exercise: 16

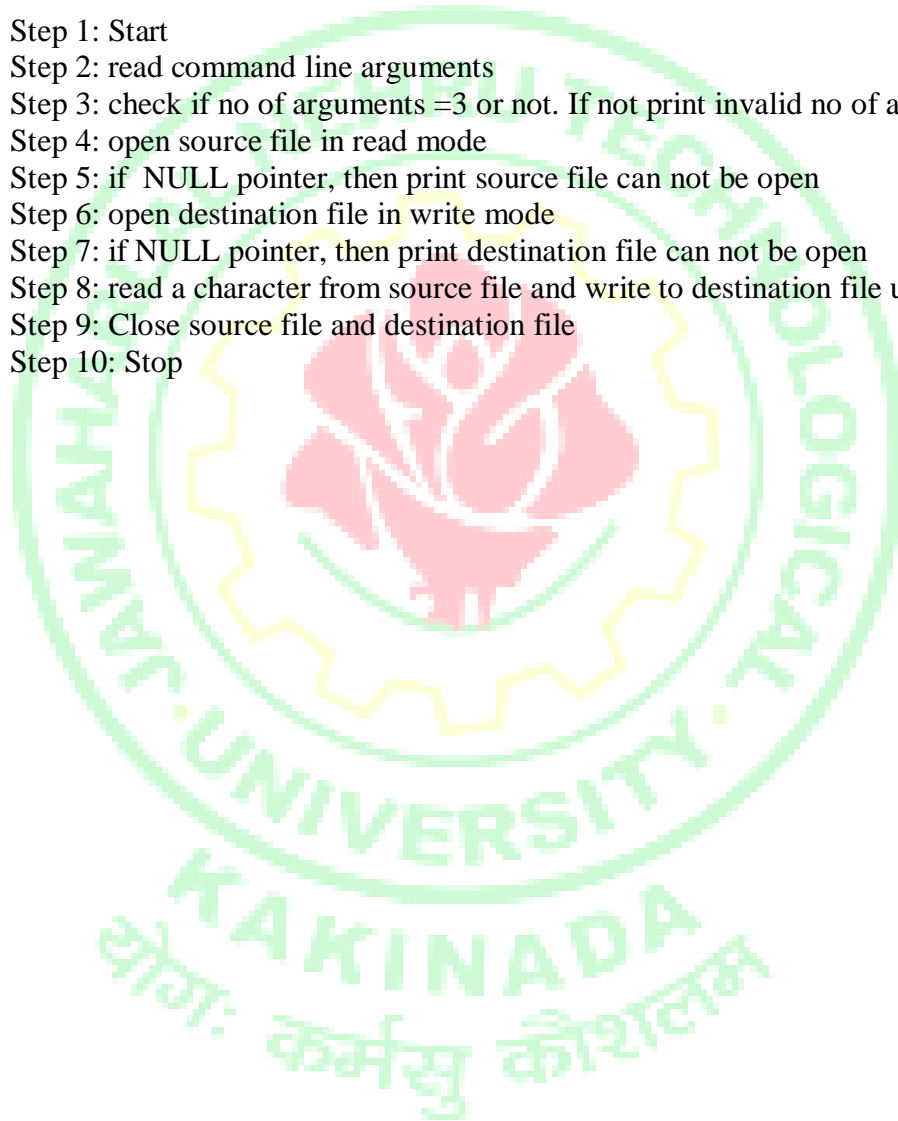
- a) Program which copies one file to another

Description:

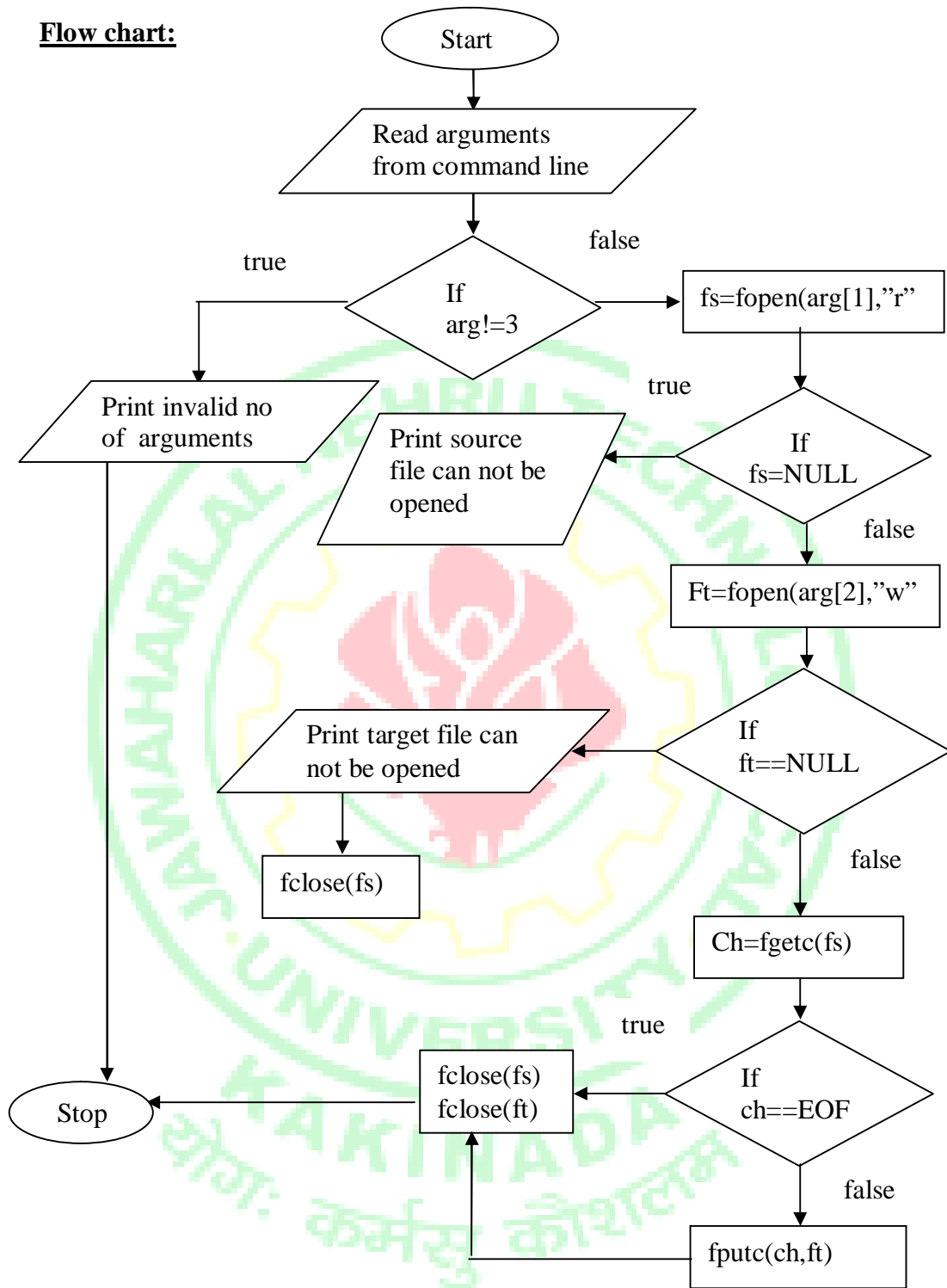
In this program we have to use the file functions to perform the copy operation from one file to another file.

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 <conio.h>
#include <process.h>

void main(int argc, char *argv[])
{
    FILE *fs, *ft;
    char ch;
    clrscr();
    if(argc!=3)
    {
        puts("Invalid number of arguments.");
        exit(0);
    }
    fs = fopen(argv[1],"r");
    if(fs==NULL)
    {
        puts("Source file cannot be opened.");
        exit(0);
    }
    ft = fopen(argv[2],"w");
    if (ft==NULL) // check the condition if the file pointer is NULL or not
    {
        puts("Target file cannot be opened.");
        fclose(fs);
        exit(0);
    }
    while(1)
    {
        ch=fgetc(fs);
        if (ch==EOF) // check the condition if the file is end or not
            break;
        else
            fputc(ch,ft);
    }
    fclose(fs);
    fclose(ft);
    getch();
}
```

Output:

source.c

 this is source text

ouput.c

Command line arguments

source.c ouput.c

source.c

 this is source text

ouput.c

 this is source text

Command line arguments

source.c

Invalid number of arguments.

Conclusion: the program is error free

VIVA QUESATIONS:

1) What is file ?

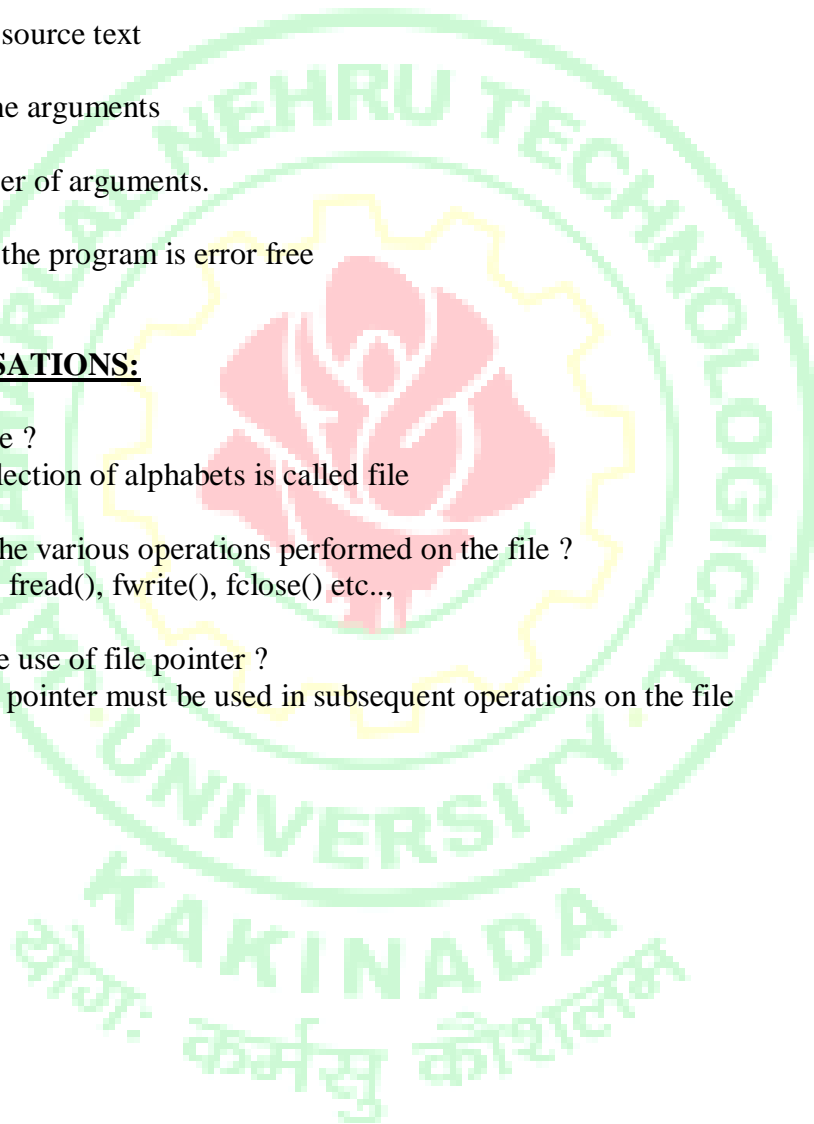
Ans: The collection of alphabets is called file

2) What are the various operations performed on the file ?

Ans: fopen(), fread(), fwrite(), fclose() etc.,

3) What is the use of file pointer ?

Ans: The file pointer must be used in subsequent operations on the file



Exercise : 16

- b) To reverse the first n characters in afile

Description:

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

Algorithm:

Step 1: Star

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

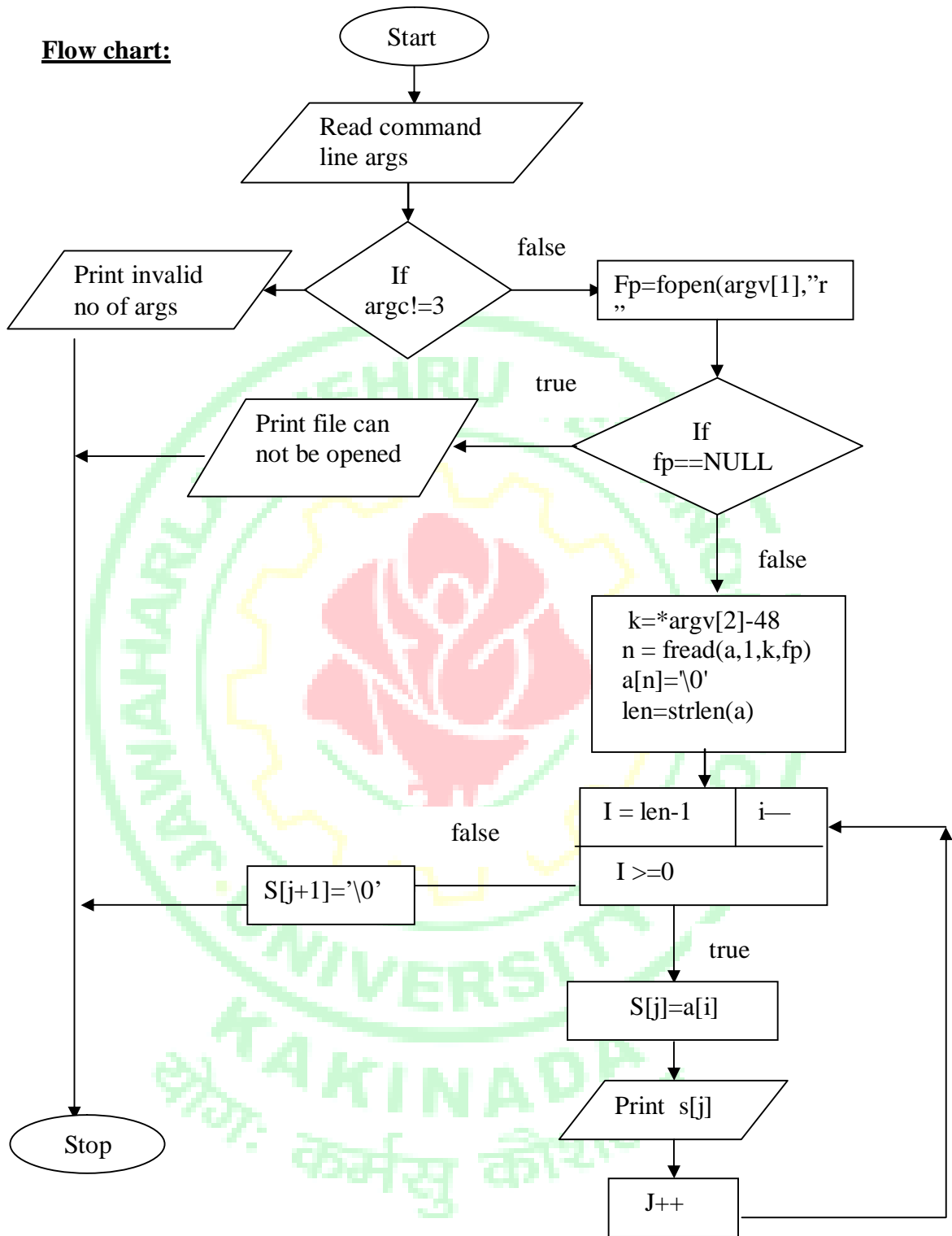
Step 8: Store chars from last position to initial position in another string(temp)

Step 9: print the temp string

Step 10: Stop



Flow chart:



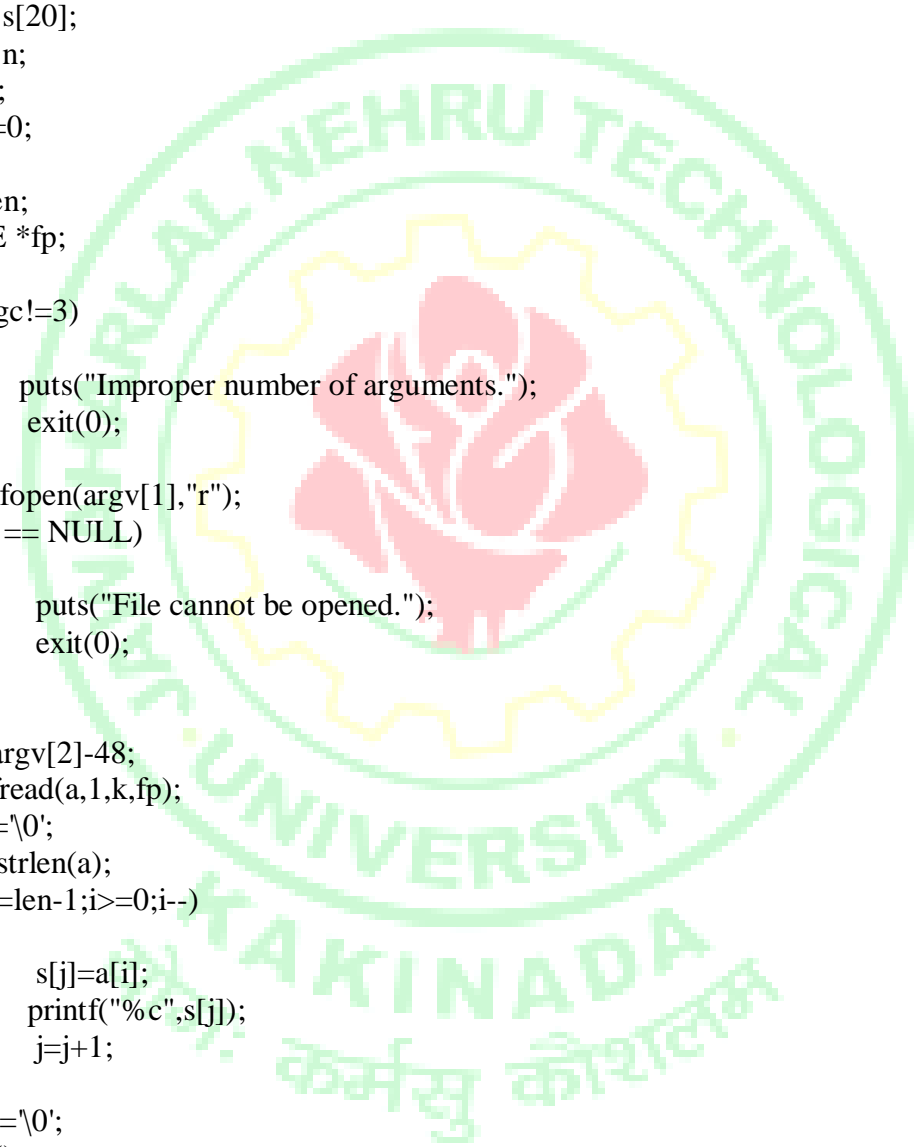
Program:

```
#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();
}
```



Output:

source.c

 this is source

ouput.c

Command line arguments

source.c ouput.c

source.c

 this is source

 ecruos si siht

Command line arguments

source.c

Invalid number of arguments.

Conclusion: the program is error free

VIVA QUESATIONS:

1) List out the file handling functions ?

Ans: fopen(), fprintf(),fclose(),fscanf(),fgetc(),fputc(), etc..,

2) What is the use of fseek() function ?

Ans: The function fseek sets the file pointer associated with a stream to a new position

3) What is use of the fflush() function ?

Ans: If the given stream has a buffered output, fflush writes the output of the stream to the associate file.

