

Experiment No. 1

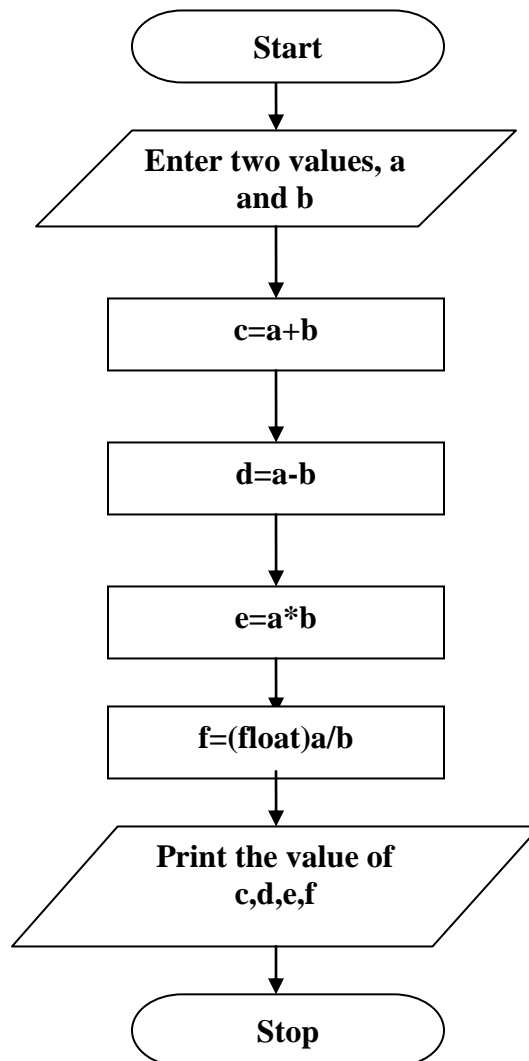
Aim : Write a Program to perform basic arithmetic operations.

Theory/Description:

This Program takes two integers **a**, **b** as input and performs all the basic arithmetic operations like addition, subtraction, multiplication, and division on them. In this Program the variable **c** represents the result of operation.

Algorithm :

- Step 1: Input any two numbers a, b.
- Step 2: Calculate the sum of two numbers($a + b$).
- Step 3: Calculate the difference of two numbers($a - b$).
- Step 4: Calculate the product of two numbers($a * b$).
- Step 5: Calculate the quotient of two numbers(a / b).
- Step 6: Print the sum, difference, product, and quotient.



Program :

```
/** PROGRAM TO PERFORM BASIC ARITHMETIC OPERATIONS **/  
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int a,b;  
    float c,d,e,f;  
    printf("Enter two values:");  
    scanf("%d%d",&a,&b);  
    c=a+b;  
    d=a-b;  
    e=a*b;  
    f=(float)a/b;  
    printf("%f%f%f%f",c,d,e,f);  
    getch();  
}
```

Input :

Enter two values:7 4

Output :

11

3

28

1.400000

Assignment

1. Write a Program to generate the following pattern.

```
-----  
/       \  
\  
/       \  
-----
```

2. Write a Program to calculate simple interest and compound interest.

Which of the following are invalid variable names and why?

BASICSALARY	_basic	basic-hra	#MEAN
group.	422	population in 2006	over time
mindovermatter	FLOAT	hELLO	queue.
Teams'svictory	Plot#3	2015_DDay	

3. Point out the errors, if any in the following C statements.

- a. `int = 314.562*150;`
- b. `name = 'Ajay';`
- c. `varchar = '3';`
- d. `3.14 * r * r * h = vol_of_cyl;`
- e. `K = (a * b) (c + (2.5 a + b) (d + e));`
- f. `m_inst = rate of interest * amount in rs;`
- g. `si = principal * rateofinterest * numberofyears/100;`
- h. `area = 3.14 * r * * 2;`
- i. `volume = 3.14 * r ^ 2 * h;`
- j. `k = ((a * b) + c) (2.5 * a + b);`
- k. `a = b = 3 =4;`
- l. `count = count +1 ;`

4. Evaluate the following expressions and show their hierarchy.

- a) $g = \text{big} / 2 + \text{big} * 4 / \text{big} - \text{big} + \text{abc} / 3;$
($\text{abc} = 2.5$, $\text{big} = 2$, assume g to be a float)
- b) $\text{on} = \text{ink} * \text{act} / 2 + 3 / 2 * \text{act} + 2 + \text{tig};$
($\text{ink} = 4$, $\text{act} = 1$, $\text{tig} = 3.2$, assume on to be an int)
- c) $s = \text{qui} * \text{add} / 4 - 6 / 2 + 2 / 3 * 6 / \text{god};$
($\text{qui} = 4$, $\text{add} = 2$, $\text{god} = 2$, assume s to be an int)
- d) $s = 1 / 3 * a / 4 - 6 / 2 + 2 / 3 * 6 / g;$
($a = 4$, $g = 3$, assume s to be an int)

5. Convert the following equations into corresponding C statements.

- a) $Z = 8.8 (a + b) 2 / c - 0.5 + 2 a / (q + r) / (a + b) * (1 / m)$
- b) $X = - b + (b * b) + 2 4 a c / 2 a$
- c) $R = 2 v + 6.22 (c + d) / g + v$

Viva -voce questions

1. Explain types of operators in C?
2. Give the precedence of different operators?
3. What is the difference between post increment and pre increment?
4. What is ternary operator?

Experiment No. 2

Aim : Write a Program to take the radius of a sphere as input and print its volume and surface area.

Description : This Program takes radius of a sphere as input and prints its volume and surface area. In the Program variable **r** represents radius, **v** represents volume of sphere and **sa** represents surface area.

Formulae : Volume of sphere= $\frac{4}{3} \pi r^3$

Surface area of sphere= $4 \pi r^2$

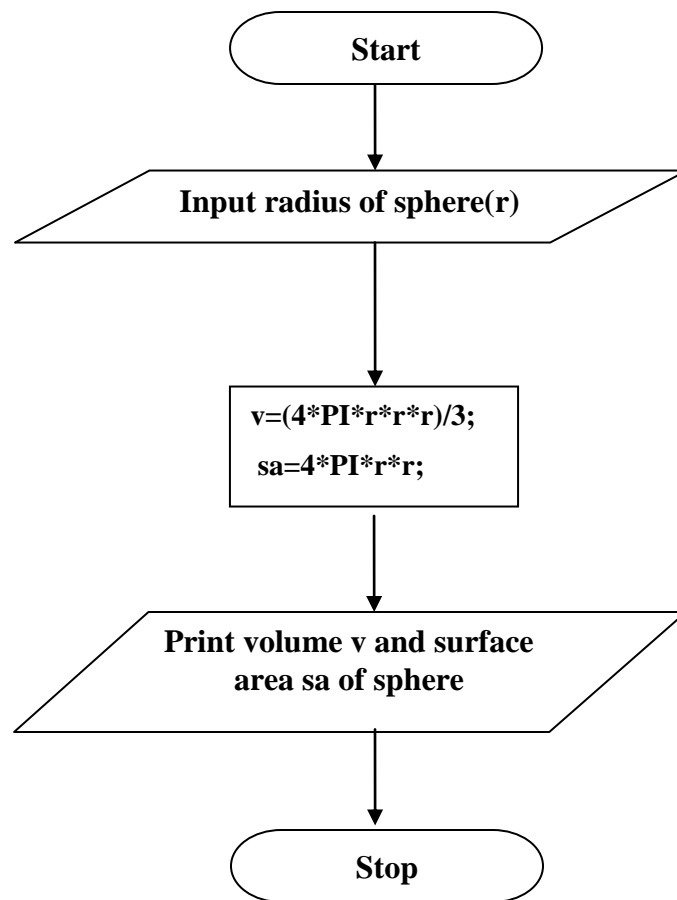
Algorithm :

Step 1: Input the radius of sphere(r).

Step 2: Calculate the volume of sphere using the formula $\frac{4}{3} * 3.14159 * r * r * r$.

Step 3: Calculate the surface area of sphere using the formula $4 * 3.14159 * r * r$.

Step 4: Print the volume and surface area.



Program :

```
/*... PROGRAM TO CALCULATE VOLUME AND SURFACE AREA OF A
SPHERE...*/
#include<stdio.h>
#include<conio.h>
#define PI 3.14159
void main( )
{
    float v,sa,r;
    clrscr( );
    printf("Enter radius of sphere :");
    scanf("%f",&r);
    v=(4*PI*r*r*r)/3;
    sa=4*PI*r*r;
    printf("Volume = %f\nSurface area=%f",v,sa);
    getch( );
}
```

Input :

Enter radius of sphere :5

Output :

Volume = 392.750000

Surface area=314.200012

Assignment

1. Write a Program to read a "float" representing a number of degrees Celsius, and print as a "float" the equivalent temperature in degrees Fahrenheit. Print your results in a form such as 100.0 degrees Celsius converts to 212.0 degrees Fahrenheit. Given as input an integer number of seconds, print as output the equivalent time in hours, minutes and seconds. Recommended output format is something like 7322 seconds is equivalent to 2 hours 2 minutes 2 seconds.
2. Given the values of a,b,c. Write a Program to rotate their values.

Viva -voce questions

1. What is macro ?
2. What is preprocessors?
3. What would be the output of the following Programs :

- a)

```
main()
{
    int i = 2, j = 3, k, l;
    float a, b;
    k = i / j * j;
    l = j / i * i;
    a = i / j * j;
    b = j / i * i;
    printf(“%d%d%f%f”,k,l,a,b);
}
```
- b)

```
main()
{
    int a, b;
    a = - 3 - - 3;
    b = - 3 - - ( - 3 );
    printf(“a = %d b = %d, a, b );
}
```
- c)

```
main()
{
    float a = 5, b = 2;
    int c;
    c = a % b;
    printf(“%d”, c );
}
```
- d)

```
main ( )
{
    printf(“\n\n\n\n”);
    printf(“\n/n/nn/n”);
}
```

```
e) main ( )
    {
      int a, b;
      printf("Enter values of a and b");
      scanf(" %d%d ", &a, &b);
      printf("a = %d b = %d", a, b);
    }
```

4. Pick up the correct alternative for each of the following questions.

a. C language has been developed by :

- 1) Ken Thompson
- 2) Dennis Ritchie
- 3) Peter Norton
- 4) Martin Richards

b. C can be used on :

- 1) Only MS-DOS operating system
- 2) Only Linux operating system
- 3) Only Windows operating system
- 4) All the above

c. C Program are converted into machine language with the help of :

- 1) An editor
- 2) A compiler
- 3) An operating system
- 4) None of the above

d. A character variable can at a time store :

- 1) 1 character
- 2) 8 characters
- 3) 254 characters
- 4) None of the above

e. The statement `char ch='Z';` would store in `ch`

- 1) The character Z
- 2) ASCII value of Z
- 3) Z along with the single inverted commas
- 4) Both(1) and (2)

5. What does `void main(void)` mean.

6. In C can we have comments inside a comment.

7. How do variables and symbolic names differ?

8. What is the main difference between an identifier and a keyword.

9. Distinguish between the following pairs :

- a) `getchar` and `scanf` functions.
- b) `%s` and `%c` specifications for reading.
- c) `%s` and `%[]` specifications for reading.
- d) `%g` and `%f` specifications for printing.
- e) `%f` and `%e` specifications for printing.

Experiment No. 3

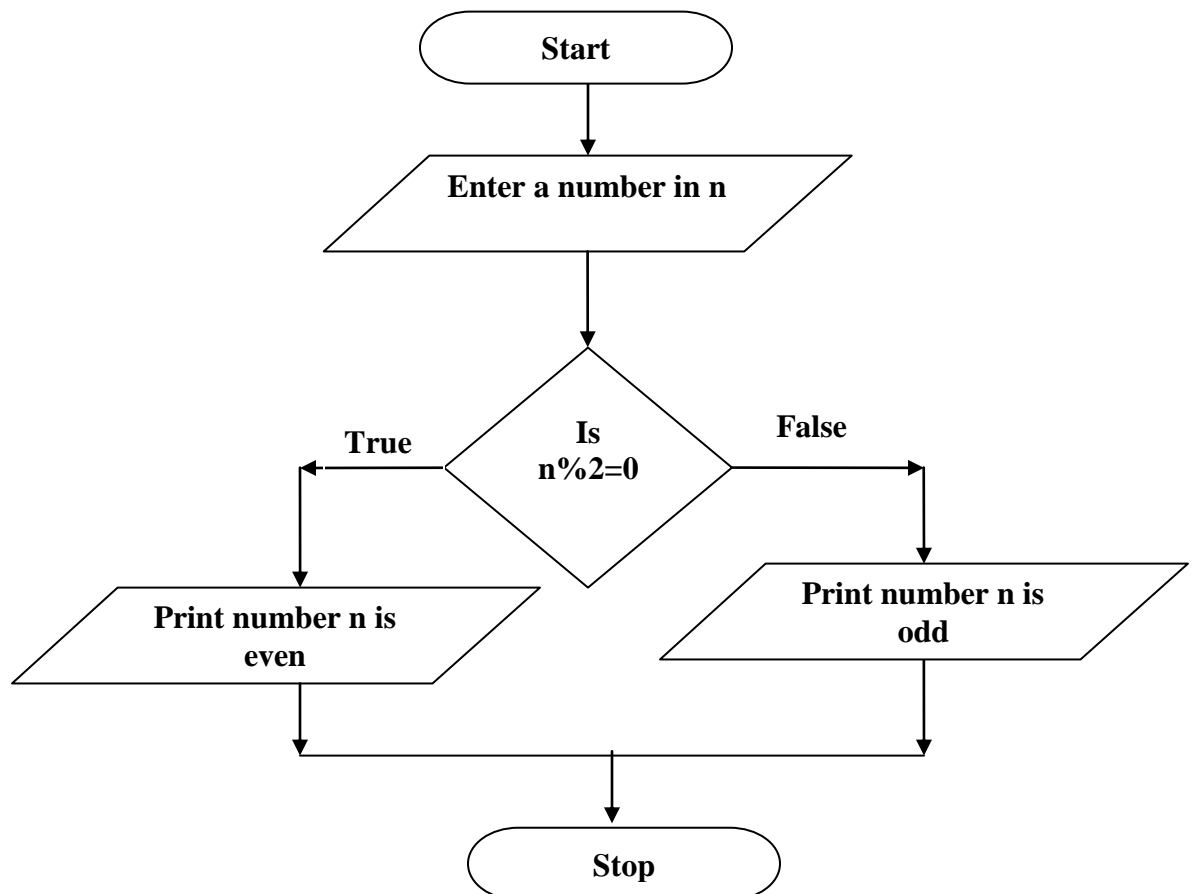
Aim: Write a Program to check whether a given number is even or odd.

Description :

This Program takes a number as input and finds whether that number is even or odd. In this Program the variable **n** represents the number which is checked for even or odd. To check whether the number is even or odd the value of **n** is divided by 2 if the remainder is zero then the number is even otherwise number is odd.

Algorithm :

- Step 1: Input any integer.
- Step 2: Check if number is even.
- Step 3: If number is even print number is even.
- Step 4: If it is not even print number is odd.



Program :

```
/*... PROGRAM TO CHECK WHETHER THE GIVEN NUMBER IS EVEN OR  
ODD...*/  
#include<stdio.h>  
#include<conio.h>  
void main( )  
{  
    int n;  
    printf("Enter a number:");  
    scanf("%d",&n);  
    if(n%2==0)  
        printf("\nGiven number is even");  
    else  
        printf("\nGiven number is odd");  
    getch( );  
}
```

Input :

Enter a number:7

Output :

Given number is odd

Assignment

1. Write a Program which reads two integer values. If the first is less than the second, print the message up. If the second is less than the first, print the message down. If the numbers are equal, print the message equal. If there is an error reading the data, print a message containing the word Error and perform `exit(0);`

Viva -voce questions

1. Give the syntax of if else statement?
2. What do you mean by decision making statements? how many types of decision making statements are there in C.
3. What would be the output of the following Programs :

```
c. main ( )
{
    int a = 300, b, c;
    if ( a >= 400 )
        b = 300;
        c = 200;
    printf(“\n%d %d”, b, c);
}
```

```
d. main ( )
{
    int a = 500, b, c;
    if ( a >= 400 )
        b = 300;
        c = 200;
    printf(“\n%d%d”, b, c);
}
```

```
e. main ( )
{
    int x = 3, y = 5;
    if ( x == 3 )
        printf(“\n%d”, x);
    else ;
        printf(“\n%d”, y);
}
```

```
f. main ( )
{
    int x = 3;
    float y = 3.0;
    if ( x == y )
```

```

    printf("\nx and y are equal");
else
    printf("\nx and y are not equal");
}

```

g. main()

```

{
    int x = 3, y, z;
    y = x = 10;
    z = x < 10;
    printf("\nx = %d y = %d z = %d", x, y, z);
}

```

4. Point out the errors, if any, in the following Programs:

a. main ()

```

{
    int tag = 0, code = 1;
    if ( tag == 0 )
        ( code > 1 ? print ("\n Hello") ? printf ("\n Hi"));
    else
        printf("\nHello Hi!!");
}

```

b. main ()

```

{
    int ji = 65;
    printf("\mji >= 65 ? %d : %c", ji);
}

```

c. main ()

```

{
    int x = 10, y = 10;
    x = 20 && y != 10 ? printf ("True") : printf ("False");
}

```

d. main ()

```

{
    int n = 9;
    ( n == 9 ? printf ("You are correct"); : printf("You are wrong"););
}

```

e. main ()

```

{
    int kk = 65 ll;
    ll = ( kk == 65 : printf ("\n kk == 65") : printf("\n kk != 65") );
}

```

Experiment No. 4

AIM: Write a Program to calculate the roots of a quadratic equation :

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

Description :

This Program takes values of three coefficient of equation as input and calculates its roots.

In this Program variables **a**, **b** and **c** represents coefficients of equation, **d** represents discriminant **r1** and **r2** represents roots of equation.

A quadratic equation has two roots which are given by the following formulae :

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Here in calculation three cases are possible :

1. If $d > 0$ then

$$r1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$r2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

2. If $d = 0$ then

$$r1 = r2 = -\frac{b}{2a}$$

3. If $d < 0$ the roots are imaginary.

Algorithm :

Step 1: Input the coefficients of the quadratic equation (for eg. a,b,c).

Step 2: Calculate the discriminant (d) using appropriate formula.

Step 3: Check if discriminant is greater than 0. If it is go to step 5.

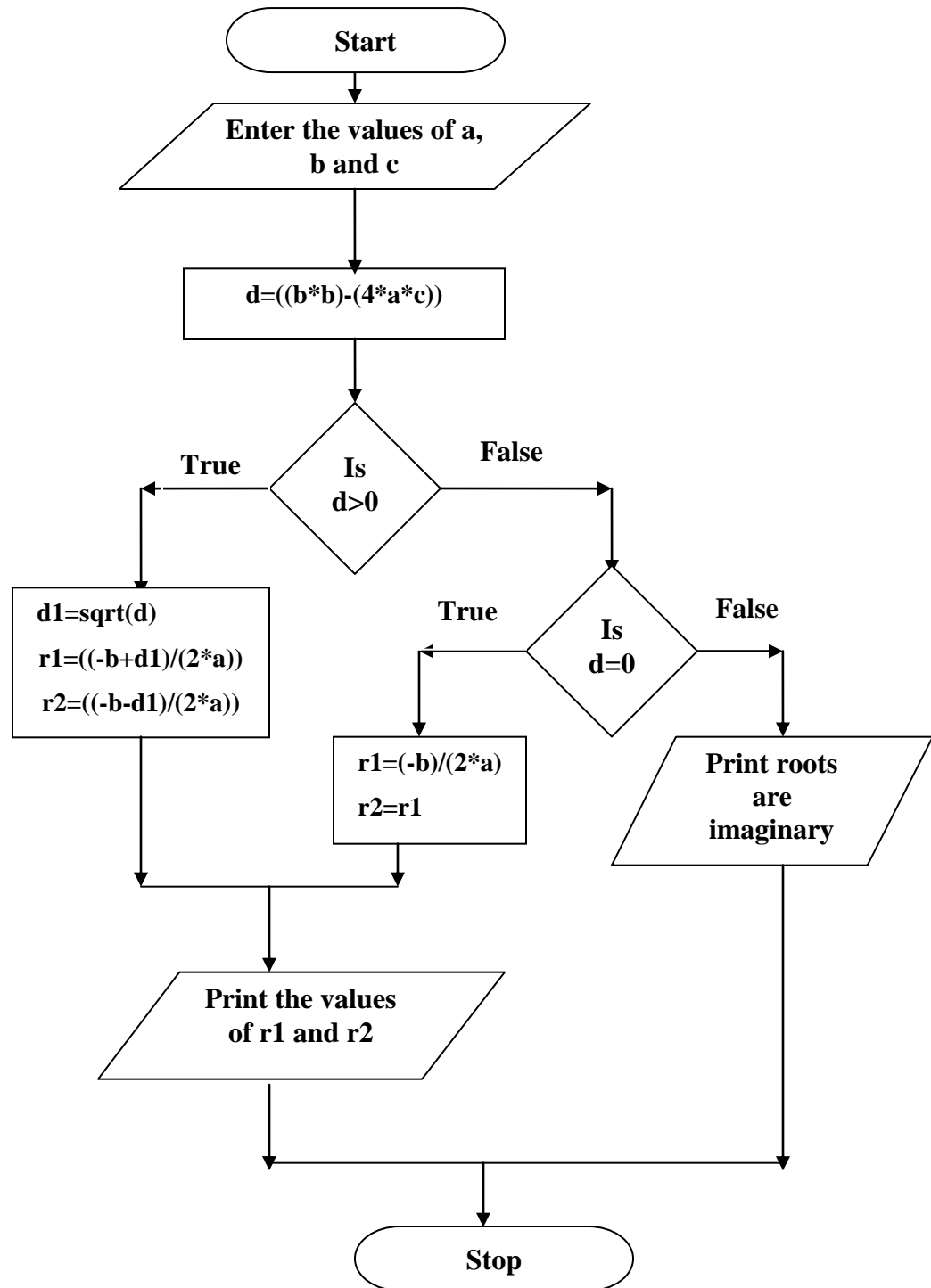
Step 4: If discriminant is not greater than 0 then check if discriminant is equal to 0.

If it is go to Step 5 otherwise go to Step 6.

Step 5: Calculate the roots. Go to Step 7.

Step 6: Print 'roots are imaginary' and return.

Step 7: Print the calculated roots and return.



Program : /*...PROGRAM TO FIND THE ROOTS OF A QUADRATIC EQUATION...*/

```
#include<conio.h>
#include<stdio.h>
#include<math.h>
int main( )
{
    int a,b,c,d;
    float r1,r2,d1;
    clrscr();
    printf("Enter Three Constants: ");
    scanf("%d%d%d",&a,&b,&c);
    d=((b*b)-(4*a*c));
    if(d>0)
    {
        printf("The Roots of equation are : ");
        d1=sqrt(d);
        r1=(-b+d1)/(2*a);
        r2=(-b-d1)/(2*a);
        printf("Root1 = %f\n Root2 = %f",r1,r2);
    }
    else if(d==0)
    {
        printf("\nThe Roots of equation are Real and Equal");
        r1=(-b)/(2*a);
        printf("\nThe Roots are =Root1=Root2= %f",r1);
    }
    else
    {
        printf("The Roots of equation are Imaginary");
    }
    getch();
    return 0;
}
```

Input :

Enter Three Constants: 1 2 1

Output :

The Roots of equation are Real And Equal

The Roots are =Root1=Root2= -1.000000

Input :

Enter Three Constants: 2 3 4

Output :

The Roots of equation are Imaginary

Assignment

1. Write a Program to calculate the division of students according to the following rules:

Above and equal to 75% -honours

Above and equal to 60% but less than75%- First division

Above and equal to 45% but less than60%- Second division

less than 45%-Fail

Viva -voce questions

1. What do you mean by else-if ladder?
2. In which situation do we use else-if ladder?

Experiment No. 5

Aim : Write a Program to find the largest of the three numbers. Using nested if...else.

Concept : Nested if...else.

Description :

This Program takes three integers **a**, **b** and **c** as input and finds the greatest among them.

In this Program values of two variables are first compared with each other and the value greater among them is then compared with the value of the third variable to find the greatest value.

Algorithm :

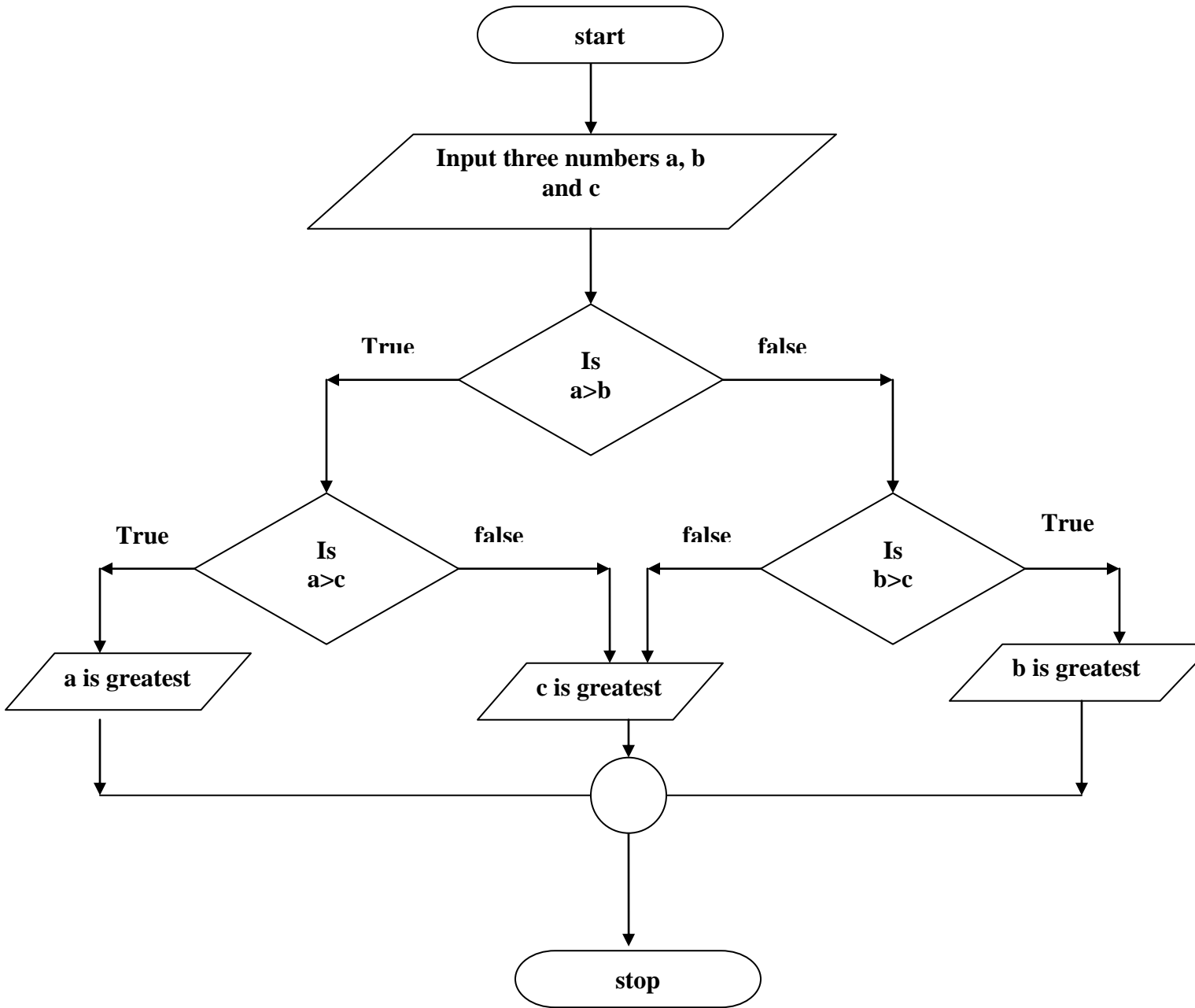
Step 1: Input any three numbers(for eg. a,b,c)

Step 2: Check if a is greater than b. if it is goto Step 3 otherwise go to Step 4.

Step 3: Check if a is greater than c. If it is print 'a is greatest' and return to operating system otherwise go to Step 5.

Step 4: Check if b is greater than c. If it is print 'b is greatest' and return to operating system.

Step 5: Print 'c is greatest'.



Program :

```
/*...PROGRAM TO FIND GREATEST AMONG THREE INTEGERS...*/
#include<stdio.h>
#include<conio.h>
void main( )
{
    int a,b,c;
    clrscr( );
    printf("Enter three numbers : \n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        if(a>c)
            printf("%d is greatest",a);
        else
            printf("%d is greatest",c);
    }
    else
    {
        if(b>c)
            printf("%d is greatest",b);
        else
            printf("%d is greatest",c);
    }
    getch();
}
```

Input :

Enter three numbers : 4 6 2

Output :

6 is greatest

Assignment

1. Write a Program to calculate whether a student is good only in English or in hindi or in maths or in any two or all three subjects or poor in all subjects depending on whether he/ she has obtained above or equal to 75 in individual subjects.

Viva -voce questions

1. What do you mean by nested if – else?
2. Draw the flowchart diagram for nested if-else.

Experiment No. 6

Aim: Write a Program to take three sides of a triangle as input and verify whether the triangle is an isosceles, scalene or an equilateral triangle.

Description :

In this Program variables **a**, **b** and **c** represents three sides of a triangle. Values of **a**, **b** and **c** are checked to find whether the triangle is an isosceles, scalars or an equilateral triangle.

- If the values of **a**, **b** and **c** are equal then the triangle is equilateral.
- If the values of two sides are equal then the triangle is isosceles.
- Otherwise triangle is scalars.

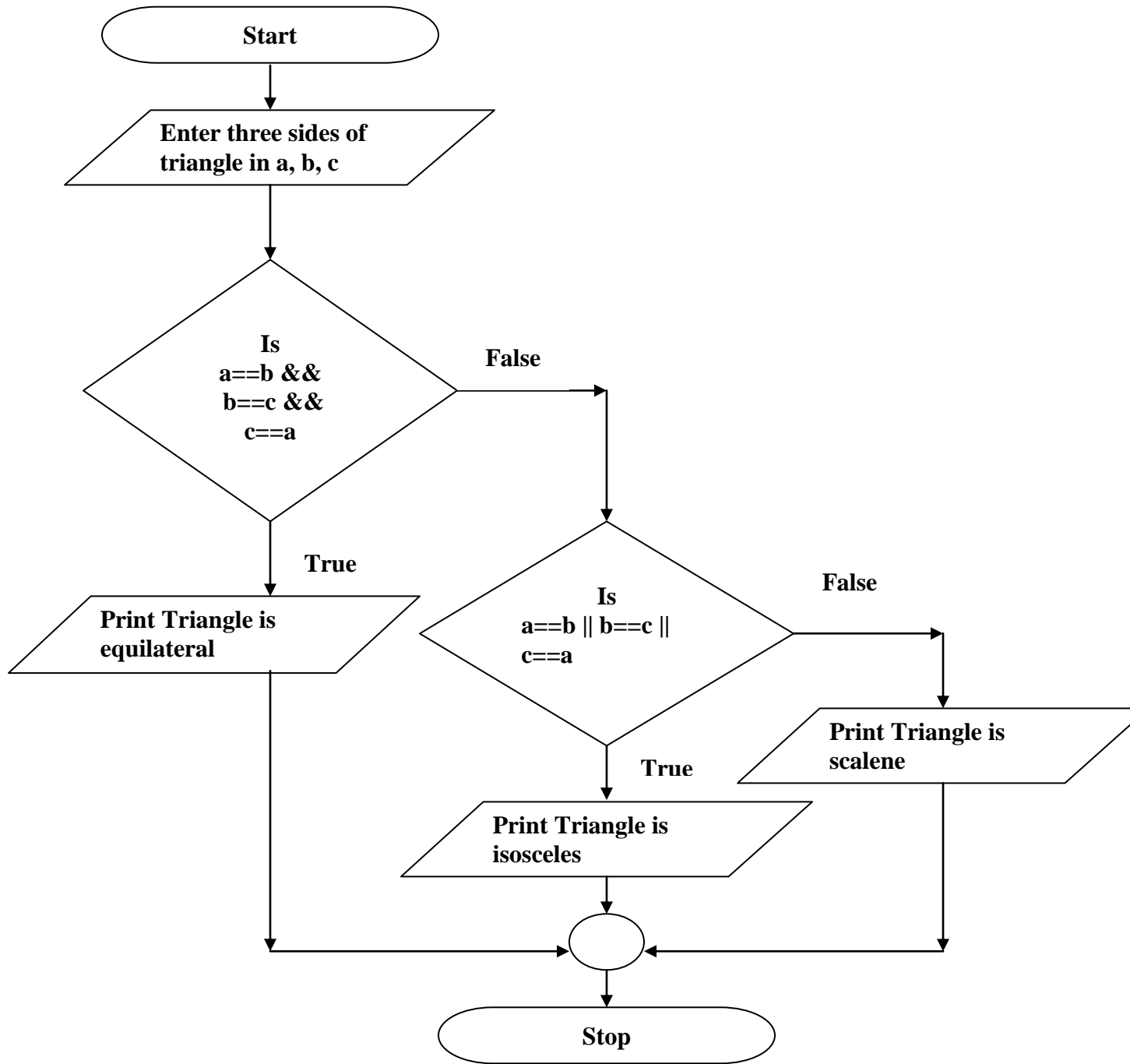
Algorithm:

Step 1: Input any three sides of a triangle (for eg. a,b,c)

Step 2: Check if all three sides are equal. If it is print ' Triangle is equilateral ' otherwise go to Step 3.

Step 3: Check if any two sides are equal. If it is print ' Triangle is isosceles ' otherwise go to Step 4.

Step 4: Print ' Triangle is scalene'.



Program :

```
/*...PROGRAM TO VERIFY WHETHER THE TRIANGLE IS AN ISOSCELES,  
SCALARS OR AN EQUILATERAL TRIANGLE...*/  
#include<stdio.h>  
#include<conio.h>  
void main( )  
{  
    int a,b,c;  
    clrscr( );  
    printf("Enter the three sides of a triangle:");  
    scanf("%d%d%d",&a,&b,&c);  
    if(a==b && b==c &&c==a)  
        printf("\nTriangle is equilateral");  
    else  
    {  
        if(a==b || b==c || c==a)  
            printf("\nTriangle is isosceles");  
        else  
            printf("\nTriangle is scalars");  
    }  
    getch( );  
}
```

Input :

Enter the three sides of a triangle:2 2 2

Output :

Triangle is equilateral

Input :

Enter the three sides of a triangle:2 3 4

Output :

Triangle is scalars

Input :

Enter the three sides of a triangle:2 3 2

Output :

Triangle is isosceles

Assignment

1. Write a Program to input the weight and height of 5 students and calculate the number of students whose weight is greater than 45 and height is more than 5 feet2 inches.

Viva -voce questions

- 1.What would be the output of the following :

```
main()  
{  
    int i = 4, j, num;  
    j = (num < 0 ? 0 : num * num);  
    printf("\n%d", j);  
}
```

- 2.Rewrite the following using conditional operator :

```
main()  
{  
    int x, min, max;  
    scanf("%d%d", &max, &x);  
    if(x < max)  
        max = x;  
    else  
        min = x;  
}
```

3. What would be the output of the following :

```
int i = 5, j = 3;  
if(i + j)  
    printf("HELLO");  
else  
    printf("HI");
```

Experiment No. 7

Aim: Write a Program to take three positive integers as input and verify whether they form a Pythagorean triplet.

Description : This Program takes three positive integers **a**, **b** and **c** as input and compares their values to find greatest among them. Then the square of the greatest value is checked with the sum of the squares of the other two values and if the square of the greatest value is equal to the sum of the square of the other two values then they form Pythagorean triplet, and if not they do not form Pythagorean triplet.

Algorithm :

Step 1: Input any three sides of a triangle (for eg. a,b,c)

Step 2: Check if a is greater than b and a is greater than c. if it is go to Step 3 else go to Step 4.

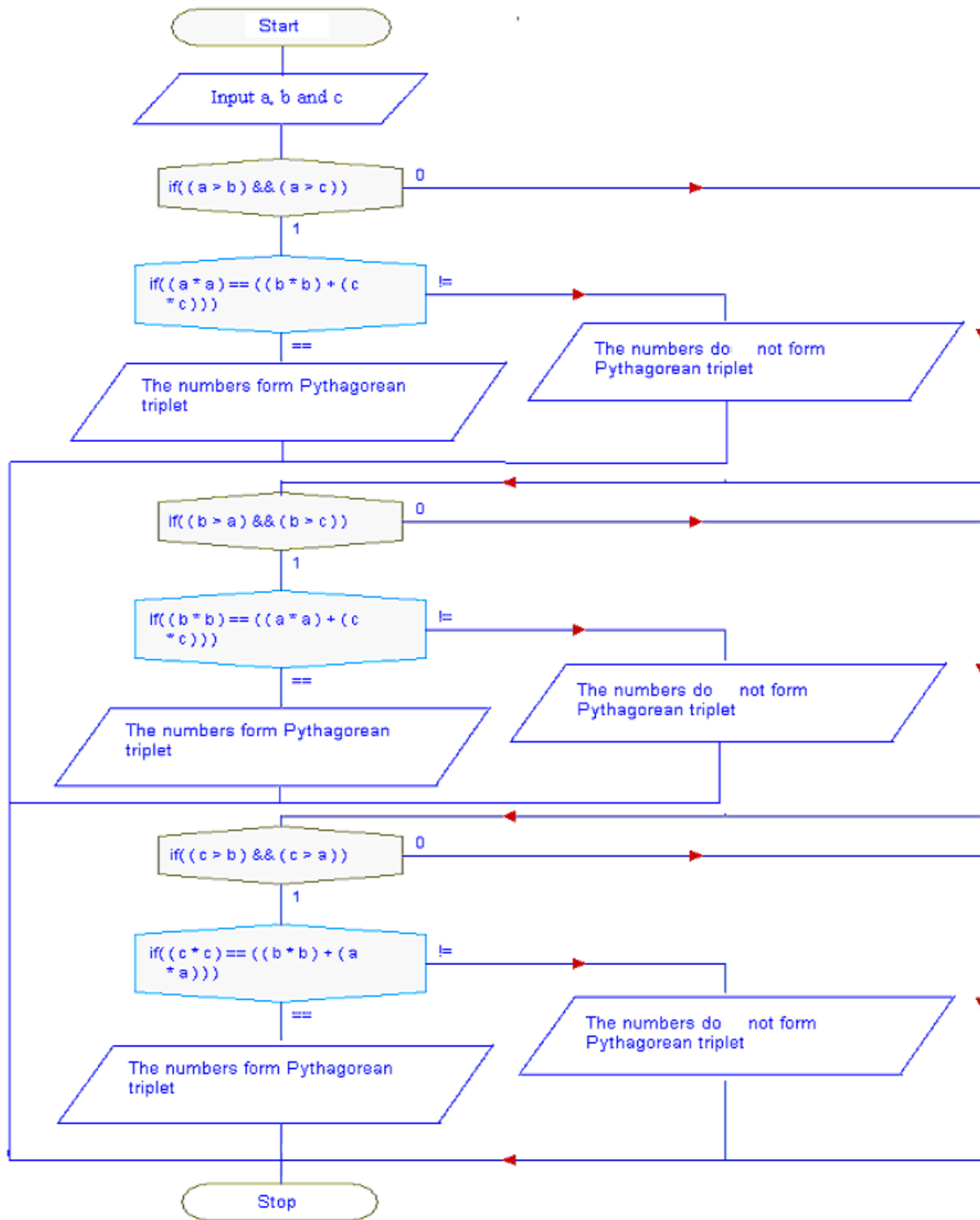
Step 3: Check if square of a is equal to sum of square of b and square of c. if it is print 'Pythagorean triplet' and return otherwise go to Step 7.

Step 4: Check if b is greater than a and b is greater than c. if it is go to Step 5 else go to Step 6.

Step 5: Check if square of b is equal to sum of square of a and square of c. if it is print 'Pythagorean triplet' and return otherwise go to Step 7.

Step 6: Check if square of c is equal to sum of square of a and square of b. if it is print 'Pythagorean triplet' and return otherwise go to Step 7.

Step 7: Print ' Not a Pythagorean Triplet'



Program:

```
#include<conio.h>
#include<conio.h>
int main()
{
    int a,b,c;
    clrscr();
    printf("Enter three positive integers : ");
    scanf("%d%d%d",&a,&b,&c);
    if((a>b)&&(a>c))
    {
        if((a*a)==((b*b)+(c*c)))
        {
            printf("The numbers form Pythagorean triplet");
        }
        else
        {
            printf("The numbers do not form Pythagorean triplet");
        }
    }
    else
    if((b>a)&&(b>c))
    {
        if((b*b)==((a*a)+(c*c)))
        {
            printf("The numbers form Pythagorean triplet ");
        }
        else
        {
            printf("The numbers do not form Pythagorean triplet ");
        }
    }
    else
    if((c>b)&&(c>a))
```

```

    {
        if((c*c)==((b*b)+(a*a)))
        {
            printf("The numbers form Pythagorean triplet ");
        }
        else
        {
            printf("The numbers do not form Pythagorean triplet ");
        }
    }
    getch();
    return 0;
}

```

Input :

Enter three positive integers : 3 4 5

Output :

The numbers form Pythagorean triplet

Input :

Enter three positive integers : 2 3 4

Output :

The numbers do not form Pythagorean triplet

Assignment

1. Input the length of four sides of a polygon and calculate whether the polygon is a rectangle or square or trapezium.

Viva -voce questions

1. Point out the errors, if any, in the following Programs:

- a.

```
main ()
{
    int tag = 0, code = 1;
    if ( tag == 0 )
        ( code > 1 ? print ("\n Hello") ? printf ("\n Hi"));
    else
        printf("\nHello Hi!!");
}
```
- b.

```
main ()
{
    int ji = 65;
    printf("\mji >= 65 ? %d : %c", ji);
}
```
- c.

```
main ()
{
    int x = 10, y = 10;
    x = 20 && y != 10 ? printf ("True") : printf ("False");
}
```
- d.

```
main ()
{
    int n = 9;
    ( n == 9 ? printf ("You are correct"); : printf ("You are wrong"));
}
```
- e.

```
main ()
{
    int kk = 65 ll;
    ll = ( kk == 65 : printf ("\n kk == 65") : printf ("\n kk != 65" ) );
}
```

2. What is meant by branching.

3. The following is a segment of a Program:

```
x = 1;
y = 1;
```

```
if(n > 0)
x = x + 1;
y = y - 1;
printf(“%d %d”, x, y);
```

What will be the value of x and y if n assumes a value of a) 1 and b) 0.

4. What would be the output of the following :

```
main()
{
    int i = 4, z = 12;
    if(i == 5 || z > 50)
        printf(“\nDean fo students affairs”);
    else
        printf(“\nDosa”);
}
```

5. Point out errors , if any in the following :

```
main()
{
    int i = 2, j = 5;
    if (i == 2 && j == 5)
        printf(“\nSatisfied at last”);
}
```

Experiment No. 8

Aim: Write a menu driven Program to perform various arithmetic operations.

Description :

This Program takes two values **a**, **b** from the keyboard, asks the user to enter choice whether he wants to perform addition, subtraction, multiplication or division and then perform the operation chosen by the user and displays the result of that operation as output. After performing operation according to user's choice the Program asks the user whether he wants to continue the above process and the user's answer in terms of 'y' or 'n' is stored in a variable **ch**, in case the value of **ch** is 'y' the whole process is repeated again. In this Program the variable **choice** represents the user's choice and the variable **c** represents the result of operation chosen by the user.

Algorithm:

Step 1: Input any two numbers.

Step 2: Print 1 for addition, 2 for subtraction, 3 for multiplication, 4 for division

Step 3: If the user inputs 1

- a. Add the two numbers and store in c.
- b. Print the value of c.
- c. Come out of the Program.

Step 4: If the user inputs 2

- a. Calculate the difference of the two numbers and store in c.
- b. Print the value of c.
- c. Come out of the Program.

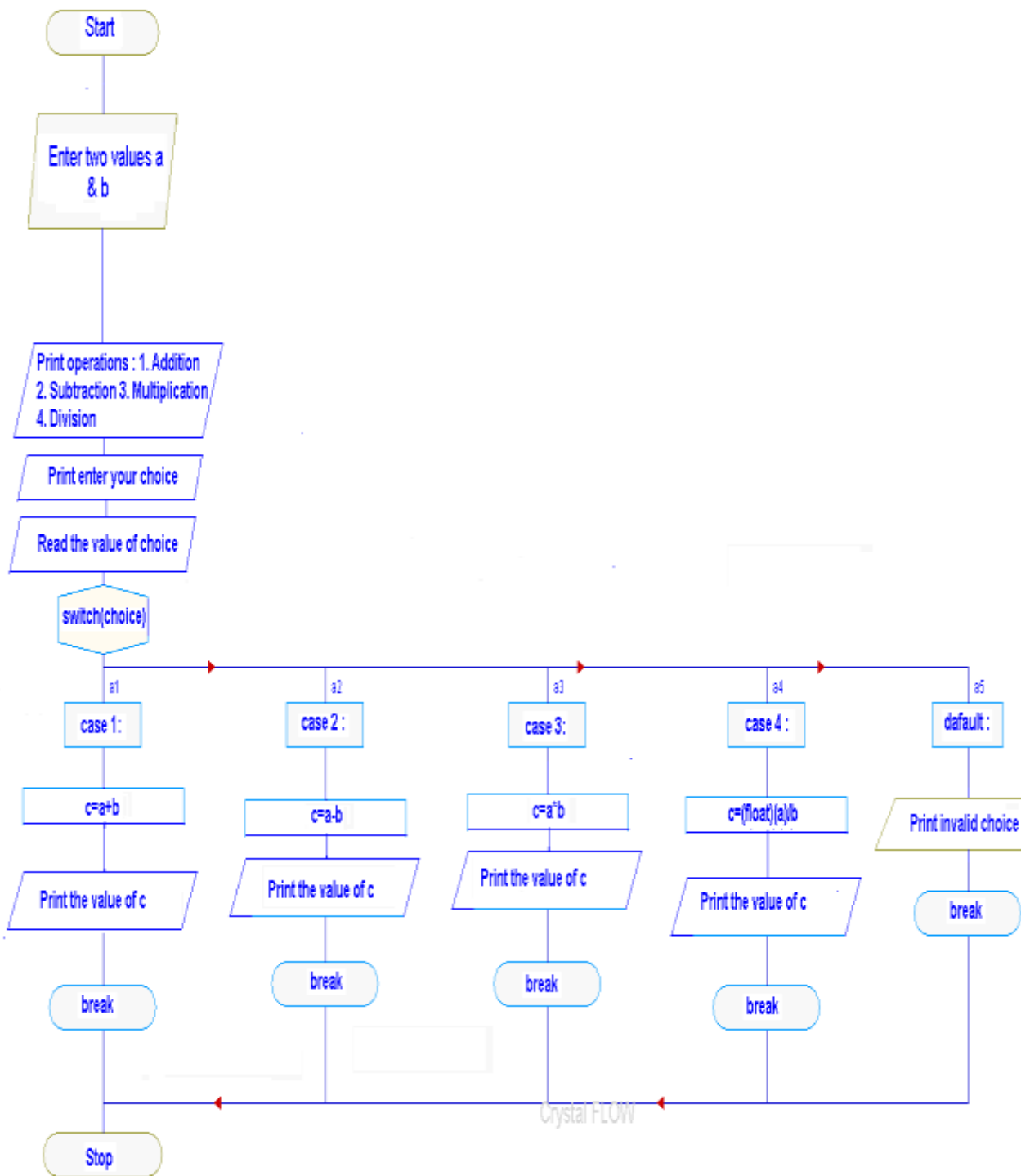
Step 5: If the user inputs 3

- a. Multiply the two numbers and store in c.
- b. Print the value of c.
- c. Come out of the Program.

Step 6: If the user inputs 4

- a. Divide the two numbers and store in c.
- b. Print the value of c.
- c. Come out of the Program.

Step 7: Print Invalid choice.



Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c,choice;
    clrscr();
    printf("Enter two values:");
    scanf("%d%d",&a,&b);

    printf("Operations:\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division"
);
    printf("\nEnter your choice:");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: c=a+b;
                printf("\nResult of addition is :%d",c);
                break;
        case 2: c=a-b;
                printf("\nResult of subtraction is :%d",c);
                break;
        case 3 :c=a*b;
                printf("\nResult of multiplication is :%d",c);
                break;
        case 4: c=(float)(a)/b;
                printf("\nResult of division is :%d",c);
                break;
        default : printf("Invalid choice");
                break;
    }
    getch();
}
```

Input:

Enter two values:2 5

Operations:

- 1.Addition
- 2.Subtraction
- 3.Multiplication
- 4.Division

Enter your choice:1

Output :

Result of addition is :7

Assignment

1. Read an integer value. Assume it is the number of a month of the year; print out the name of that month
2. Read an integer value. Assume it is the number of a week day; print out the name of that day

Viva -voce questions

1. What would be the output of the following

```
main()  
{  
  int i = 4;  
  switch(i)  
  {  
    default :  
      printf("\n A mouse is an elephant built by Japanese");  
    case 1 :  
      printf("\n Breeding rabbits is a hare raising experience");  
    case 2 :  
      printf("\n Friction is a drag");  
    case 3 :  
      printf("\nIf practice makes perfect, then nobody's perfect");  
  }}
```

2. Point out error, if any, in the following

```
main()
{
    int i = 4,j = 2;
    switch(i)
    {
        case 1 : printf("\n HELLO");
                break;
        case j : printf("\n HI");
                break;
    }
}
```

3. Point out error, if any, in the following

```
main()
{
    int i = 4,j = 2;
    switch(i)
    {
        case 1 : printf("\n HELLO");
                break;
        case 1 * 2 + 4: printf("\n HI");
                break;
    }
}
```

4. Point out errors, if any, in the following.

```
main()
{
    int a=10
    switch(a)
    {
    }
    printf("Programmres never die, they just get lost in the processing");
}
```

5. What is more efficient a switch statement or an if-else cahin?

6. We want to test whether a value lies in the range 2 to 4 or 5 to 7. Can we do this using a switch?

Experiment No. 9

Aim: Write a Program to evaluate square roots for five numbers making a loop using goto statement.

Description :

This Program takes a five numbers as input and evaluates their square root. The variable **count** keeps the count of numbers read. When the count is less than or equal to 5, goto **read**; directs the control to the label **read**, otherwise the Program stops. Before calculating square root the value of number is checked and if the number is positive then only its square root is calculated otherwise a message is displayed that the number is negative.

Algorithm:

Step 1: Initialize count=1.

Step 2: Check if count is less than or equal to 5. If it is Repeat steps 3,4,5,6,7 otherwise return to operating system.

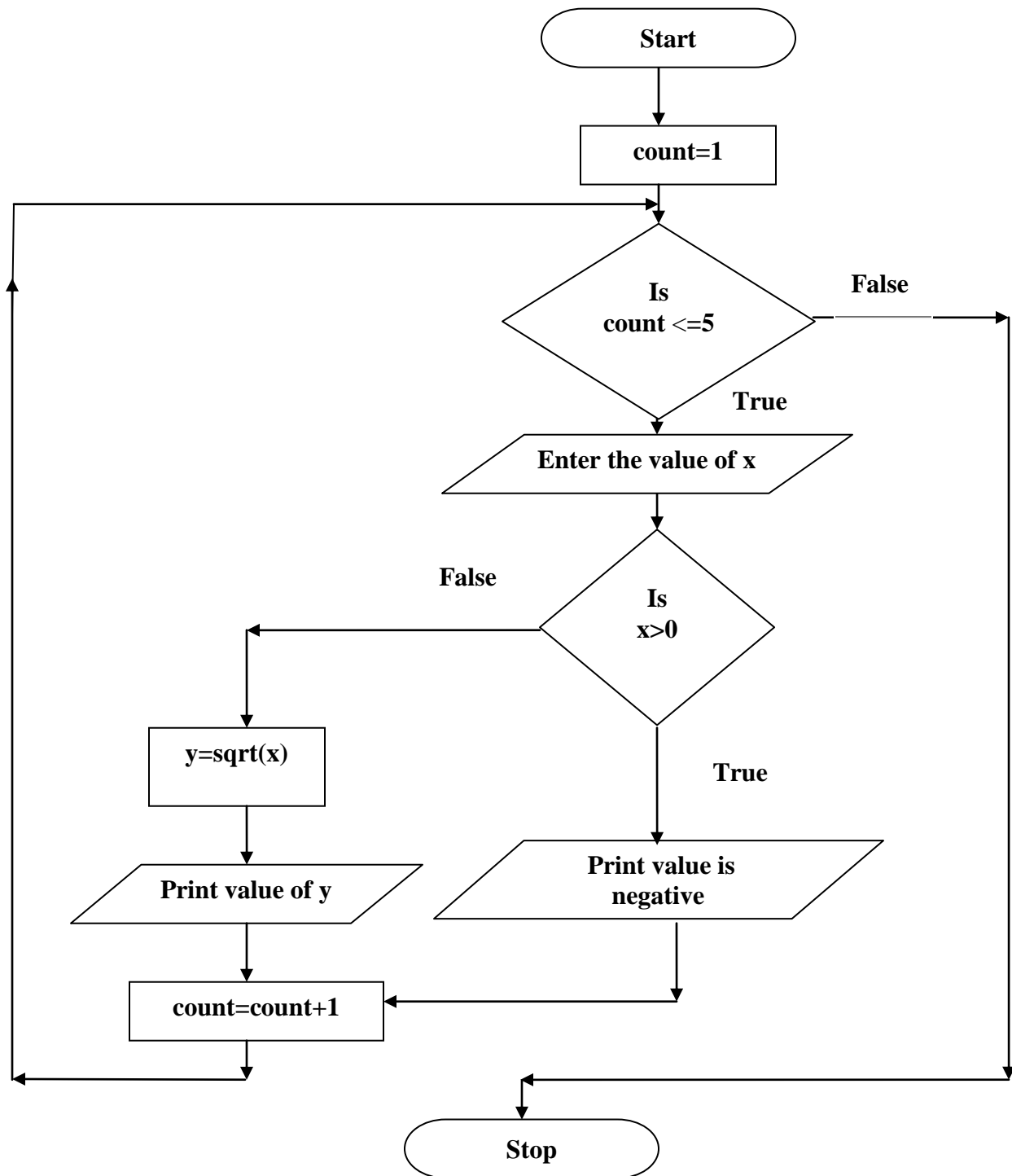
Step 3: Input any number (x).

Step 4: If x is greater than or equal to 0 then go to Step 5 otherwise print ' Value is negative' and go to Step 7.

Step 5: Calculate square root of x and store it in y.

Step 6: Print the value of y.

Step 7: Increment the value of count by 1.



Program :

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main( )
{
    double x,y;
    int count=1;
    printf("Enter five values :");
    read :
    scanf("%lf",&x);
    if(x<0)
        printf("value - %d is negative ",count);
    else
    {
        y=sqrt(x);
        printf("square root of %lf is : %lf",x,y);
    }
    count=count+1;
    if(count<=5)
        goto read;
    getch();
}
```

Input :

Enter five values :

25 16 -36 9 4

Output :

square root of 25 is : 5.000000

square root of 16 is : 4.000000

value - 3 is negative

square root of 9 is : 3.000000

square root of 2 is : 2.000000

Assignment

1. Write a Program for arithmetic operation using goto statement?

Viva- voce questions

1. Use of goto should be avoided. Explain a typical example where we find the application of goto becomes necessary.
2. In what ways does a switch statement differ from if statement?
3. What could we form a loop using goto statement ?
4. Write a Program to count the sum of five numbers using goto statement.

Experiment No. 10

Aim: Write a Program to take a number as input and calculate the sum of its digits.

Description :

In this Program the variable **n** represents number whose sum of digits is to be calculated and **sum** represents the sum of its digits. To calculate sum number is divided with ten till its value is greater than zero and the remainder of the division are added to find the required sum.

Algorithm:

Step 1: Initialize sum =0.

Step 2: Input any integer(n).

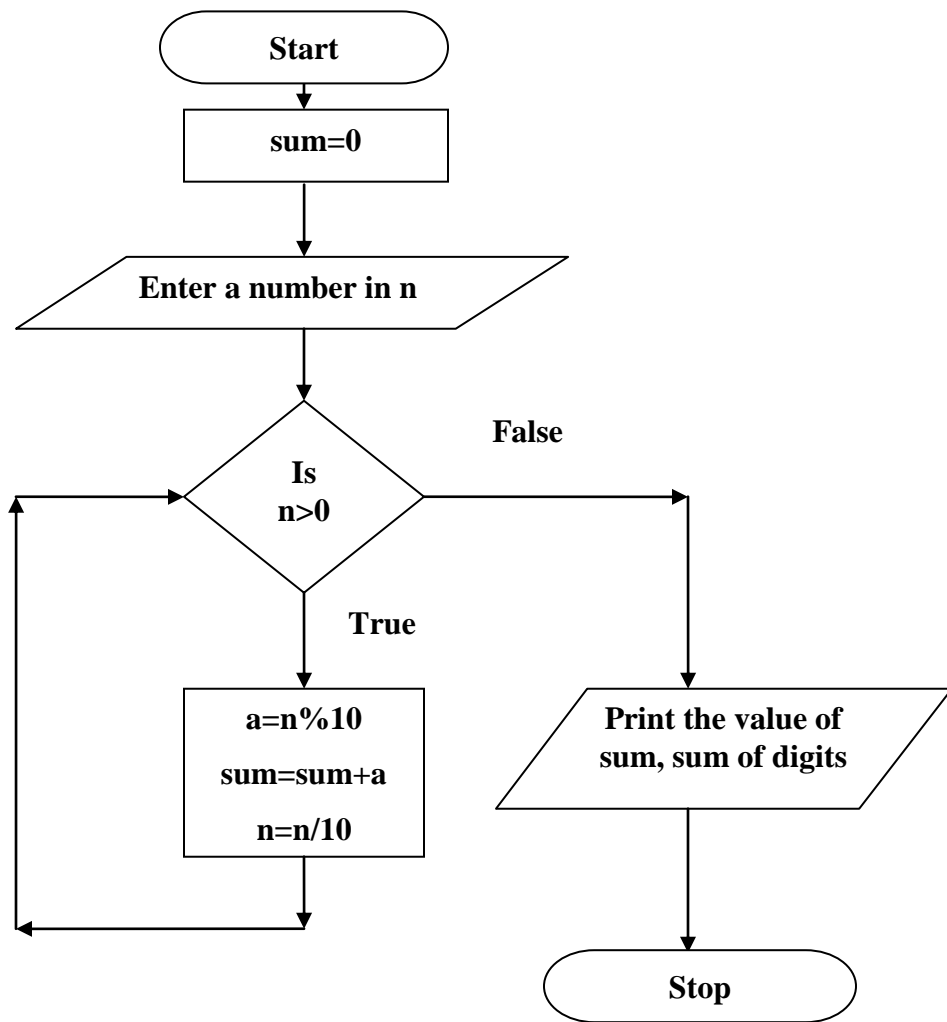
Step 3: Repeat Steps 4, 5, 6 while $n \geq 0$.

Step 4: Apply modulus operator on n and store the result in any variable (a).

Step 5: Calculate the sum of sum and a and store it back in sum.

Step 6: Apply the division operator on n and store the result in n .

Step 7: Print the value of variable sum as the sum of digits.



Program :

```
#include<conio.h>
#include<stdio.h>
void main()
{
    int n,a,sum=0;
    clrscr();
    printf("\nEnter a Number : ");
    scanf("%d",&n);
    while(n>0)
    {
        a=n%10 ;
        sum=sum+a;
        n=n/10;
    }
    printf("\nsum of the digits of given number is : %d",sum);
    getch();
}
```

Input :

Enter a Number : 1256

Output :

sum of the digits of given number is : 14

Assignment

1. Write a Program to reverse the given string using for loop?
2. Write a Program to find the no of vowels in a given string using for loop/

Viva-voce questions

1. What is meant by looping?
2. What is a null statement? Explain typical use of it.
3. How can we create an infinite loop?
4. What would be the output of the following Programs :

a) main ()

```
{
    int i = 1;
    while ( i <= 10)
    {
        printf("\n%d", i);
    }
}
```

b) main ()

```
{
    int x = 4;
    while( x == 1 )
    {
        x = x - 1;
        printf("\n%d", x);
    }
}
```

c) main()

```
{
    char x;
    while(x = 0;x <= 255;x ++ )
        printf("\nAscii value % d character %c", x, x);
}
```

5. Write a Program to read in 10 numbers and compute the average, maximum and minimum values.

Experiment No. 11

Aim : Write a Program to evaluate the equation : $y=x^n$

Description : In this Program the variable **x** represents the number and the variable **n** represents the power by which **x** is raised to and **y** represents the required result. The variable **y** is initialized to **1** and then multiplied by **x**, **n** times using while loop. The loop control variable **count** is initialized outside the loop and incremented inside the loop. When the value of **count** becomes greater than **n**, the control exits the loop.

Algorithm :

Step 1 : Read the values of x and n.

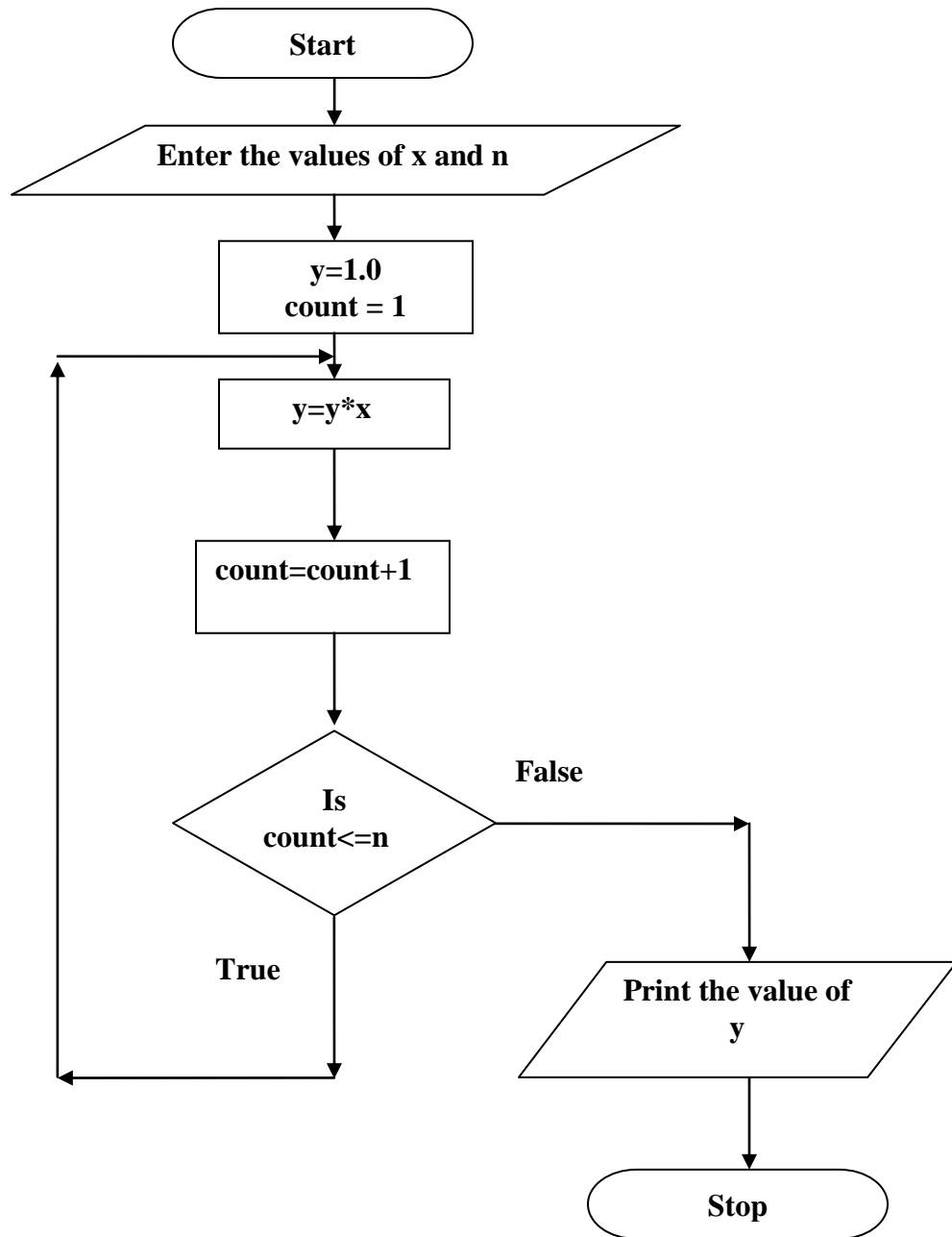
Step 2 : Initialize $y = 1.0$ and $count = 1$.

Step 3 : Repeat Steps 4, 5 while $count \leq n$.

Step 4 : $y = y * x$.

Step 5 : Increment $count = count + 1$.

Step 6 : Print the value of y.



Program :

```
#include<conio.h>
#include<stdio.h>
void main()
{
    int count=1,n;
    float x,y;
    printf("Enter the values of x and n:");
    scanf("%d%d",&x,&n);
    y=1.0;
    do
    {
        y=y*x;
        count++;
    }while(count<=n);
    printf("%d to the power %f is : %f",n,x,y);
    getch( );
}
```

Input :

Enter the values of x and n:2 5

Output :

5 to the power 2 is : 32

Assignment

1. Calculate the factorial of 5 numbers input by the user using do-while.

Viva-voce questions

1. What is the difference between while and do-while loop?
2. What do you mean by entry control & exit controlled loop?
3. What would be the output of the following :

```
int y, x = 0;
do
{
    y = x;
}while(x == 0);
```

4. Write down the syntax of do..while loop.

Experiment No. 12

Aim : Write a Program to calculate factorial of a given number.

Description :

In this Program variable **n** represents the number whose factorial is to be found and **f** represents the factorial of **n**. The variable **f** is initialized to **1** and then multiplied by the loop control variable **i**, **n** times. When the value of **i** becomes greater than **n**, the control exits the loop.

Algorithm:

Step 1: Initialize $f = 1$

Step 2: Input the value of number whose factorial is to be found (n).

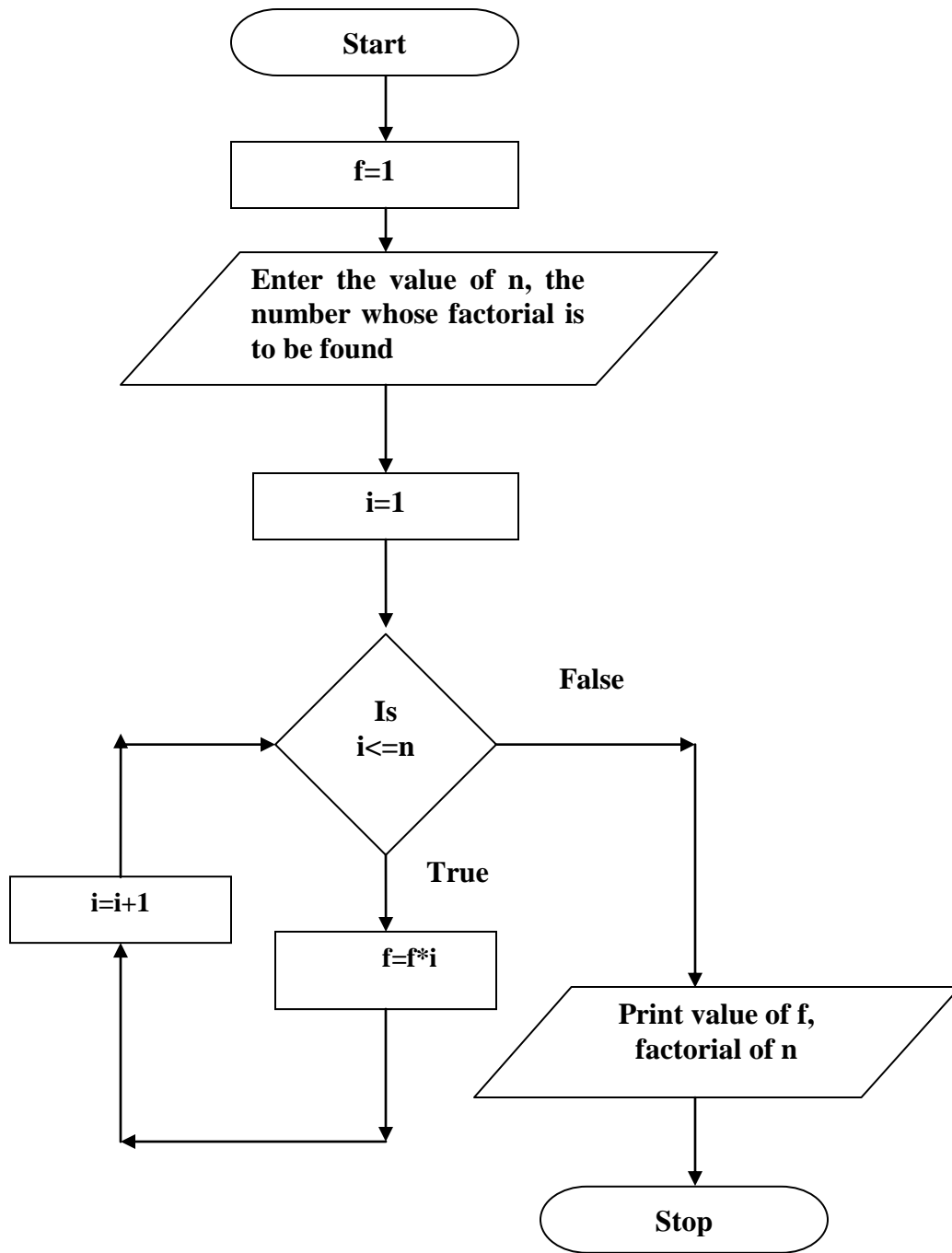
Step 3: Initialize $i = 1$.

Step 4: Repeat steps 5, 6 while $i \leq n$.

Step 5: Multiply the present value of i with f and store it in f .

Step 6: Increment the current value of i by 1.

Step 7: Print the value of variable f as the factorial of the number.



Program :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    long int f=1;
    int n,i;
    printf("Enter the number whose factorial is to be found:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    f=f*i;
    printf("Factorial of %d is: %ld",n,f);
    getch( );
}
```

Input :

Enter the number whose factorial is to be found: 6

Output :

Factorial of 6 is: 720

Assignment

1. Write a Program to calculate the result of series $\log(1+x)$.

Viva-voce questions

1. Explain the format of for loop.
2. Can we omit any one of the section of for loop?
3. What would be the output of the following :

```
for(i=0;i<=10;i++);  
{  
    i++;  
    printf("%d",i);  
}
```

4. What would be the output of the following :

```
for( ; ; )  
{  
    printf("HELLO");  
}
```

Experiment No. 13

Aim : Write a Program to print the Floyd's triangle :

```
1
2 3
4 5 6
7 8 9 10
.....
```

Description :

This Program takes number of rows **n** to be displayed as input and prints the triangle up to that number of rows. In this Program two loops are there one for rows which is up to **n** and the other is for columns which is up to the control variable of the outside loop. A variable **c** is initialized to zero outside the loop and incremented and printed inside the loop to display the desired triangle.

Algorithm:

Step 1: Input the number of rows upto which Floyd's triangle is to be generated(**n**).

Step 2: Initialize $i=1$, $j=1$ and $c=0$.

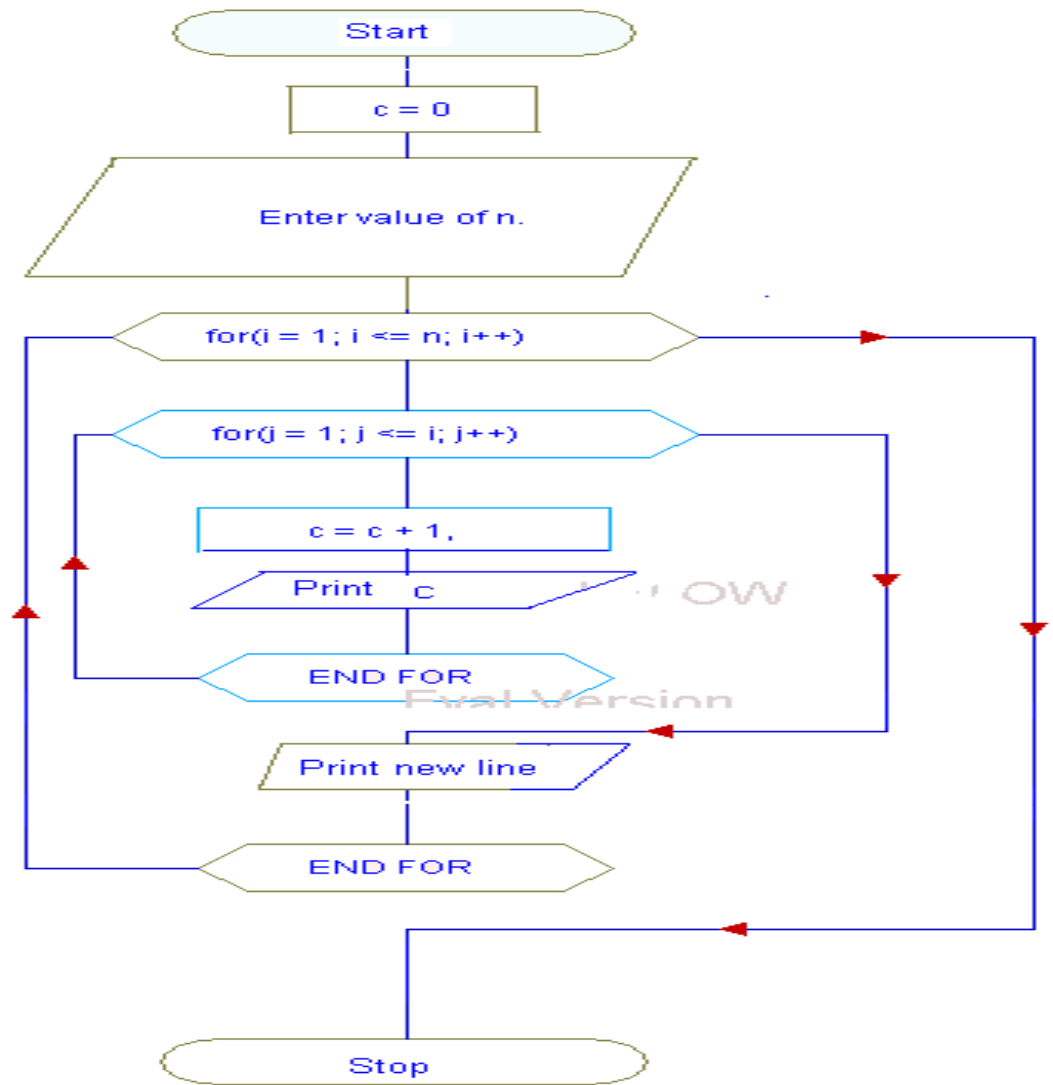
Step 3: Repeat steps 4, 5, 6, 7 while $i \leq n$.

Step 4: Repeat steps 5,6 while $j \leq i$.

Step 5: Increment the value of **c** by 1.

Step 6: Print the value of **c**.

Step 7: Print new line.



Program :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n,i,j,c=0;
    clrscr();
    printf("Enter value of n:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=i;j++)
        {
            c=c+1;
            printf("%-2d ",c);
        }
        printf("\n");
    }
    getch();
}
```

Input :

Enter value of n : 5

Output :

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Assignment

1. Write a Program to generate Pascal's triangle.

Viva-voce questions

1. What do you mean by nested for loop?
2. What would be the output of the following :

```
for(i=1;i<=5;i++)  
{  
    for(j=1;j<=i;j++)  
    {  
        printf("%d",j);  
    }  
}
```


Experiment No. 14

Aim : Write a Program to evaluate the square root of a series of numbers and print the results. The process stops when the number 9999 is typed in.

Description : In this Program the variable **number** represents the number whose square root is to be calculated and the variable **count** keeps the count of numbers of **number**. Variable **negative** counts the number of negative numbers. In case the series contains any negative number, the process of evaluation of square root should be bypassed for such numbers because the square root of a negative number is not defined. The continue statement is used to achieve this. The Program also prints a message saying that the number is negative and keeps an account of negative numbers. The final output includes the number of positive values evaluated and the number of negative items encountered.

Algorithm:

Step 1: Initialize count=0, negative=0.

Step 2: Check if count is less than or equal to 100. If it is go to step 3 otherwise go to Step 9.

Step 3: If number entered is 9999 then go to Step 9 otherwise proceed.

Step 4: If the number is negative then print 'number is negative' and increment the variable negative and continue with the next iteration of the loop otherwise proceed.

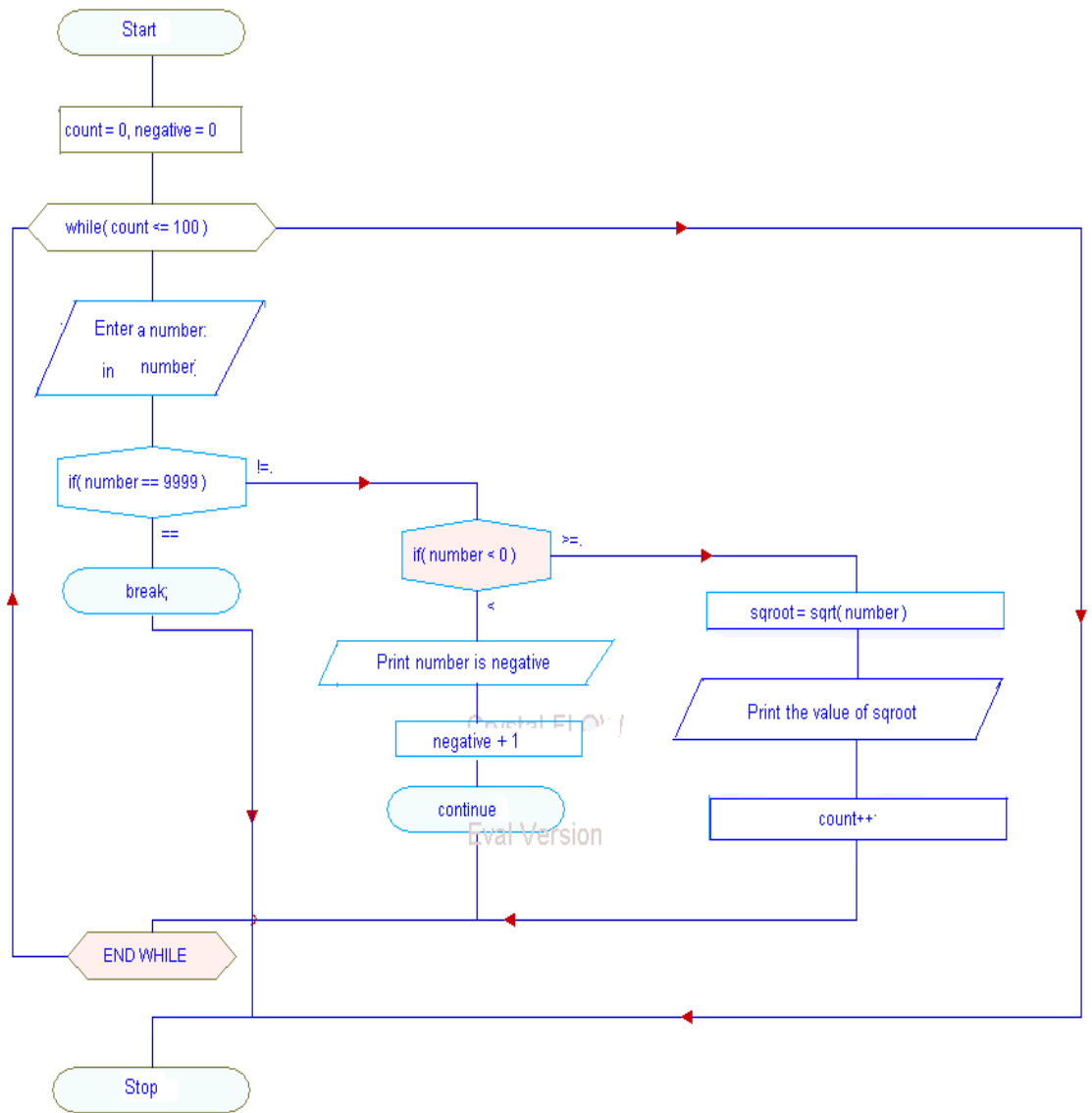
Step 5: If the number is positive calculate the square root .

Step 6: Print the square root.

Step 7: Increment count

Step 8: Go to Step 2.

Step 9: Print the value of negative and count.



Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int count=0,negative=0;
    double number,sqroot;
    while(count<=100)
    {
        printf("Enter a number: ");
        scanf("%lf",&number);
        if(number==9999)
            break;
        if( number<0)
        {
            printf("Number is negative\n");
            negative++;
            continue;
        }
        sqroot=sqrt(number);
        printf("Number=%lf\n Square root=%lf\n",number,sqroot);
        count++;
    }
    printf("Number of items done=%d\n",count);
    printf("\nNegative items =%d\n",negative);
    getch( );
}
```

Output:

Enter a number: 25.0

Number=25.000000

Square root=5.000000

Enter a number: 40.5

Number=40.500000

Square root=6.363961

Enter a number: -9

Number is negative

Enter a number: 9999

Number of items done=2

Negative items =1

Assignment

1. Write a Program to take as input 100 numbers from the users. If the number is negative it comes out of the Program.

Viva-voce questions

1. Explain break and continue statement?
2. The break statement is used to exit from :
 - a) An if statement
 - b) A for loop
 - c) A Program
 - d) The main() function
3. Which of the following statement is used to take the control to the beginning of the loop?
 - a) exit
 - b) break
 - c) continue
 - d) None of these

Experiment No. 15

Aim : Write a Program to find the sum of the elements of an array.

Description :

This Program takes an array as input and displays the sum of its elements as output. This Program also takes number of elements in the array as input. In this Program variable **a[]** represents the array, **n** represents the number of elements in the array and **sum** represents the sum of the elements in the array. To calculate sum the variable **sum** is initialized to zero the added with elements of array inside a loop. the final value of **sum** is printed outside the loop.

Algorithm:

Step 1: Input the number of elements of the array(n).

Step 2: Initialize i=0.

Step 3: Repeat Step 4, 5 till i is less than or equal to n .

Step 4: Enter the value of the element of array(a[i]).

Step 5: Increment the value of i by 1.

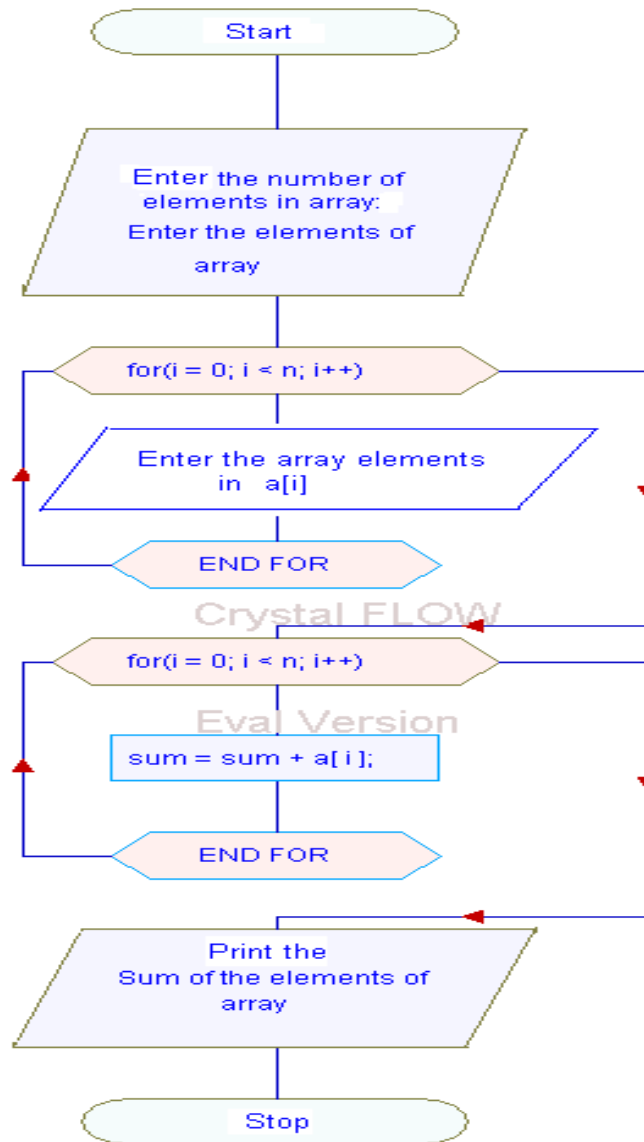
Step6: Initialize i=0, sum=0.

Step 7: Repeat Steps 8, 9 till i is less than or equal to n .

Step 8: Add the current value of sum to the current value of a[i] and store the result in sum.

Step 9: Increment the value of i by 1.

Step 10: Print the value of the variable sum as the sum of the elements.



Program :

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    int a[10],sum=0,i;
    printf("Enter the number of elements in array:");
    scanf("%d",&n);
    printf("\nEnter the elements of array :\n");
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
        sum=sum+a[i];
    printf("Sum of the elements of array is :%d",sum);
    getch();
}
```

Input :

Enter the number of elements in array:5

Enter the elements of array :

1
2
3
4
5

Output :

Sum of the elements of array is :15

Assignment

1. Write a Program to count the vowels and letters in free text given as standard input. Read text a character at a time until you encounter end-of-data.
2. Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed.

Viva-voce questions

1. What do you mean by the term array? In which situation do we use arrays.
2. Identify error, if any, in each of the following statements.
 - a) `int score(100);`
 - b) `float values[10,15];`
 - c) `float average[ROW],[COLUMN]`
 - d) `char name[15];`
 - e) `int sum[];`
3. What are the different ways to initialize an array?

Experiment No. 16

Aim: Write a Program that will take two integer arrays as input and insert the common elements of both array into a third array.

Description :

This Program takes two arrays $x[]$, $y[]$ as input, copies their common elements in third array $z[]$ and displays the elements of third array as output. To find the common elements the elements of array x and y are compared one by one and before inserting common element in array z a check is perform to see whether that element is already present in array or not and if not that element is inserted in array z .

Algorithm:

Step 1: Input the elements of the first array.

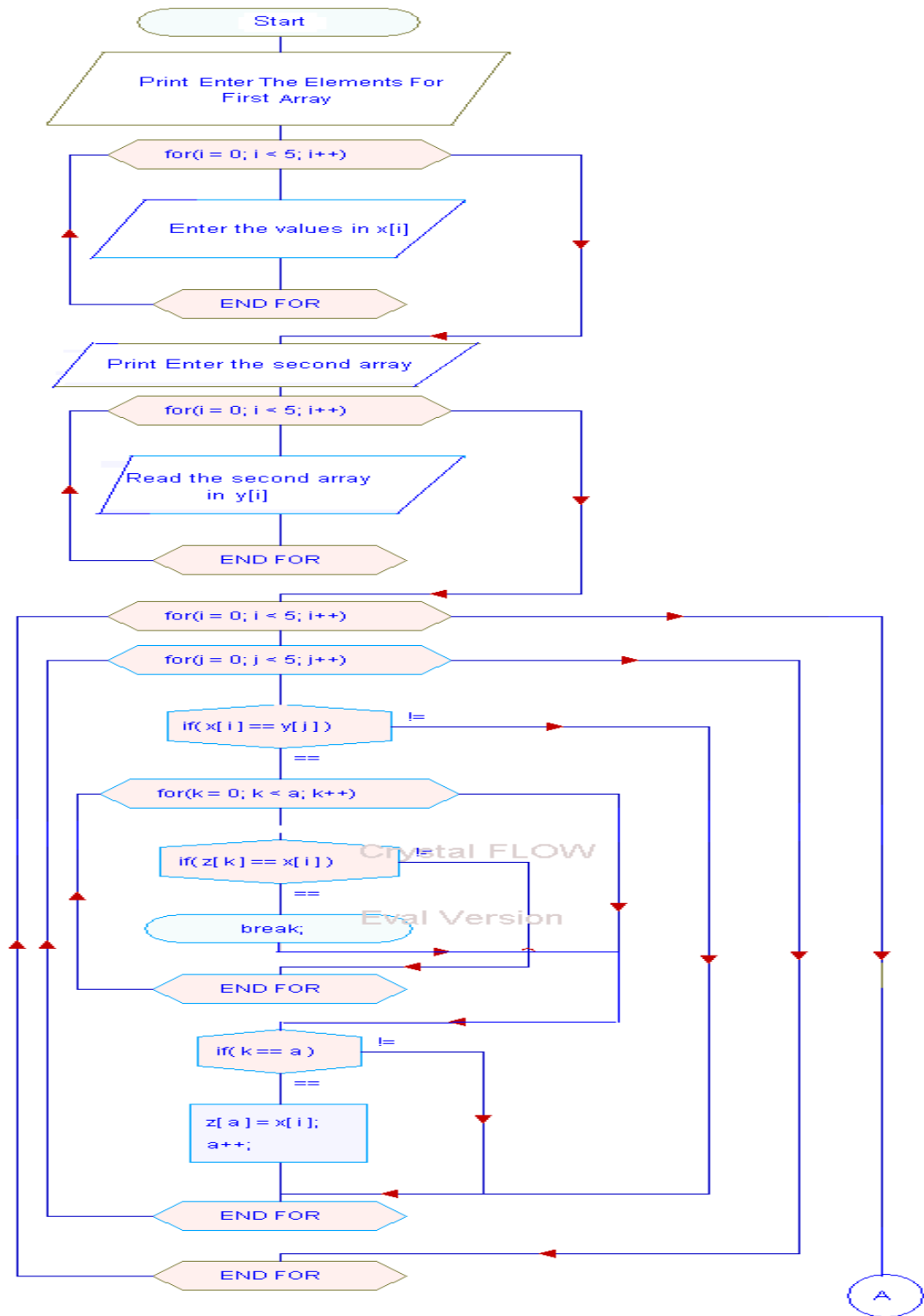
Step 2: Input the elements of the second array.

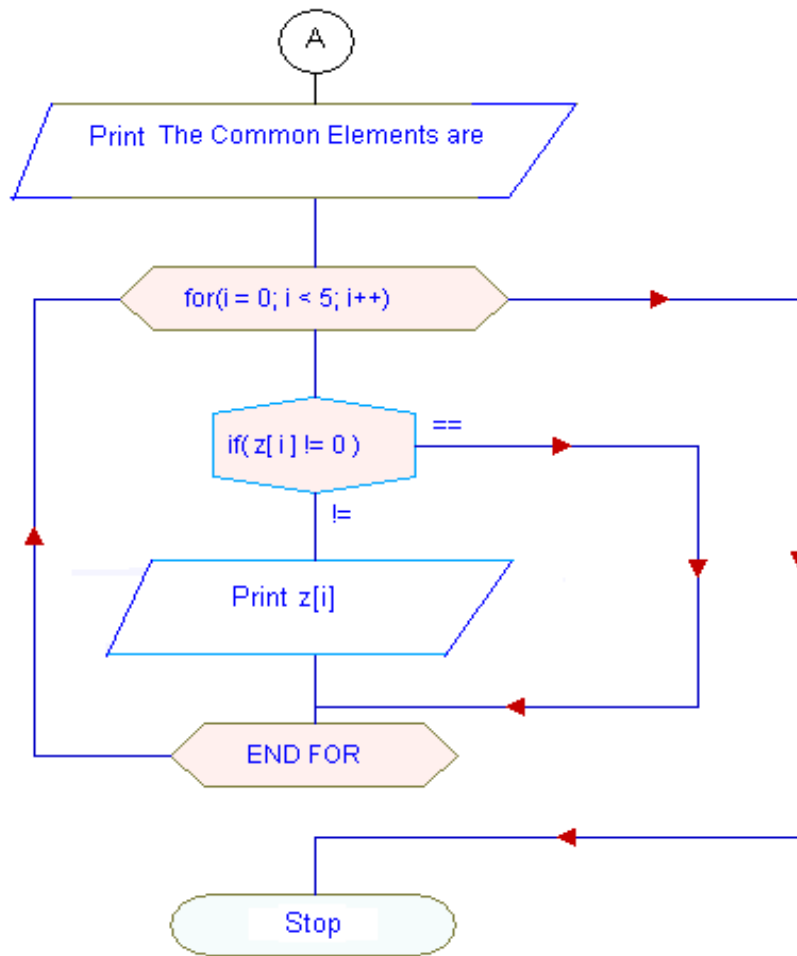
Step 3: Initialize a third array which will contain the common elements with all its elements 0.

Step 4: The elements of the two array input by the user are compared.

Step 5: If there is a match then the third array is searched if that element already exists.

Step 6: If it exists element is not inserted else it is inserted.





Program :

```
#include<conio.h>
#include<stdio.h>
int main()
{
    int i,j,k,a=0,x[5],y[5],z[5]={0};
    clrscr();
    printf("\nEnter The Elements For 1st Array : \n");
    for(i=0;i<5;i++)
    {
        scanf("%d",&x[i]);
    }
    printf("\nEnter The Elements For 2nd Array : \n");
    for(i=0;i<5;i++)
    {
        scanf("%d",&y[i]);
    }
    for(i=0;i<5;i++)
    {
        for(j=0;j<5;j++)
        {
            if(x[i]==y[j])
            {
                for(k=0;k<a;k++)
                {
                    if(z[k]==x[i])
                    break;
                }
                if(k==a)
                {
                    z[a]=x[i];
                    a++;
                }
            }
        }
    }
}
```

```
        }
    }
    printf("\nThe Common Elements are : \n");
    for(i=0;i<5;i++)
    {
        if(z[i]!=0)
            printf("\n%d",z[i]);
    }
    getch();
    return 0;
}
```

Input :

Enter The Elements For 1st Array :

1
2
3
4
5

Enter The Elements For 2nd Array :

2
5
6
7
8

Output :

The Common Elements are :

2
5

Assignment

1. Write a Program to find an element in an array of integers using linear search.

Viva-voce questions

1. What is the difference among the 5's in these two expressions ?

```
int num[5];  
num[5] = 11;
```

- a) first is particular element, second is type
 - b) first is array size, second is particular element
 - c) first is particular element, second is array size
 - d) both specify array size
2. An array is a collection of
 - a) different data types scattered throughout memory
 - b) the same data type scattered throughout memory
 - c) the same data type placed next to each other in memory
 - d) different data types placed next to each other in memory

3. Point out the errors, if any in the following :

```
main()  
{  
    int a[10], i;  
    for(i = 1; i <= 10; i ++)  
    {  
        scanf("%d", a[i]);  
        printf("%d", a[i]);  
    }  
}
```

4. What would be the output of the following :

```
main()  
{  
    int sub[50], i;  
    for(i = 0; i <= 48; i ++)  
    {  
        sub[i] = i;  
        printf("\n%d", sub[i]);  
    }  
}
```

Experiment No. 17

Aim : Write a Program to sort an array using bubble sort.

Description : This Program takes a one dimensional array as input and arrange them ascending order using bubble sort technique. In this Program variable **a[]** represents an array, consisting of **n** elements.

The sorting ALGORITHM works as follows :

1. Compare the first two elements in array say **a[1]**, and **a[2]**. If **a[2]** is smaller than **a[1]**, then interchange their values.
2. Compare **a[2]** and **a[3]**; interchange if **a[3]** is smaller than **a[2]**.
3. Continue this process till the last two elements are compared and interchanged.
4. Repeat the above steps **n-1** times.

In repeated trips through array, the smallest elements '**bubble up**' to the top. Because of this bubbling up effect, this ALGORITHM is called **bubble sorting**.

The bubbling effect is illustrated below for four items :

TRIP-1

80	35	35	35
35	80	65	65
65	65	80	10
10	10	10	80

TRIP-2

35	35	35
65	65	10
10	10	65
80	80	80

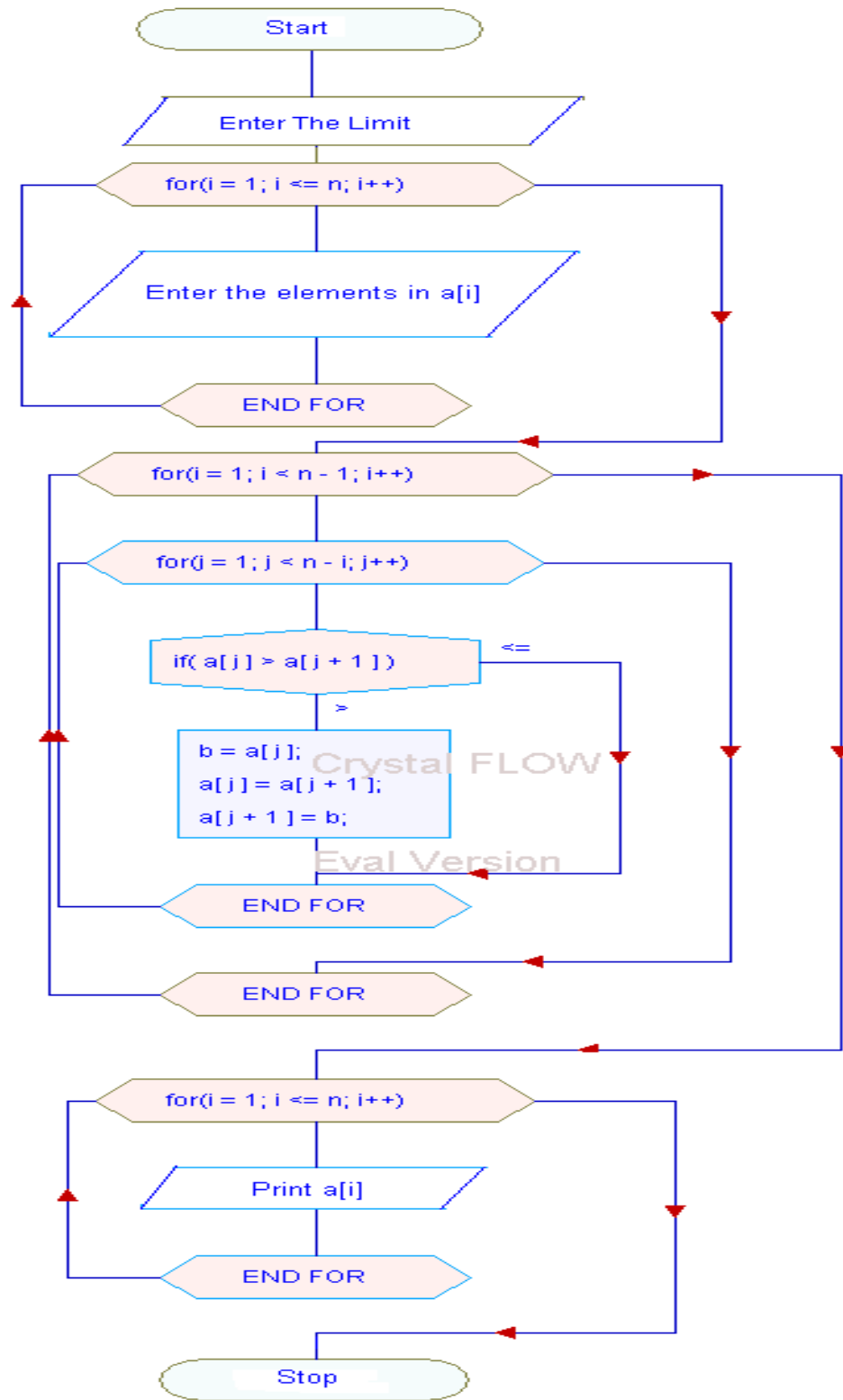
35
10
65
80

10
35
65
80

TRIP-3

During the first trip, three pairs of items are compared and interchanged whenever needed. It should be noted that the number 80, the largest among the items, has been moved to the bottom at the end of first trip. This means that the element 80 need not be considered any further. Therefore TRIP-2 requires only two pairs to be compared. This time the number 65 has been moved down the list. Each trip brings the smallest value 10 up by one level.

The number of steps required in a trip is reduced by one for each trip made. The entire process will be over when a trip contains only one step. If the list contains n elements, then the number of comparisons involved would be $\mathbf{n(n-1)/2}$.



Program :

```
#include<conio.h>
#include<stdio.h>
int main()
{
    int i,j,b,n,a[10];
    clrscr();
    printf("Enter The Limit : ");
    scanf("%d",&n);
    printf("\nEnter array Elements : \n");
    for(i=1;i<=n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=1;i<n-1;i++)
    {
        for(j=1;j<n-i;j++)
        {
            if(a[j]>a[j+1])
            {
                b=a[j];
                a[j]=a[j+1];
                a[j+1]=b;
            }
        }
    }
    printf("\nThe Sorted Elements : \n");
    for(i=1;i<=n;i++)
    printf(" %d ",a[i]);
    getch();
    return 0;
}
```

Input :

Enter The Limit :6

Enter array Elements :

2

5

3

7

6

1

Output :

The Sorted Elements :

1

2

3

5

6

7

Assignment

1. Write a Program to sort a list of values using insertion sort.

Viva-voce questions

1. What would be the output of the following :

a)

```
main()
{
    int num[26], temp;
    num[0] = 100;
    num[25] = 200;
    temp = num[25];
    num[25] = num[0];
    num[0] = temp;
    printf("\n%d %d", num[0], num[25]);
}
```

b)

```
main()
{
    int array[26], i;
    for(i = 0; i <= 25; i++)
    {
        array[i] = 'A' + i;
        printf("\n%d %c", array[i], array[i]);
    }
}
```

2. Which element of array does this expression reference?

num[4]

3. Are the following declaration correct ?

```
int a(25);
int size = 10, b[size];
int c = {0,1,2};
```

Experiment No. 18

Aim : Write a Program to display the sum of two matrices.

Description :

This Program takes two matrices as input and displays their sum as output. This Program also takes number of rows and columns in matrices as input. In this Program variables r and c represents the number of rows and columns respectively. Array variables x and y represents matrices to be added and z represents the resultant matrix . To obtain the result the elements of matrix x are added with the corresponding elements of matrix y .

Therefore the sum of two matrices is given by

$$z[i][j]=x[i][j]+y[i][j]$$

Where i and j are the indexes for rows and columns.

Algorithm:

Step 1: Input the elements of the first matrix.($x[][]$)

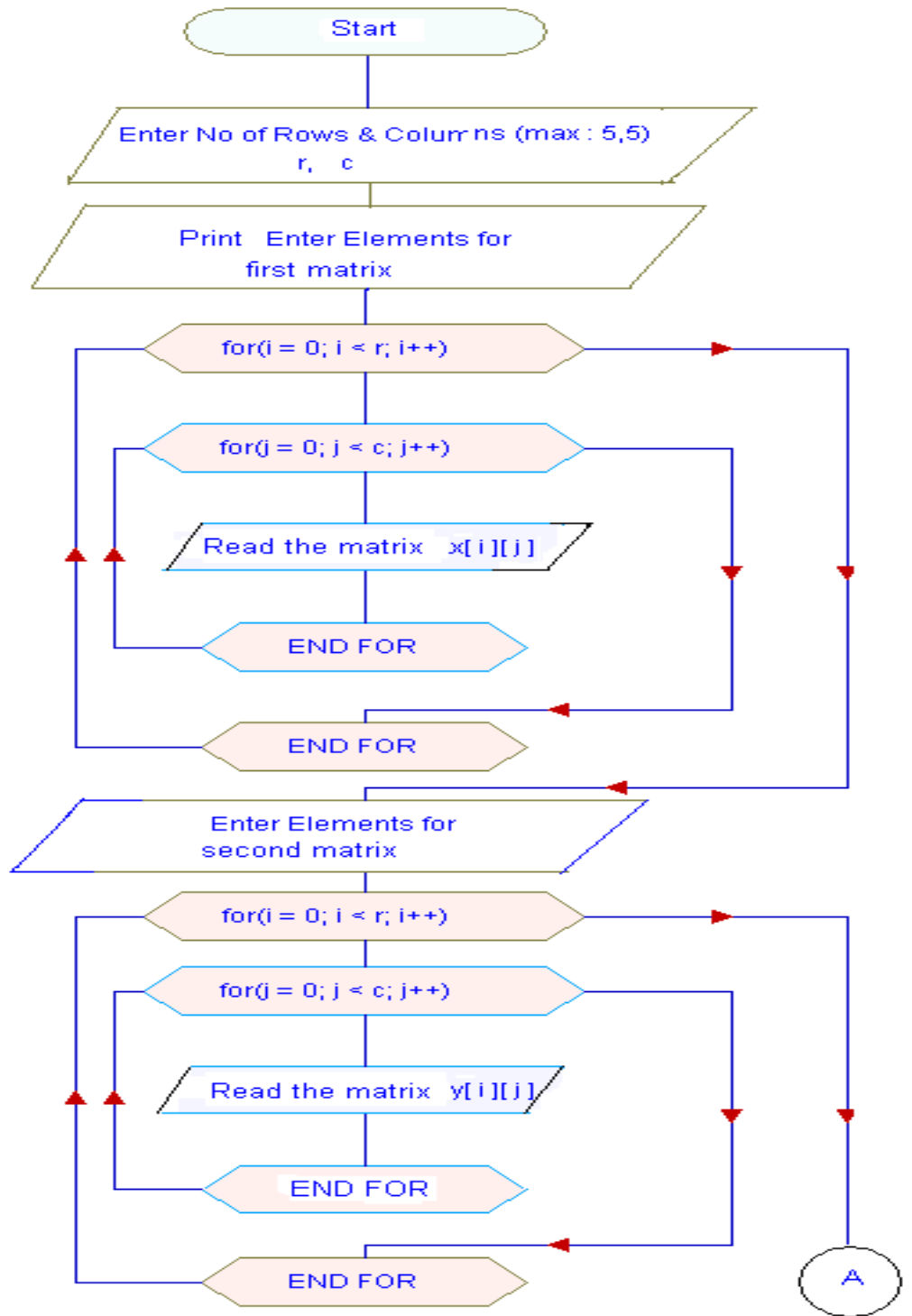
Step 2: Input the elements of the second matrix.($y[][]$)

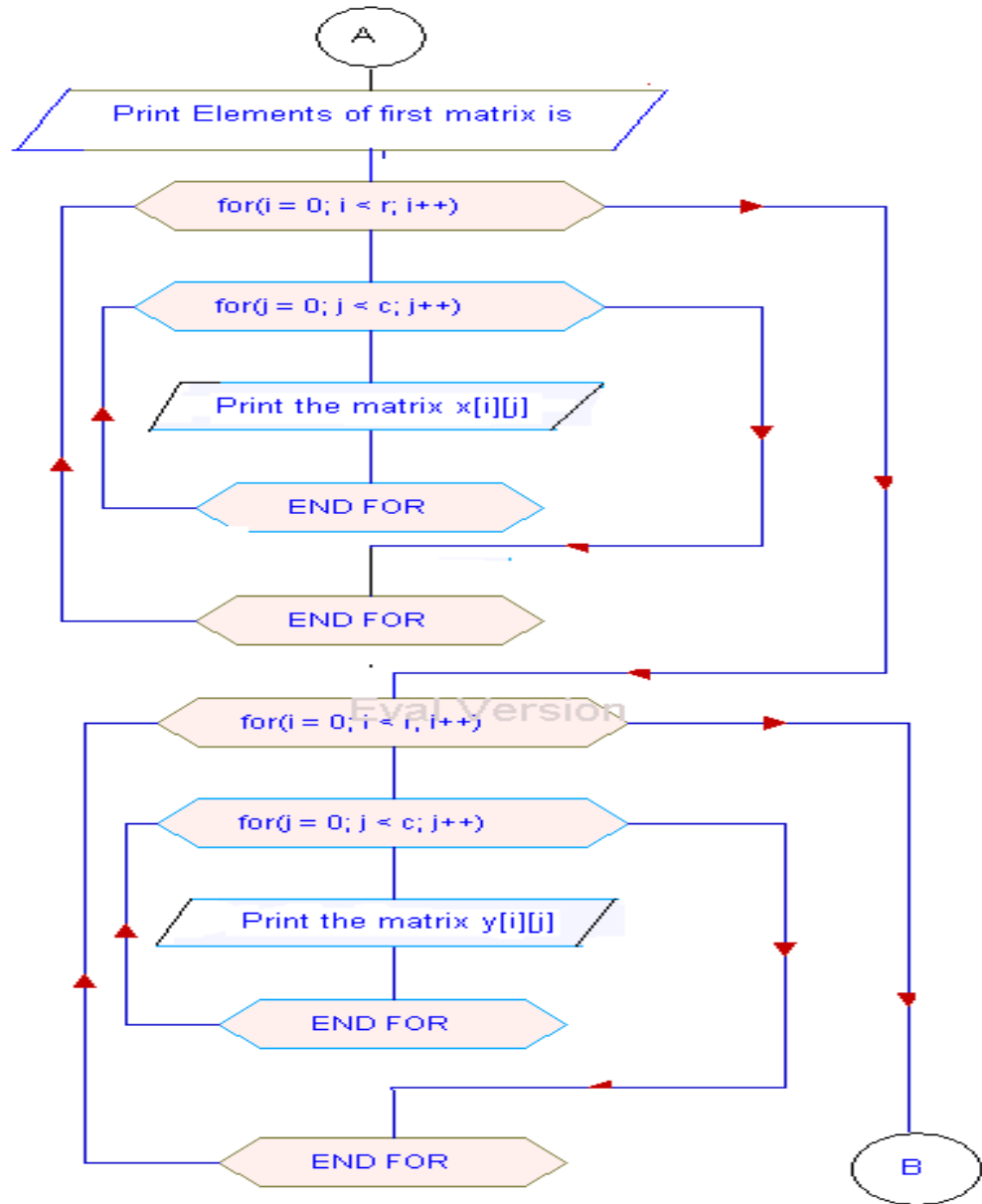
Step 3: Print the elements of the first matrix.

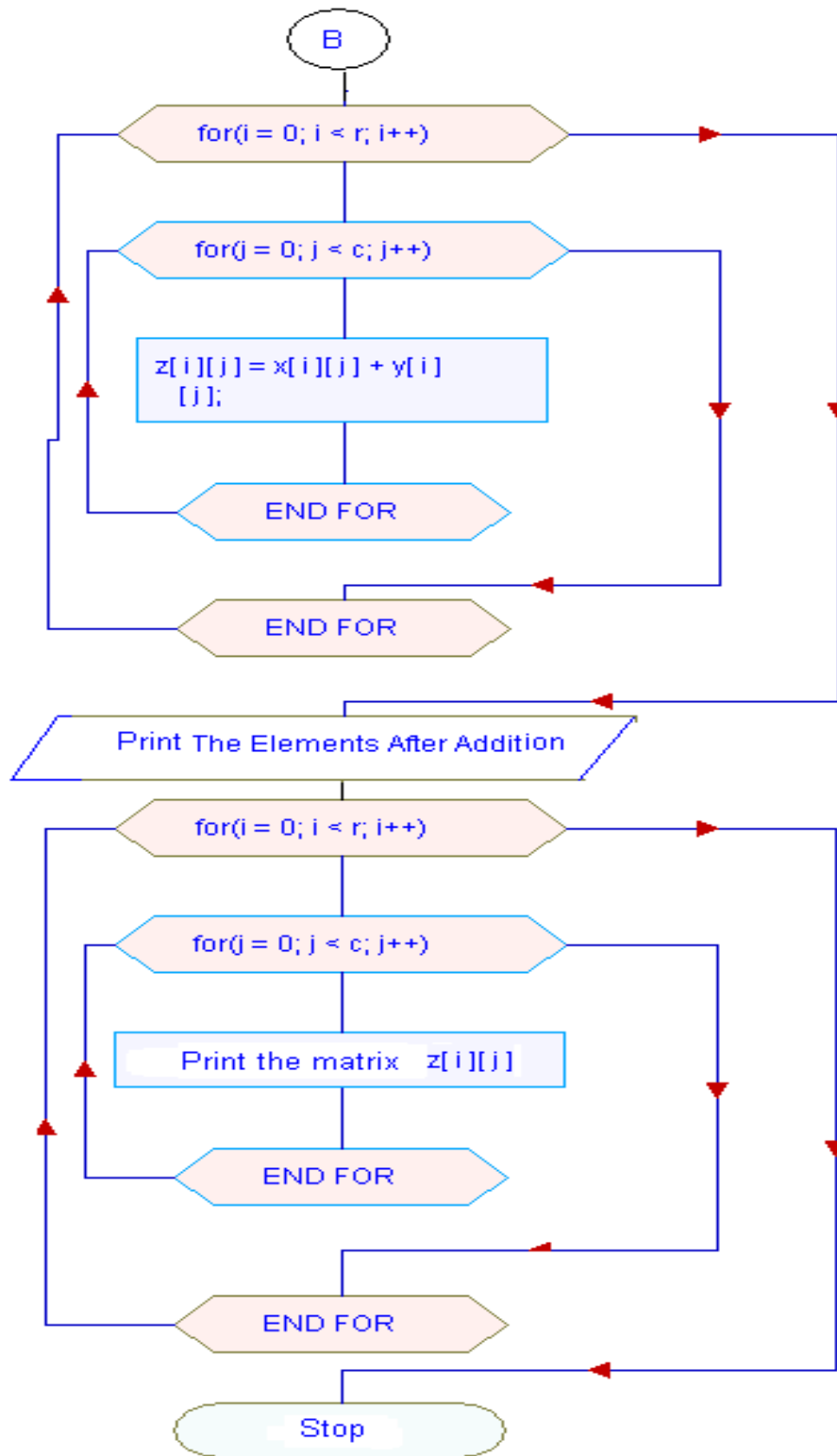
Step 4: Print the elements of the second matrix.

Step 5: Add the elements of the two matrices.

Step 6: Print the sum of the two matrices.($z[][]$)







Program :

```
#include<conio.h>
#include<stdio.h>
int main()
{
    int x[5][5],y[5][5],z[5][5],i,j;
    int r,c;
    clrscr();
    printf("\nEnter No of Rows & Columns (max : 5,5) :");
    scanf("%d%d",&r,&c);
    printf("\nEnter Elements for 1st Matrix : ");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            scanf("%d",&x[i][j]);
    }
    printf("\nEnter Elements for 2nd Matrix : ");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            scanf("%d",&y[i][j]);
    }
    printf("\nThe Elements In 1st Matrix : \n\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            printf(" \t%2d ",x[i][j]);
        printf("\n");
    }
    printf("\nThe Elements In 2nd Matrix : \n\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            printf(" \t%2d ",y[i][j]);
```

```

        printf("\n\n");
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            z[i][j]=x[i][j]+y[i][j];
    }
    printf("\nThe Elements After Addition : \n\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            printf(" \t%2d ",z[i][j]);
        printf("\n");
    }
    getch();
    return 0;
}

```

Input :

Enter No of Rows & Columns (max : 5,5) :2 3

Enter Elements for 1st Matrix :

1 2 3

2 3 4

Enter Elements for 2nd Matrix :

2 3 4

5 6 7

Output :

The Elements In 1st Matrix :

1 2 3

2 3 4

The Elements In 2nd Matrix :

2 3 4

5 6 7

The Elements After Addition :

3 5 7

7 9 11

Assignment

1. Write a Program to multiply two 3*3 matrices.

Viva-voce questions

1. What will be the output of the following Program :

```
main()
{
    static int a[3][3] = {
        1, 2, 3,
        4, 5, 6,
        7, 8, 9
    };

    static int *ptr[3] = { a[0], a[1], a[2] };
    int **ptr1 = ptr ;
    int i ;
    printf ( "\n" );

    for ( i = 0 ; i <= 2 ; i++ )
        printf ( "%d ", *ptr[i] );

    printf ( "\n" );
    for ( i = 0 ; i <= 2 ; i++ )
        printf ( "%d ", *a[i] );

    printf ( "\n" );
    for ( i = 0 ; i <= 2 ; i++ )
    {
        printf ( "%d ", **ptr1 );
        ptr1++;
    }
}
```

2. What do you mean by multidimensional array? Explain the syntax to define it.

3. How can we initialize a multidimensional array? Explain.

Experiment No. 19

Aim : Write a Program to read and display a line of text using getchar and putchar functions.

Description :

This Program reads a string character by character as input and prints it character by character as output. In this Program **c** represents the character read from the keyboard and **str** represents the string in which these characters are stored. The function **getchar** is used to read the string character by character from the keyboard till the new line character (**'\n'**) is encountered and the function **putchar** is used to print it character by character on the screen till the null character (**'\0'**) is encountered.

Algorithm:

Step 1: Initialize $i=0$.

Step 2: Print 'Enter a string'.

Step 3: Repeat Steps 4, 5 until the user presses the input key.

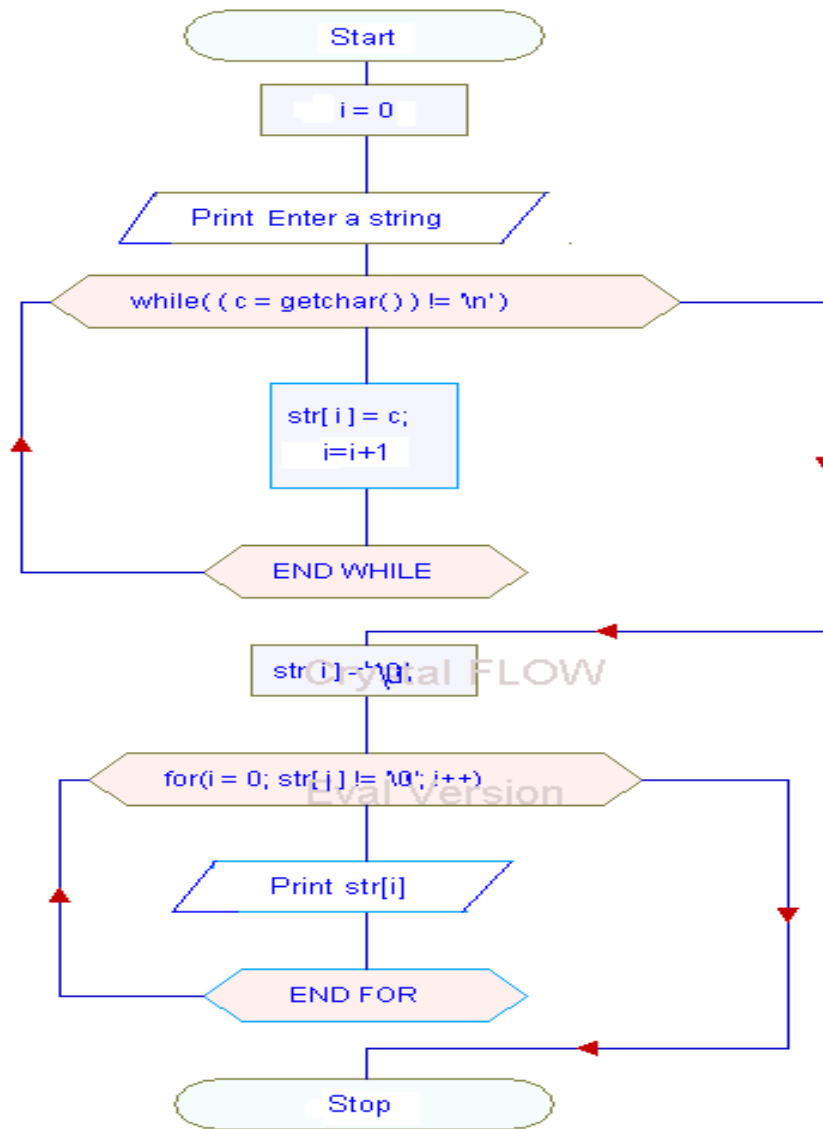
Step 4: Input a character and store it in $str[i]$.

Step 5: Increment the value of i .

Step 6: Assign NULL character to $str[i]$.

Step 7: Repeat Step 8 till NULL character is reached.

Step 8: Print the characters in str .



Program :

```
#include <stdio.h>
#include <conio.h>
void main( )
{
    char str[ 30 ], c;
    int i = 0;
    printf( "Enter a string:" );
    while( ( c = getchar( ) ) != '\n' )
    {
        str[ i ] = c;
        i++;
    }
    str[ i ] = '\0';
    printf( "\nThe string is :");
    for(i = 0; str[ i ] != '\0'; i++)
        putchar( str [ i ] );
    getch( );
}
```

Input :

Enter a string:this is a character array

Output :

The string is :this is a character array

Assignment

1. Write a Program to calculate the number of commas and number of tab characters in a string.

Viva-voce questions

1. What do you mean by string?
2. Describe the limitation of using getchar and scanf functions for reading string.
3. What would be the output of the following :

```
main()  
{  
    printf( 5 + "Fascimile");  
}
```

- a) Error
 - b) Fascimile
 - c) mile
 - d) None of the above
4. How would you output \n on screen?
 5. What would be the output of the following?

```
main()  
{  
    char ch = 'A';  
    printf("%d%d", sizeof(ch), sizeof('A'));  
}
```

- a) 1 1
- b) 1 2
- c) 2 2
- d) 2 1

Experiment No. 20

Aim : Write a Program take 5 names as input and print the longest name.

Description :

This Program takes a list of five names as input and displays the name with the largest number of characters as output. In this Program variable **name** represents the array of five names. To obtain the longest name the length of the names are compared with each other one by one in a loop that is executed five times, and after the complete execution of loop we get the length and index of the longest name that is then used to print the longest name.

Algorithm:

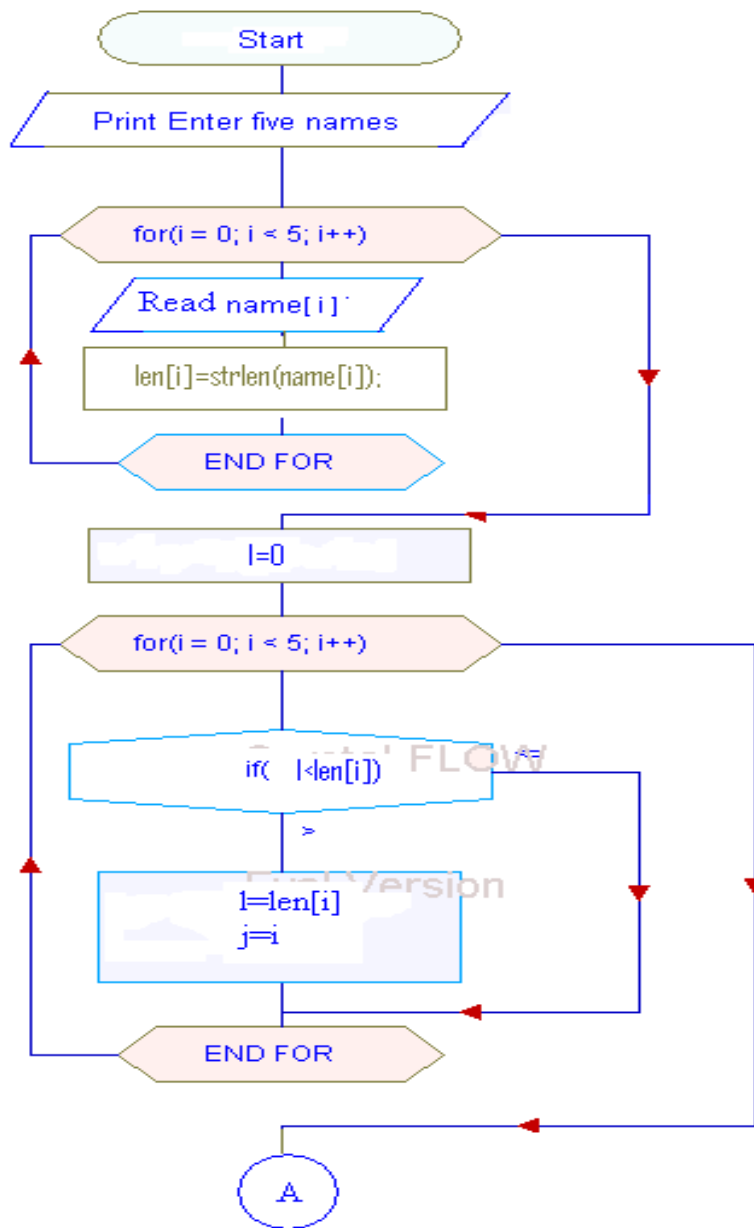
Step 1: Input any five names (name[5]) and store the length of each name(len[5]).

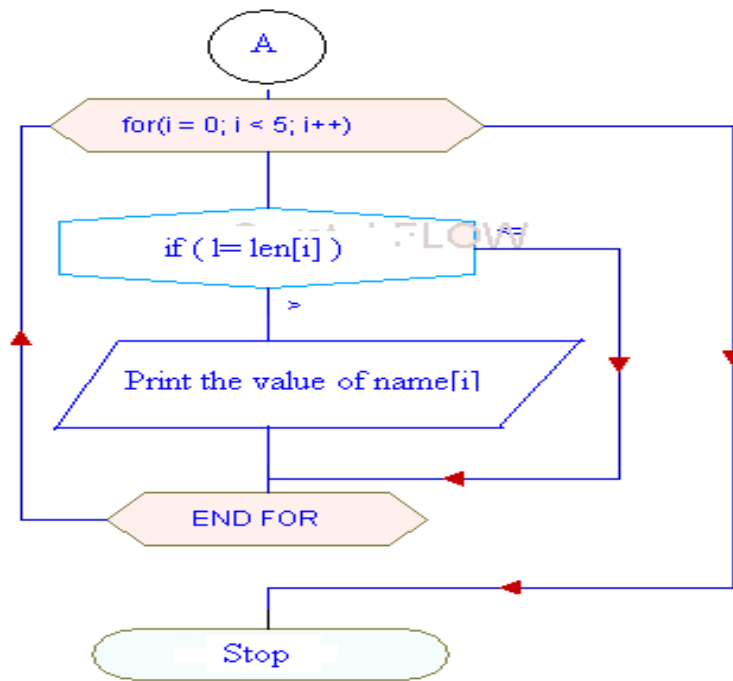
Step 2: Initialize l=0.

Step 3: Store the length of longest name in l.

Step 4: Compare l with the lengths of all the elements of the array name.

Step 5: As it matches the name with the corresponding length is printed as the longest name.





Program :

```

#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
    char name[5][15];
    int l,len[10],large[20],j=0,i;
    clrscr();
    printf("Enter five names:");
    for(i=0;i<5;i++)
    {
        gets(name[i]);
        len[i]=strlen(name[i]);
    }
}

```

```

        l=0;
        for(i=0;i<5;i++)
        {
            if(l<len[i])
            {
                l=len[i];
                j=i;
            }
        }
        printf("\nLongest name(s) is :");
        for(i=0;i<5;i++)
        {
            if(l==len[i])
                printf("%s\n",name[i]);
        }
        getch();
    }

```

Input :

Enter five names:

Dennis Ritchie

B Ram

Peterson

Adam

Sam

Output :

Longest name(s) is : Dennis Ritchie

Assignment

1. Write a Program to check whether the input string is palindrome or not using `strrev` and `strcmp` functions.
2. Write a Program which reads your name from the keyboard and outputs a list of ASCII codes which represent your names.

Viva –Voce Questions

1. In which file the string manipulation functions are defined?
 - a) `stdlib.h`
 - b) `string.h`
 - c) `stdio.h`
 - d) `conio.h`
2. If we read entire line “NEW YORK” with the following statement
`scanf(“%s%s”, adr1, adr2);`
then what would be the values of `adr1` and `adr2`?
3. How can we initialize a string?
4. What is the difference between `scanf` and `gets` function?
5. If the value of string `city` is `NEW YORK` then
What would be the output of the following :
 - a) `printf(“%10.4s”, city);`
 - b) `printf(“%-10.4s”, city);`
 - c) `printf(“%4s”, city);`
 - d) `printf(“%10s”, city);`

Experiment No. 21

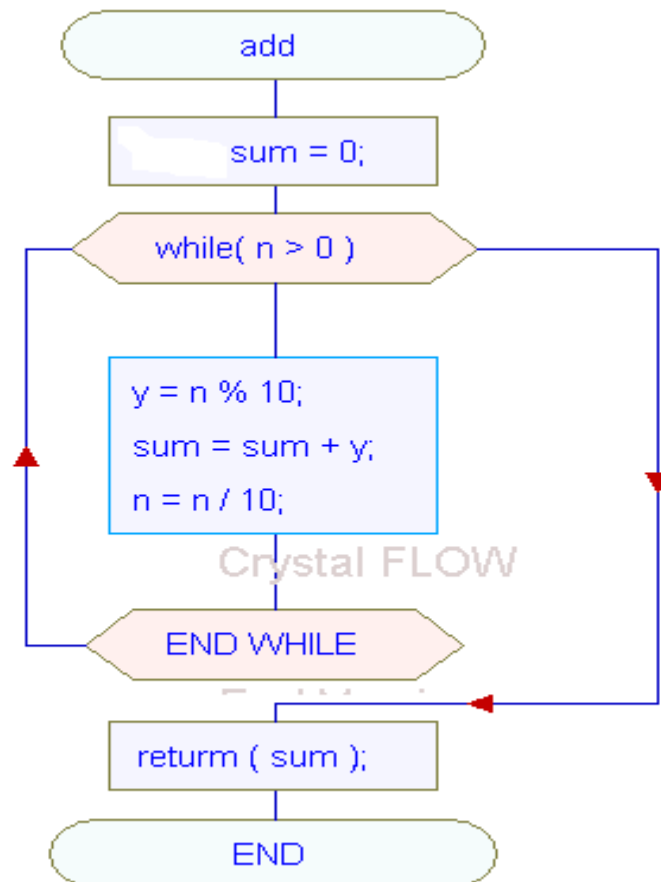
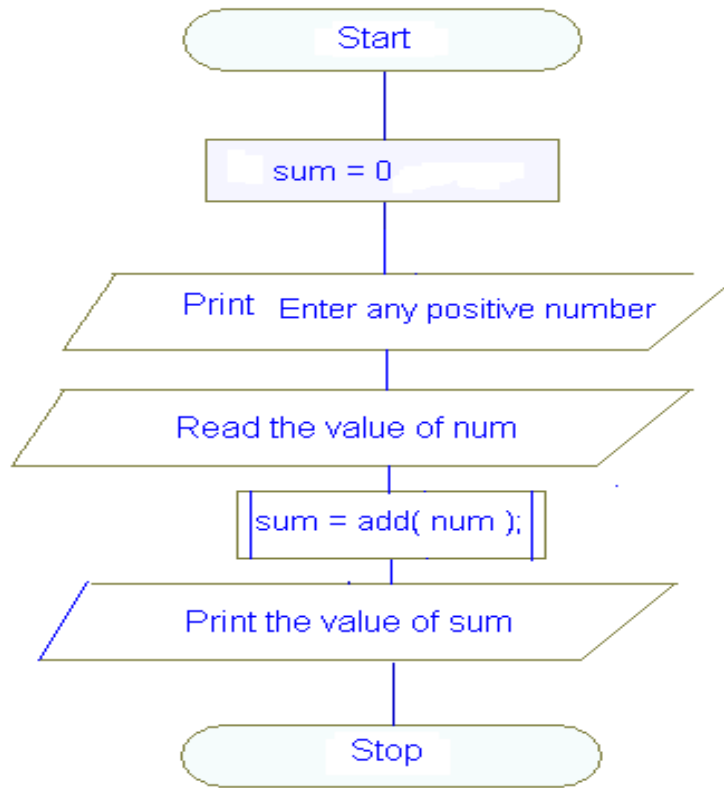
Aim : Write a Program that will take an integer as argument and return the sum of its digits using a user defined function.

Description :

In this Program the variable **n** represents number whose sum of digits is to be calculated and **sum** represents the sum of its digits. To calculate sum number is divided with ten till its value is greater than zero and the remainder of the division are added to find the required sum.

Algorithm:

- Step 1: Initialize n=0.
- Step 2: Input any number num
- Step 3: Call the function add passing num as parameter.
- Step 4: Inside the function add
 - a. Initialize sum=0.
 - b. Calculate the sum of digits in the variable sum.
 - c. Return the value of the variable sum to the calling function.
- Step 5: Print the value of the sum received from the called function.



Program :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int sum=0,num;
    int add(int);
    clrscr( );
    printf("Enter any positive number :");
    scanf("%d",&num);
    sum=add(num);
    printf("\nSum of the digits of given number is :%d",sum);
    getch( );
}
int add(int n)
{
    int y,sum=0;
    while(n>0)
    {
        y=n%10;
        sum=sum+y;
        n=n/10;
    }
    return(sum);
}
```

Input :

Enter any positive number :8228

Output :

Sum of the digits of given number is :20

Assignment

1. Write a Program to perform addition, subtraction, multiplication and division using a separate function for each.
2. Distinguish between following :
 - a) Actual and formal arguments.
 - b) Global and local variables.
 - c) Automatic and static variables.
 - d) Scope and visibility of variables.
 - e) & operator and * operator.

Viva-voce questions

1. What do you mean by functions?
2. The main is user defined function. How does it differ from other user-defined functions
3. What is prototyping? Why it is necessary?

Experiment No. 22

Aim : Write a Program to swap two integer values using call by reference.

Description:

In **call by reference** addresses of variables are passed to functions as arguments and pointers are used to access their values in the function. This Program takes two integers **a**, **b** as input and displays them after swapping their values as output. This Program consist of a function **swap** to which the addresses of **a**, **b** are passed as arguments and it interchanges their values using pointer variables **p** and **p1**. The pointer variables **p** and **p1** contains the addresses of **a** and **b** passed to function **swap** respectively.

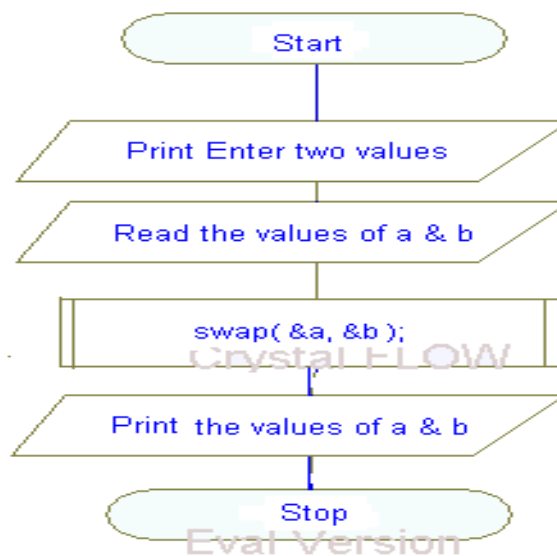
Algorithm:

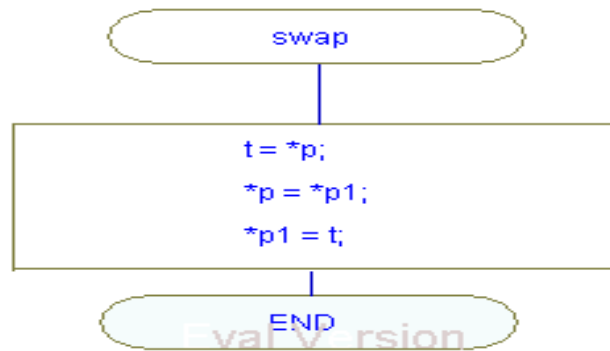
Step 1: Input any two values (a, b)

Step 2: Call the function **swap** by passing the addresses of two variables.

Step 3: Inside the function exchange the values of two addresses using a third pointer variable.

Step 4: Print the values of original variables whose addresses were passed.





Program :

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    void swap(int *,int *);
    printf("Enter two values :");
    scanf("%d%d",&a,&b);
    printf("Values before swapping: a=%d,b=%d",a,b);
    swap(&a,&b);
    printf("Values after swapping : a=%d, b=%d",*p,*p1);
    getch( );
}
void swap(int *p,int *p1)
{
    int t;
    t=*p;
    *p=*p1;
    *p1=t;
}
  
```

Input :

Enter two values :4 8

Output :

Values before swapping: a=4,b=8

Values after swapping : a=8, b=4

Assignment

1. Write a Program to generate the multiplication table using call by reference.

Viva-voce questions

1. Differentiate call-by-reference and call-by-value?
2. What do you mean by address operator?

Experiment No. 23

Aim : Write a Program to print the following Fibonacci series using recursive function :

1 1 2 3 5 8 13

Description : This Program takes number of terms as input and prints the Fibonacci series upto that number of terms. Here variable **n** represents number of terms and **a, b** are the first two terms of the series. This Program makes recursive call to function **febo** to print the series.

Algorithm:

Step 1: Input the number of terms required in Fibonacci series (n).

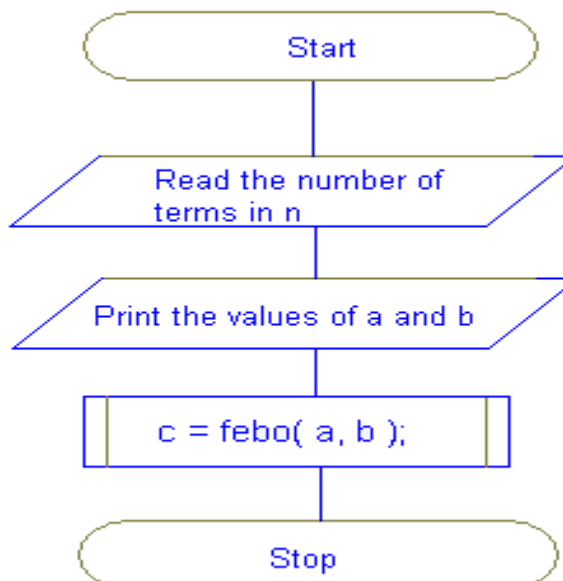
Step 2: Initialize the two variables a=1 and b=1.

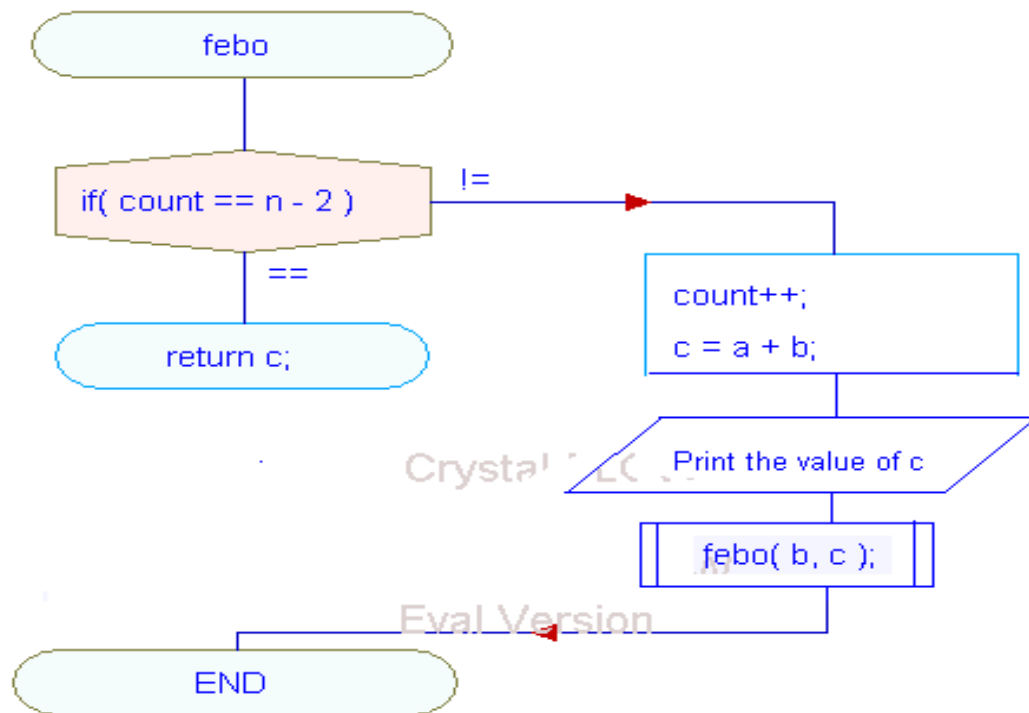
Step 3: Print the two values.

Step 4: Call the function fibo passing the values of a and b.

Step 5: Inside the function

- a. Repeat steps b, c, d and e until the value of count is equal to n-2.
- b. Store the sum of values a and b in c.
- c. Increment the value of count.
- d. Print the value of c.
- e. The function should call itself passing the values of b and c as parameters.
- f. Return the value of c to the calling function.





Program :

```

#include<stdio.h>
#include<conio.h>
int count=0,a=1,b=1,n,c,f;

```

```

void main()
{
    int sum;
    int febo(int,int);
    clrscr();
    printf("Enter the number of terms:");
    scanf("%d",&n);
    printf("%d\t%d\t",a,b);
    c=febo(a,b);
    getch();
}
febo(int a,int b)
{
    if(count==n-2)
    return c;
    else
    {
        count++;
        c=a+b;
        printf("%d\t",c);
        febo(b,c);
    }
}

```

Input :

Enter the number of terms:6

Output :

1 1 2 3 5 8

Assignment

1. Write a Program using recursive function to calculate factorial of a number

Viva-voce questions

1. What do you mean by the term recursion?
2. Why recursion is considered to harmful in some cases?

Experiment No. 24

Aim : Write a Program to print first n even numbers using recursive function.

Description : This Program takes the number of terms up to which the even series is to be displayed as input and displays the series containing only even numbers as output. In this Program the variable **n** represents the number of terms and **eve** is a function that is passed $2 * n$ i.e. the maximum value in the series as argument. **eve** is recursive a function that returns the terms of series. The recursive call is terminated when the value of **n** becomes **2** and in that case function returns **2** and terminated.

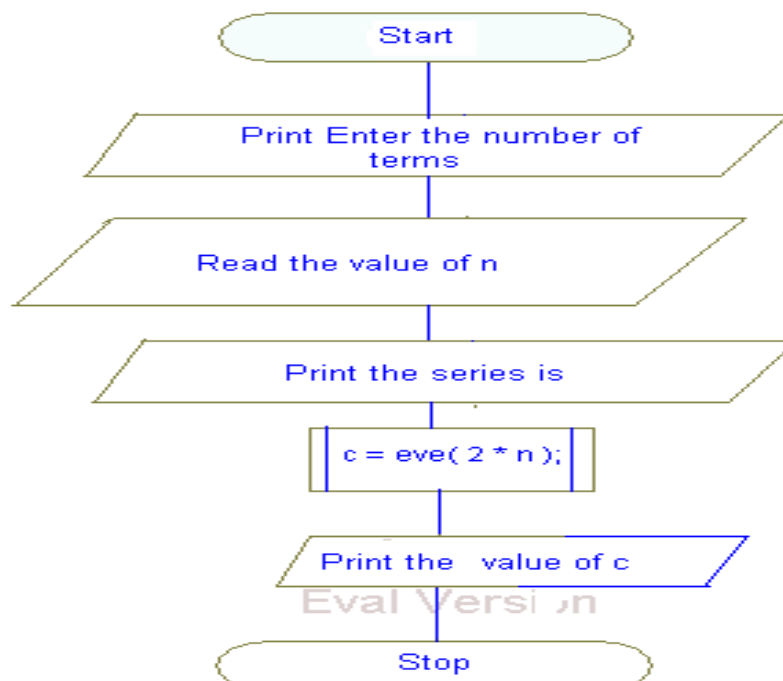
Algorithm :

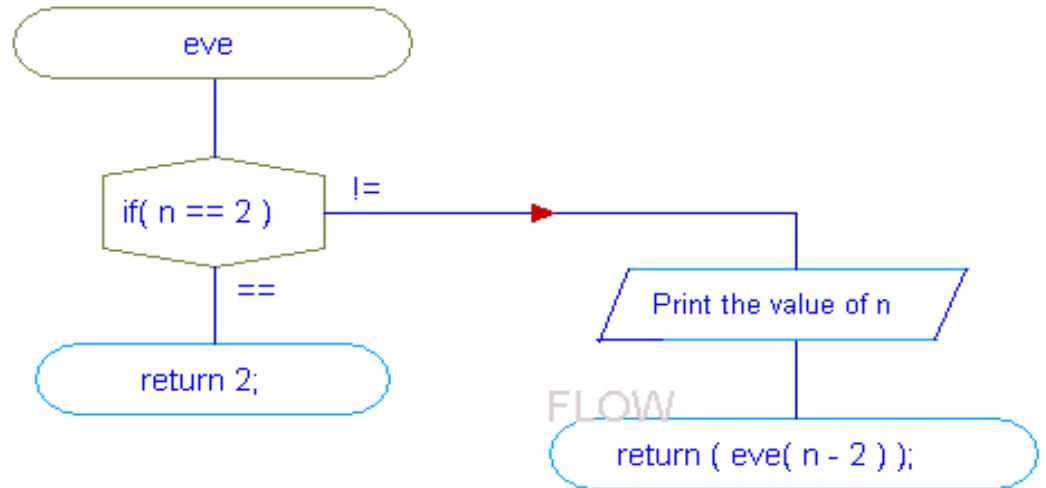
Step 1 : Read the number of terms (ex n)

Step 2 : Call the function eve with argument $2*n$

- a. Check the value of n if it is 2 then return 2
- b. Print the value of n and call the function again with n-2

Step 3 : Store the returned value in c and print the value of





Program :

```

#include<stdio.h>
#include<conio.h>
int a=0;
void main()
{
    int n,c;
    int eve(int);
    clrscr();
    printf("Enter number of terms :\n");
    scanf("%d",&n);
    printf("\nSeries is :\n");
    c=eve(2*n);
    printf("%d",c);
    getch();
}
int eve(int n)
{
    if(n==2)

```



```
        return 2;
    else
    {
        printf("%d\n",n);
        return (eve(n-2));
    }
}
```

Input :

Enter number of terms : 6

Output :

Series is :

12

10

8

6

4

2

Assignment

1. Write a Program using recursion to evaluate $x - x^3/3! + x^5/5! \dots$ using recursive function.

Viva –Voce Questions

1. Write down the difference between iterative and recursive functions.
2. Is recursive Program is better with respect to time and space than a non recursive Program.

Experiment No. 25

Aim : Write a Program to define a structure student that will contain the roll number, name and total marks of a student the Program will ask the user to input details of 6 students and print the details of all the students whose total marks is greater than a given value.

Description :

This Program takes the details of a list of students as input and displays the details of the students whose marks is greater than 65. In this Program **student** is a structure that contains following members:

roll : Its type is integer and this member represents Roll number of a student.

name : **name** is a character array and it represents Name of a student.

marks : Its type is float and it represents Marks of a student.

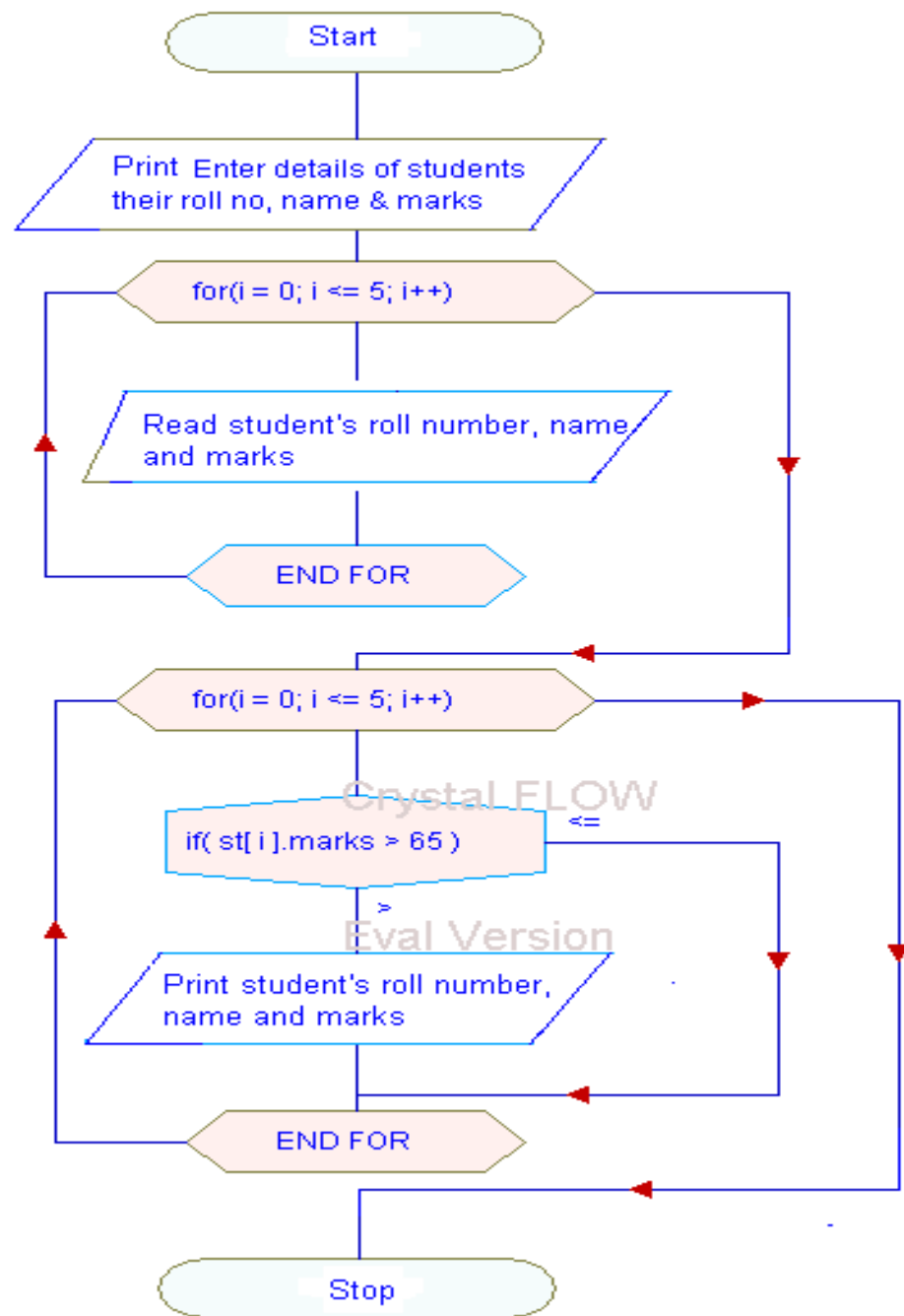
st : **st** represents the array of structure that is used to store details of students.

Algorithm:

Step 1 : Input the roll number , name and marks of 6 students.

Step 2: Check the marks of each student.

Step 3: If marks of any student is greater than or equal to 65 then print student's rollnumber , name and marks.



Program :

```
#include<stdio.h>
#include<conio.h>
void main( )
{
    struct student
    {
        int roll;
        char name[10];
        float marks;
    }st[5];
    int i;
    printf("Enter details of students their roll no, name & marks:");
    for(i=0;i<=5;i++)
        scanf("%d%s%f",&st[i].roll,st[i].name,&st[i].marks);
    for(i=0;i<=5;i++)
    {
        if(st[i].marks>65)
        {
            printf("%d %s %f",st[i].roll,st[i].name,st[i].marks);
        }
    }
    getch( );
}
```

Input :

Enter details of students their roll no, name & marks:

101 C 65

102 XYZ 75

119 DEF 55

103 GHI 70

104 HJY 80

Output :

102 XYZ 75.000000

103 GHI 70.000000

104 HJY 80.000000

Assignment

1. Define a structure called cricket that will describe the following information

player name

Team name

Batting average

Using cricket declare an array player with 50 elements and write a Program to read the information about all the 50 players and print a team wise list containing names of players with their batting average.

2. Point out errors, if any, in the following Program :

```
main()
{
    struct employee
    {
        char name[25];
        int age;
        float bs;
    };
    struct employee e;
    strcpy(e.name,"Hacker");
    age = 25;
    printf("\n%s%d",e.name,age);
}
```

Viva-voce questions

1. What is structure?
2. What would be the output of the following Programs :

```
a) main()
{
    struct gospel
    {
        int num;
        char mess1[50];
        char mess2[50];
    }m;
    m.num = 1;
    strcpy(m.mess1, "If at all that you have is hammer");
    strcpy(m.mess2, "Everything looks like a nail");
    printf("\n%u%u%u", &m.num, m.mess1, m.mess2);
}
b) struct gospel
{
    int num;
```

```

    char mess1[50];
    char mess2[50];
}m1 = {2, "If you are driven by success", "make sure that it is a quality drive"};
main()
{
    struct gospel m2,m3;
    m2 = m1;
    m3 = m2;
    printf("\n%d%s%s", m1.num, m2.mess1, m3.mess2);
}

```

3. Ten floats are to be stored in memory. What would you prefer, an array or a structure?

4. Given the statement :

```
maruti.engine.bolts = 25;
```

which of the following is True?

- a) structure bolts is nested within structure engine
- b) structure engine is nested within structure maruti
- c) structure maruti is nested within structure engine
- d) structure maruti is nested within structure bolts

Experiment No. 26

Aim : Write a Program to define a union - contact that will contain the members mobile no. and e-mail id. Now define a structure - employee that will contain name, roll no., mode of contact and a variable of type contact as members. The Program will ask the user to give the details of two employees including mode of contact and the contact num/e-mail. Print the details of both the employees.

Description :

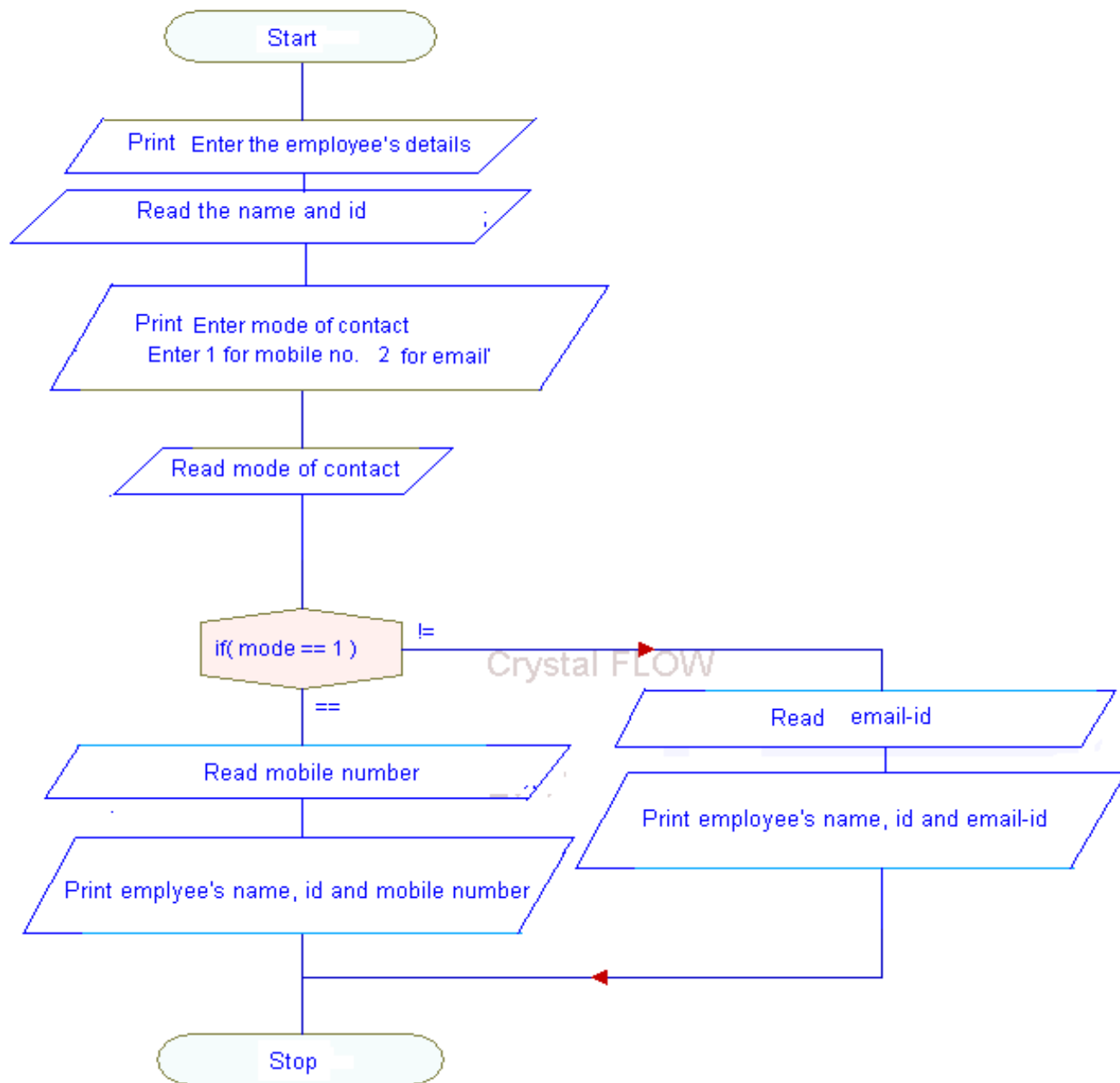
This Program takes details of employees as input, asks the user to enter the mode of contact and displays the details according to the mode of contact as output. In this Program **contact** is a **union** and has two members **mobilenno**, **email** where **mobilenno** represents Mobile number of an employee and **email** represents the Email-id.

employee is a **structure** that has the following members :

name : name is a character array that represents name of employee.

rollno : Represents employee id.

mode_of_contact : It is a variable of type contact and represents Mode of contact.



Program :

```
#include<stdio.h>
#include<conio.h>
union contact
{
    char mobileno[10];
    char email[15];
};
struct employee
{
    char name[15];
    int rollno;
    union cotact mode_of_contact;
};

void main( )
{
    struct employee emp;
    int mode;
    printf("Enter the employee's details :\n");
    scanf("%s%d",emp.name,&emp.rollno);
    printf("Enter mode of contact :\n");
    printf("Enter 1 for mobile no.\n2 for email\n");
    scanf("%d",&mode);
    if(mode==1)
    {
        printf("Enter Mobile number:");
        scanf("%s",emp.mode_of_contact.mobileno);

        printf("%s\t%d\t%s\n",emp.name,emp.rollno,emp.mode_of_contact.mobileno);
    }
    else
    {
        printf("Enter Email-id:");
```

```

scanf("%s",emp.mode_of_contact.email);

printf("%s\t%d\t%s\n",emp.name,emp.rollno,emp.mode_of_contact.mobilenos);
}
getch( );
}

```

Output :

Enter the employee's details :

XYZ 101

Enter mode of contact :

Enter 1 for mobile no.\n2 for email

1

Enter Mobile number:9827122222

XYZ 101 9827122222

Assignment

1. Declare a union number having members integer and float. Take input a number that will be either of type int or float and print it.

Viva-voce questions

1. What is union?
2. Differentiate structure and union?
3. When do we use the following :
 - a) Unions.
 - b) Bit fields.
 - c) The sizeof operator.

Experiment No. 27

Aim : Write a Program to find the greatest of three numbers using macro.

Description : This Program takes three integers **a**, **b**, **c** as input and displays the largest among them as output. In this Program a macro **max** is defined using # define preprocessor directive to which the values are passed as arguments and it returns the largest value among them. The variable **k** represents the larger value among **a** and **b** and the variable **j** represents the final largest value.

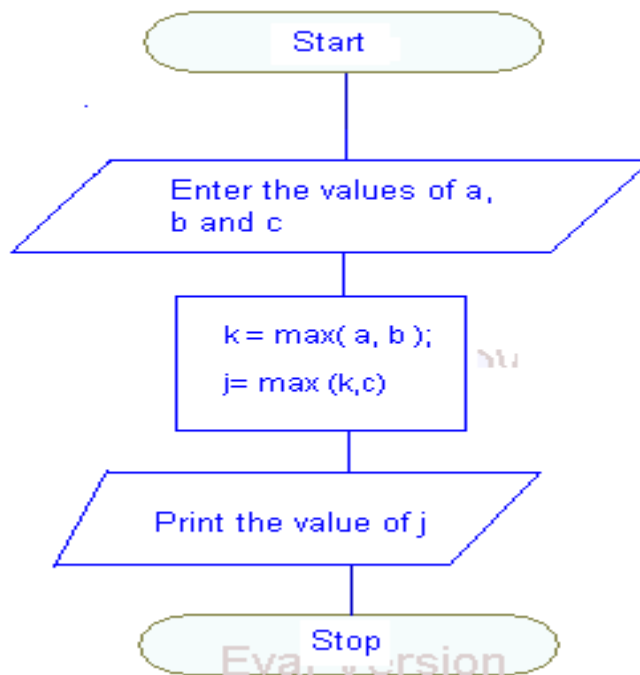
Algorithm:

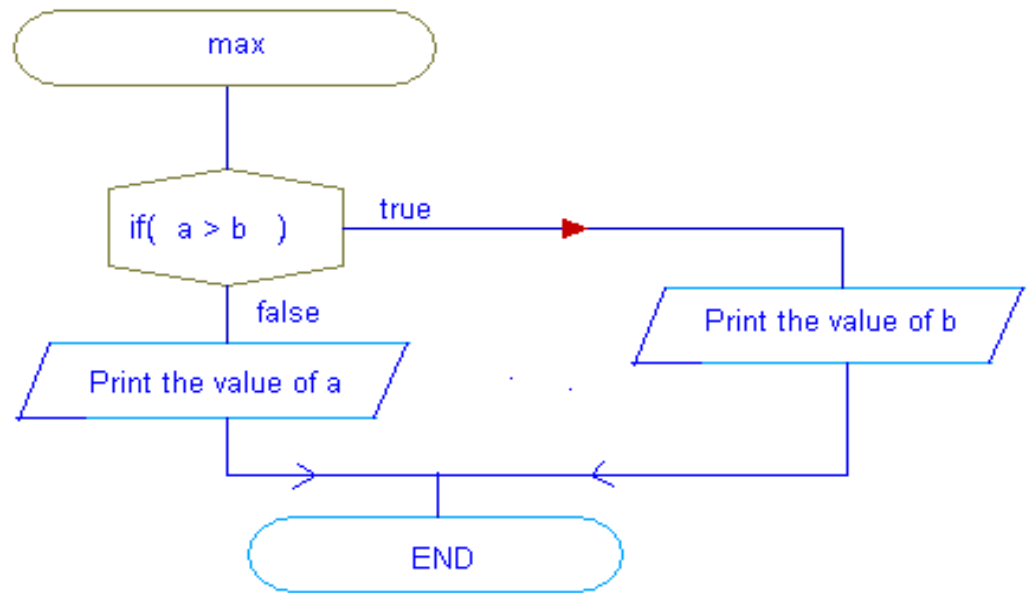
Step 1: Input the values of any three numbers(a,b,c)

Step 2: Calculate maximum of a and b using macro (max) and store it in any variable(k).

Step 3: Calculate maximum of k and c using macro (max) and store it in any variable(j).

Step 4: Print the value of j .





Program :

```
#define max(a,b) (a>b)?a:b
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b,c,k,j;
    clrscr( );
    printf("Enter three values:");
    scanf("%d%d%d",&a,&b,&c);
    k=max(a,b);
    j=max(k,c)
    printf("\nGreatest number is :%d",j);
    getch();
}
```

Output :

Enter three values : 4 7 9

Greatest number is : 9

Assignment

1. Write a Program to calculate square of a number using macro.

Viva-voce questions

1. What do you mean by preprocessor directive?
2. Why macros are considered to be harmful for long Programs?

Experiment No. 28

Aim : Write a Program using pointers to read an array of integers and print its elements in reverse order.

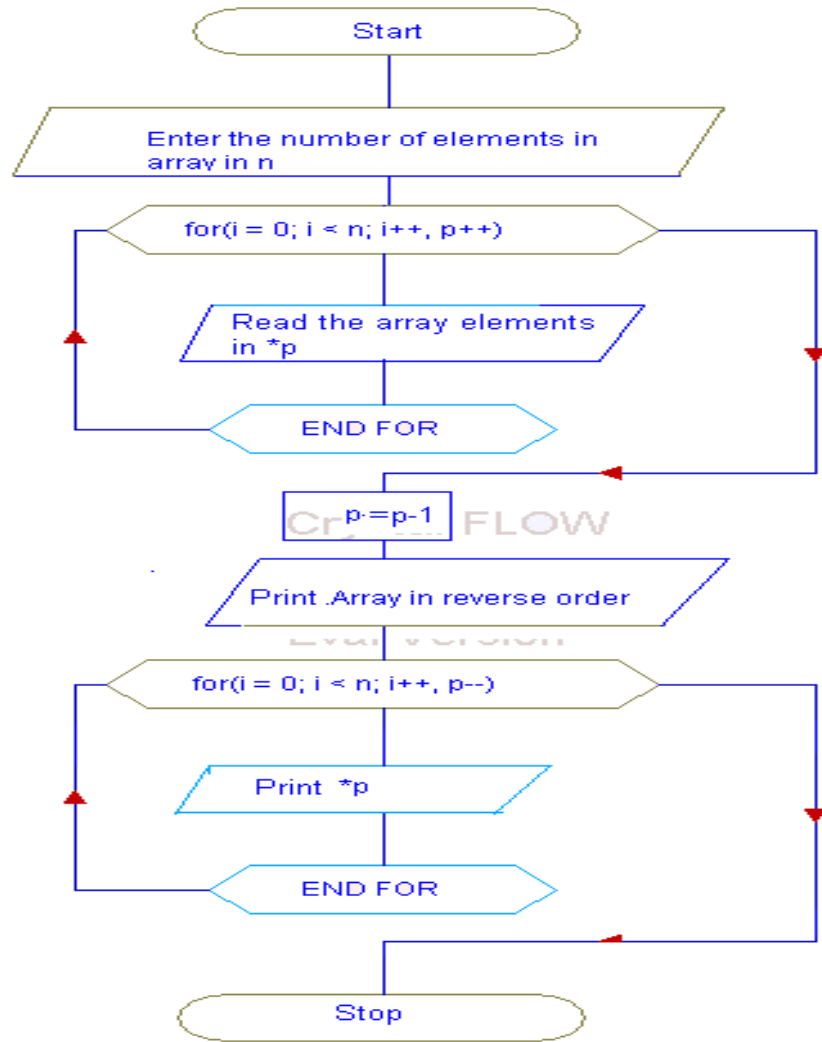
Description : This Program takes an array as input and displays the array in reverse order as output. In this Program variable **p** represents the pointer variable through which the array is read from the keyboard and displayed in reverse order. To read the array pointer variable is incremented by 1 after reading each element and to print the array in reverse order the pointer **p** is decremented by one and the value pointed by it is printed.

Algorithm:

Step 1: Input the number of elements to be inserted (ex n)

Step 2: Using for-loop input the elements of the array.

Step 3: Print the elements of the array in reverse order using for loop.



Program :

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10],i,*p,n;
    clrscr();
    printf("Enter the number of elements in array:");
    scanf("%d",&n);
    printf("Enter the elements of array:");
    for(i=0;i<n;i++,p++)
        scanf("%d",&*p);
  
```

```
        p--;  
        printf("\nArray in reverse order :\n");  
        for(i=0;i<n;i++,p--)  
            printf("%d\n",*p);  
        getch();  
    }
```

Input :

Enter the number of elements in array:5

Enter the elements of array:

1

2

3

4

5

Output :

Array in reverse order :

5

4

3

2

1

Assignment

1. Write a function “replace” which takes a pointer to a string as a parameter, which replaces all spaces in that string by minus signs, and delivers the number of spaces it replaced.

Viva-voce questions

2. What is a pointer?
3. How is a pointer initialized?
4. Distinguish between (*m)[5] and *m[5].
5. What is void pointer?
6. What do you mean by null pointer?
7. How is a pointer variable declared?
8. Are the expressions *ptr ++ and ++ *ptr same?
9. In the following Program how would you print 50 using p?

```
main()  
{  
    int a[] = {10, 20, 30, 40, 50};  
    char *p;  
    p = (char *)a;  
}
```

10. Would the following Program compile :

```
main()  
{  
    int a = 10, *j;  
    void *k;  
    j = k = &a;  
    j++;  
    k++;  
    printf(“\n%u%u”, j, k);  
}
```

11. How is a variable’s address determined?

12. What will be the output of the following :

```
main()  
{  
    int arr[ ]={12 ,13, 14, 15, 16};  
    printf(“\n%d %d %d”, sizeof(arr), sizeof(*arr), sizeof(arr[0]));  
}
```

Experiment No. 29

Aim : Write a Program to read data from the keyboard, write it to a file called INPUT, again read the same data from the INPUT file, and display it on the screen.

Description : This Program takes input from the keyboard as input and writes it, character by character to the file **INPUT**. The end of the data is indicated by entering an **EOF** character, which is **control-Z** in the reference system. Then the file **INPUT** is closed. The file **INPUT** is again opened for reading. The Program reads its contents character by character and displays it on the screen. Reading is terminated when **getc** encounters the end-of-file mark EOF.

Algorithm:

Step 1: Assign a file pointer to the file INPUT

Step 2: Take input from the keyboard using getchar function till the EOF character is entered.

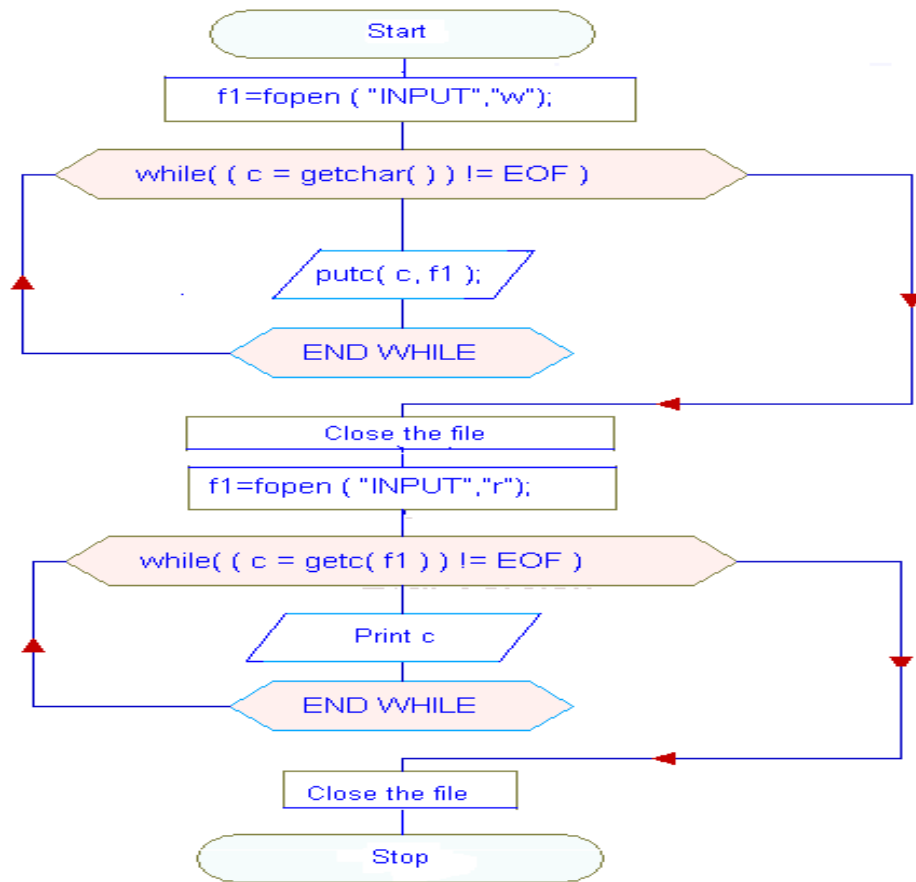
Step 3: Write characters to the file

Step 4: Close the file

Step 5: Read the characters from the file.

Step 6: Print the characters.

Step 7: Close the file



Program:

```

#include<stdio.h>
void main()
{
    FILE *fp;
    char c;
    printf("Data input\n");
    f1=fopen("INPUT","w");
    while((c=getchar())!=EOF)
        putc(c,f1);
    fclose(f1);
    /* reopen the file*/
    printf("Data output\n");
    f1=fopen("INPUT","r");

```

```
        while((c=getc(f1))!=EOF)
            printf("%c",c);
        fclose(f1);
        getch( );
    }
```

Input :

Data input

This is the Program to test the file handling

^Z

Output :

Data output

This is the Program to test the file handling

Assignment

1. Write a Program to copy the contents of one file to another file.
2. Differentiate between the following :
 1. getchar and getc
 2. putchar and putc

Viva-voce questions

1. What do you mean by file?
2. What are the different types of file in C?
3. Differentiate between binary and text file in C?

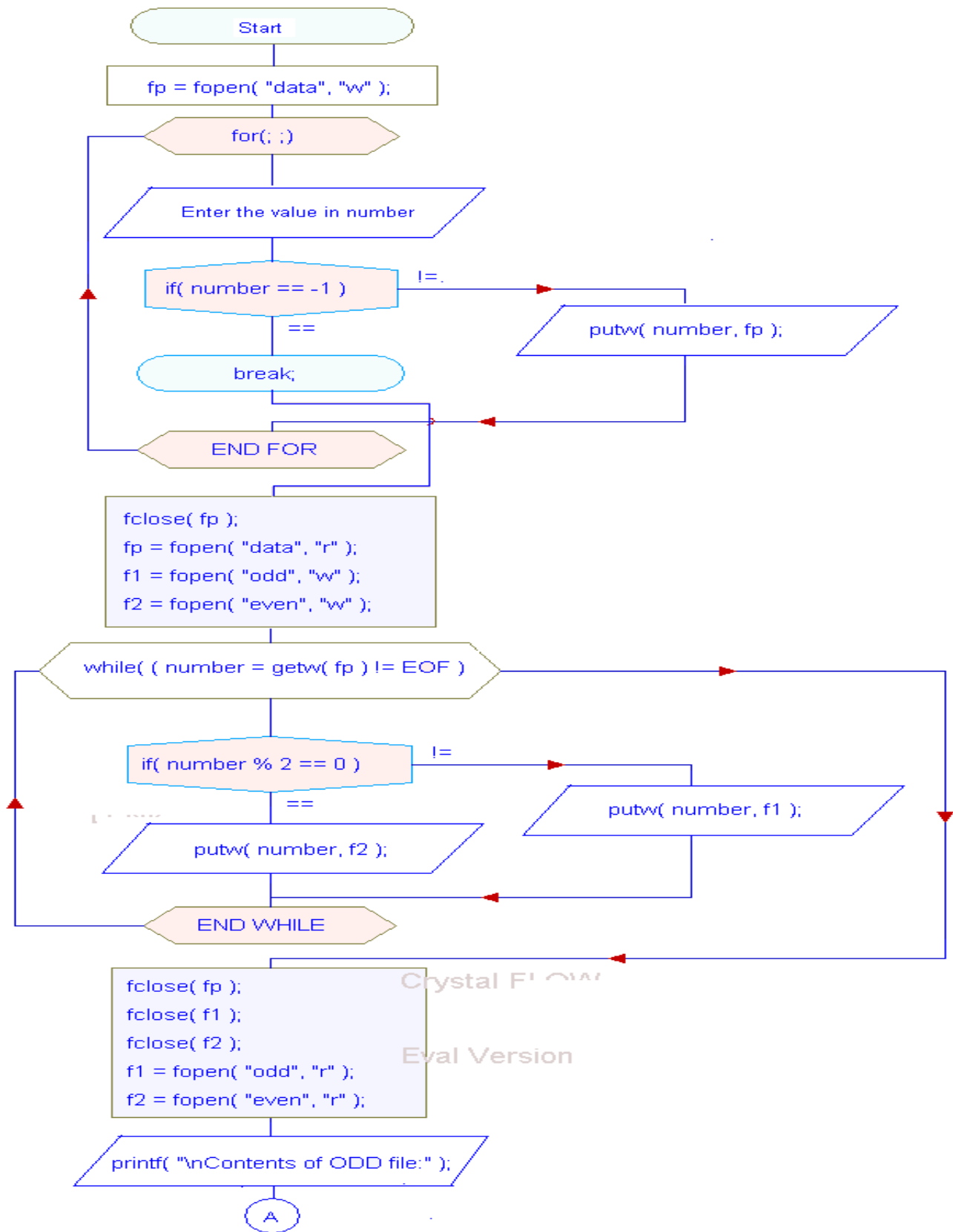
Experiment No. 30

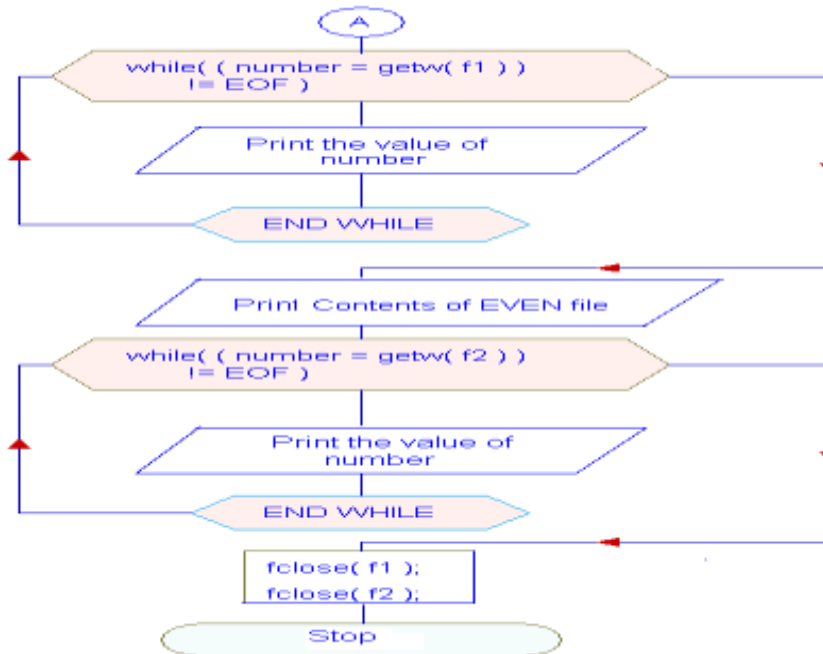
Aim : A file named data that contains a series of numbers. Write a Program to read these numbers and then write all odd numbers to a file named odd and all even numbers to a file to be called even.

Description : This Program uses three files simultaneously and therefore we need to define three-file pointers **fp,f1,f2**. First the file **data** containing integer values is created. The integer values are read from keyboard and are written to the file **data** with the help of **putw** function. Input from the keyboard is terminated when we type **-1** and the file is closed. The next step is to open all the three files, **data** for reading, **odd** and **even** for writing. The contents of **data** file are read, and written to the files **odd** and **even** after appropriate testing.

Algorithm:

- Step 1: Open the file data in write mode.
- Step 2: Input numbers from the keyboard in the variable number and write it to the file data till the number -1 is entered.
- Step 3: Close the file data
- Step 4: Open the file data in read mode.
- Step 5: Open two more files odd and even in write mode.
- Step 6: Read numbers from the file data till the end of file is reached.
- Step 7: If the number is odd write it to the file named odd otherwise write it to the file named even.
- Step 8: Print the contents of odd file till the end of file is reached.
- Step 9: Print the contents of even file till the end of file is reached.





Program :

```

#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *fp,*f1,*f2;
    int number,i;
    clrscr();
    fp=fopen("data","w");
    printf("Enter numbers :");
    for(;;)
    {
        scanf("%d",&number);
        if(number== -1)
            break;
        putw(number,fp);
    }
    fclose(fp);
    fp=fopen("data","r");
  
```

```

f1=fopen("odd","w");
f2=fopen("even","w");
while((number=getw(fp))!=EOF)
{
    if(number%2==0)
        putw(number,f2);
    else
        putw(number,f1);
}
fclose(fp);
fclose(f1);
fclose(f2);
f1=fopen("odd","r");
f2=fopen("even","r");
printf("\nContents of ODD file:");
while((number=getw(f1))!=EOF)
    printf("\n%d",number);
printf("\nContents of EVEN file:");
while((number=getw(f2))!=EOF)
    printf("\n%d",number);
fclose(f1);
fclose(f2);
getch();
}

```


Input :

Enter numbers :

1

2

3

4

5

6

7

8

9

10

-1

Output :

Contents of ODD file:

1

3

5

7

9

Contents of EVEN file:

2

4

6

8

10

Assignment

1. Write a Program to input 10 numbers and write it in afile. Calculate the sum and write the result in same file at the EOF.

Viva-voce questions

2. Differentiate between the following :
 - a) getc and getw
 - b) putc and putw
3. The FILE structure is defined in which of the following files :
 - a) stdlib.h
 - b) stdio.h
 - c) io.h
 - d) stdio.c

Experiment No. 31

Aim : Write a Program to process sequential file for payroll data.

Description : This Program takes Employee's details such as employee id, employee name, basic pay as input, calculates dearness allowance, house rent allowance, income tax & net salary and stores this details in file named payroll. Then file is again opened in read mode and the displays the entire contents of file.

In this Program which variable represents which information is given below :

eid – Employee id

ename – Employee name

basic – Basic pay

da – Dearness allowance

hra – House rent allowance

tax – Income tax

gross – Gross salary

net – Net salary

The values of **da**, **hra**, **tax**, **gross** and **net** are calculated as follows :

da=25% of **basic**

hra=10% of **basic**

gross=**basic**+**da**+**hra**

tax=30% of **gross**

net=**gross**-**tax**

Algorithm:

Step 1: Input Empid, name and basic pay until EOF character is entered.

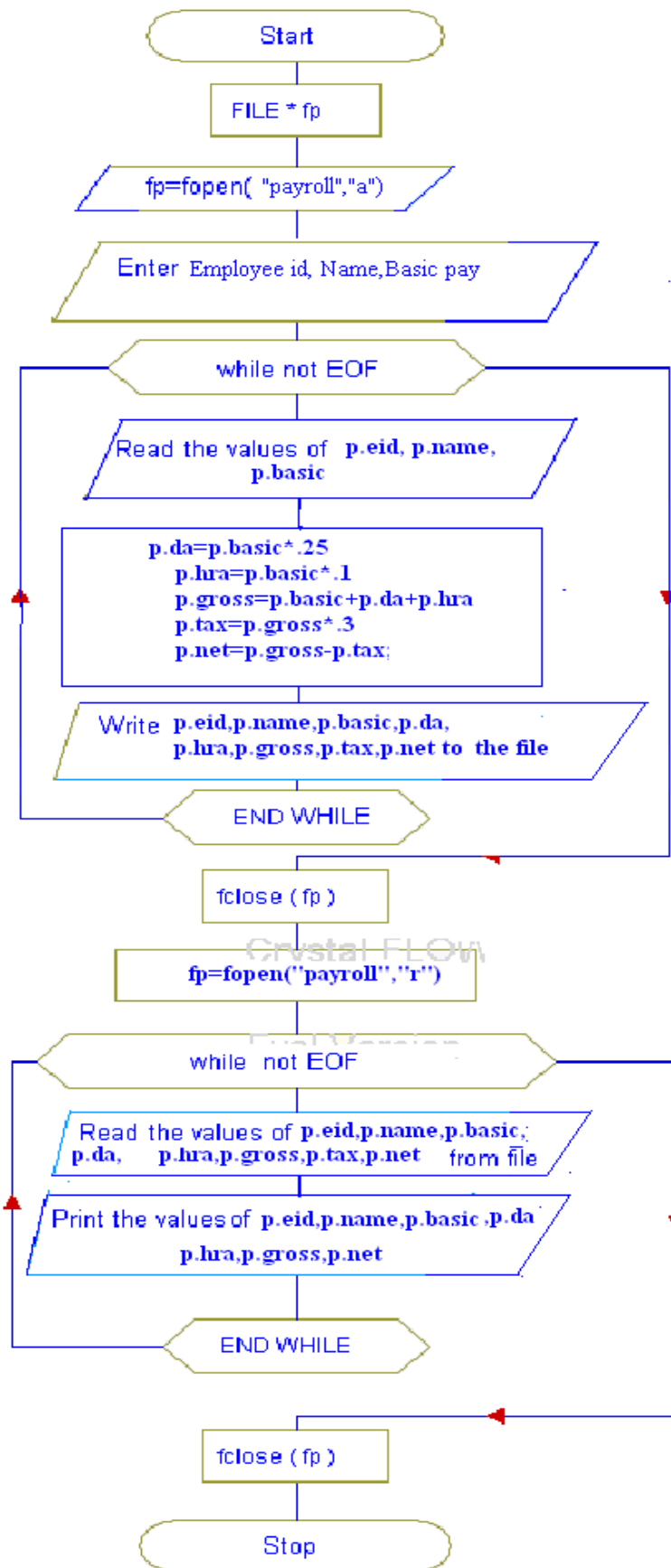
Step 2: Calculate HRA, DA, Income Tax and Net salary.

Step 3: Open a new file in write mode and write Empid, name , Basic pay, HRA, DA, Income Tax, Net salary to the file.

Step 4: Close file.

Step 5: Open file in read mode.

Step 6: Read the contents of the file and print on the screen.



Program :

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct payroll
    {
        int eid;
        char ename[15];
        int basic;
        float da,hra,tax;
        float gross,net;
    }p;
    char c;
    FILE *fp;
    clrscr();
    fp=fopen("payroll","a");
    printf("Enter Employee Id, Name, Basic Pay:\n");
    while(scanf("%d%s%d",&p.eid,p.ename,&p.basic)!=EOF)
    {
        p.da=p.basic*.25;
        p.hra=p.basic*.1;
        p.gross=p.basic+p.da+p.hra;
        p.tax=p.gross*.3;
        p.net=p.gross-p.tax;
        fwrite(&p,sizeof(p),1,fp);
    }
    fclose(fp);
    fp=fopen("payroll","r");
    printf("Cotents of file:\n");
    printf("_____ \n");
    while(fread(&p,sizeof(p),1,fp))
    {
        printf("\nEmployee Id :%d",p.eid);
```

```

        printf("\nEmployee Name :%s",p.ename);
        printf("\nBasic Pay :%d",p.basic);
        printf("\nDearness Allowance :%f",p.da);
        printf("\nHouse Rent Allowance :%f",p.hra);
        printf("\nIncome Tax :%f",p.tax);
        printf("\nNet Salary :%f",p.net);
        printf("\n-----\n");
    }
    fclose(fp);
    getch();
}

```

Input :

Enter Employee Id, Name, Basic Pay:

101

Sam

8000

102

Adam

7000

103

Jhon

6000

^Z

Output :

Cotents of file:

Employee Id :101

Employee Name :Sam

Basic Pay :8000

Dearness Allowance :2000.000000

House Rent Allowance :800.000000

Income Tax :3240.000000

Net Salary :7560.000000

Employee Id :102

Employee Name :Adam

Basic Pay :7000

Dearness Allowance :1750.000000

House Rent Allowance :700.000000

Income Tax :2835.000000

Net Salary :6615.000000

Employee Id :103

Employee Name :Jhon

Basic Pay :6000

Dearness Allowance :1500.000000

House Rent Allowance :600.000000

Income Tax :2430.000000

Net Salary :5670.000000

Assignment

1. Write a Program that will generate a data file containing the list of customers and their corresponding phone numbers. Use a structure variable to store the number and telephone of each customer. Create a data file using a sample list.

Viva-voce questions

1. If a file contains the line “I am a boy\r\0” then on reading this line into array str[] using fgets what would str[] contain:
 - a) I am a boy\r\n\0
 - b) I am a boy\r\0
 - c) I am a boy\n\0
 - d) I am a boy
2. On opening a file for reading which of the following activities are performed :
 - a) The disk is searched for existence of the file.
 - b) The file is brought into the memory
 - c) A pointer is set up which points to the first character in the file.
 - d) All the above
3. While using the statement :
fp = fopen(“myfile”,”r”);
what happens if,
 - a) “myfile .c” does not exist on the disk
 - b) “myfile.c” exist on the disk
4. Distinguish the following :
 - a) getc and getchar functions.
 - b) printf and fprintf functions.
 - c) feof and ferror functions.
5. Point out the errors, if any, in the following :

```
main()  
{  
    char fname[]=”c:\\students.dat”;  
    FILE *fp;  
    fp=fopen(fname,”tr”);  
    if(fp == NULL)  
        printf(“\n unable to open file.....”);  
}
```


Experiment No. 32

Aim : Write a Program to display sum of list of values using command line arguments.

Description : This Program takes a list of values using command line argument and the output of this Program is the sum of these values. List of values will be stored in array **argv[]**. Since **argv** is an array of character pointers so before addition we convert the elements of array **argv** in integer using **atoi** function defined in **stdlib.h** header file.

Algorithm :

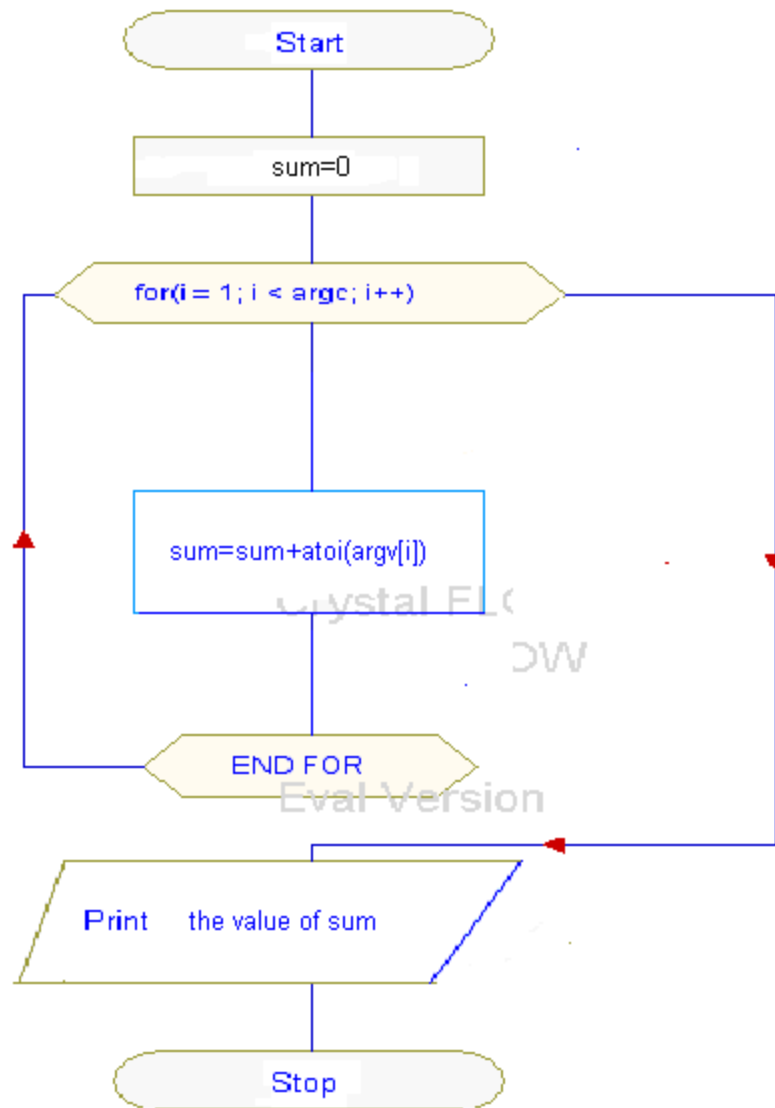
Step 1 : Set sum = 0.

Step 2 : Repeat Step3 and Step 4 while i < argc.

Step 3 : Read values from command line argument and store in argv[i].

Step 4 : Calculate sum=sum + atoi(argv[i]).

Step 5 : Print the value of sum.



Program :

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main(int argc,char* argv[])
{
    int sum=0,i;
    printf("PROGRAM FOR ADDING A LIST OF VALUES USING
    COMMAND LINE ARGUMENTS\n");
    for(i=1;i<argc;i++)
        sum=sum+atoi(argv[i]);
    printf("Sum : %d",sum);
    getch();
}
```

Output :

Compile the Program then make executable file of Program by pressing f9 key

Then go to command prompt .

```
C:\TURBOC>arg 3 4 5 6 7
```

```
PROGRAM FOR ADDING A LIST OF VALUES USING COMMAND LINE ARGUMENTS
```

```
Sum : 25
```

Assignment

1. Write a Program to sort a list of names using command line parameters.
2. What would be the output of the following?

```
main(int argc, char **argv)
{
    argc = argc - (argc - 1);
    printf("%s", argv[argc - 1]);
}
```

3. If the following Program (myprg) is run from the command line as **myprg 1 2 3**

what would be the output?

```
main(int argc, char * argv[ ])
{
    int i;
    i= argv[1] + argv[2] + argv[3];
    printf("%d", i);
}
```

- a) 123
- b) 6
- c) Error
- d) "123"

4. If the following Program (myprg) is run from the command line as **myprg 1 2 3**

what would be the output?

```
main(int argc, char * argv[ ])
{
    int i;
    for(i = 0; i < argc; i++)
        printf("%s", argv[i]);
}
```

Viva-voce questions

4. What do you mean by command line argument?
5. What do the 'c' and 'v' in argc and argv stand for?
6. According to ANSI specification which is the correct way of declaring main(), when it receives command line argument?
 1. `main(int argc, char *argv[])`
 2. `main(argc, argv)`
`int argc; char *argv[];`
 3. `main()`
`{`
`int argc; char *argv[];`
`}`
 4. None of the above

4. The maximum length of the command line arguments including the spaces between adjacent argument is
 - a. 128 characters
 - b. 256 characters
 - c. 67 characters
 - d. It may vary from one operating system to another

5. What does argc and argv mean by in
`main(int argc, char *argv[])`

6. If the following Program (myprg) is run from the command line as
myprg 1 2 3

what would be the output?

```
main(int argc, char * argv[ ])  
{  
    printf("%s", argv[0]);  
}
```

Experiment No. 33

Aim : Write a menu based Program to add, delete, list and modify the contents of a file. The file should take employee name, age and salary.

Description : It is a menu driven Program which ask the user either to insert, delete, list and modify the records from a file.

Program :

```
#include<stdio.h>
main()
{
    FILE *fp,*ft;
    char another,choice;
    struct emp
    {
        char name[40];
        int age;
        float bs;
    };

    struct emp e;
    char empname[40];
    long int recsize;

    fp=fopen("EMP.DAT","rb+");
    if ( fp== NULL)
    {
        fp= fopen("EMP.DAT ","wb+");
        if (fp==NULL)
        {
            puts("Cannot open file");
            exit();
        }
    }
}
```

```

        }
    }

    recsize= sizeof(e);
    while(1)
    {
        gotoxy(30,10);
        printf("1. Add Records");
        gotoxy(30,12);
        printf("2. List Records");
        gotoxy(30,14);
        printf("3. Modify Records");
        gotoxy(30,16);
        printf("4. Delete Records");
        gotoxy(30,18);
        printf(" 0. Exit");
        gotoxy(30,20);
        printf("Your choice");
        fflush(stdin);
        choice=getche();
        switch(choice)
        {
            case '1':
                clrscr();
                fseek(fp,0,SEEK_END);
                another='Y';
                while( another=='Y')
                {
                    printf("\n Enter name, age and basic sal");
                    scanf("%s%d%f",e.name,&e.age,&e.bs);
                    fwrite(&e,recsize,1,fp);
                    printf("\n Add another record(Y/N)");
                    fflush(stdin);
                    another= getche();
                }
            }
        }
    }

```

```

    }

    break;

case'2':

    clrscr();
    rewind(fp);
    while( fread(&e,recsize,1,fp)==1)
        printf("\n%s%d%f",e.name,e.age,e.bs);
        break;

case'3':

    clrscr();
    another='Y';
    while(another=='Y')
    {
        printf("\n Enter name of employee to modify");
        scanf("%s",empname);
        rewind(fp);
        while(fread(&e,recsize,1,fp)==1)
        {
            if (strcmp(e.name,empname)==0)
            {
                printf("\n Enter new name , age & bs");
                scanf("%s%d%f",e.name,&e.age,&e.bs);
                fseek(fp,recsize,SEEK_CUR);
                fwrite(&e,recsize,1,fp);
                break;
            }
        }
        printf("\nModify another record(Y/N)");
        fflush(stdin);
        another=getche();
    }

    break;

```


case'4':

```
clrscr();
another='Y';
while( another=='Y')
{
    printf("\n Enter name of employeeto delete");

    scanf("%s", empname);
    ft=fopen("TEMP.DAT","wb");
    rewind(fp);
    while(fread(&e,recsize,1,fp)==1)
    {
        if (strcmp(e.name,empname)!=0)
            fwrite(&e,recsize,1,ft);
    }

    fclose(fp);
    fclose(ft);
    remove("TEMP.DAT");
    rename("TEMP.DAT","EMP.DAT");
    fp=fopen("EMP.DAT","rb+");
    printf("Delete another record(Y/N)");
    fflush(stdin);
    another=getche();
}
```

break;

case'0':

```
fclose(fp);
exit();
```

```
}
```

```
}
```

```
}
```

Input

1. Add Records
 2. List Records
 3. Modify Records
 4. Delete Records
 0. Exit
- Your choice 1

Enter name, age and basic sal
SAM 23 8000

Output

Add another record(Y/N)n

1. Add Records
 2. List Records
 3. Modify Records
 4. Delete Records
 0. Exit
- Your choice

Assignment

5. Write a Program that prepares the Roll list of students. The names of students are supplied from data file and they are written on a new file by allotting the roll numbers sequentially to the students.

Viva-Voce Questions

6. Explain the functioning of fseek and ftell.
7. Explain rewind function.

Experiment No. 34

Aim: Write a Program to practice one of the Numerical Integration Method.

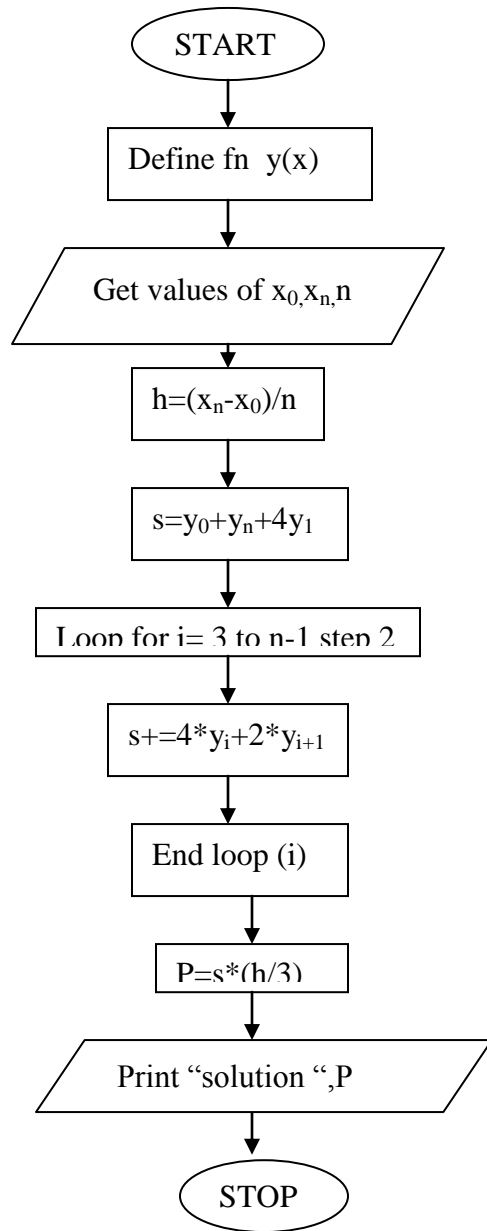
Description :

Simpson's rule is a Newton-Cotes formula for approximating the integral of a function f using quadratic polynomials (i.e., parabolic arcs instead of the straight line segments used in the trapezoidal rule). Simpson's rule can be derived by integrating a third-order Lagrange interpolating polynomial fit to the function at three equally spaced points. In particular, let the function f be tabulated at points x_0 , x_1 , and x_2 equally spaced by distance h , and denote $f_n = f(x_n)$. Then Simpson's rule states that

$\int_{x_0}^{x_2} f(x) dx = \int_{x_0}^{x_0+2h} f(x) dx$

Algorithm :

- Step01. Start of Program
- Step02. Input Lower limit a
- Step03. Input Upper limit b
- Step04. Input no. of subintervals n
- Step05. $h=(b-a)/n$
- Step06. sum=0
- Step07. sum=fun(a)+4*fun(a+h)+fun(b)
- Step08. for i=3;i<n;i+=2
- Step09. sum+=2*fun(a+(i-1)*h)+4*fun(a+i*h)
- Step10. End of loop i
- Step11. Result=sum*h/3
- Step12. Print output result
- Step13. End of Program
- Step14. Start of section fun
- Step15. Temp=1/(1+(x*x))
- Step16. Return temp
- Step17. End of section fun



Program :

```
//...HEADER FILE DECLARATION
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<process.h>
#include<string.h>
//....Function Prototype Declaration
float fun(float);
//....Main Execution Thread
void main()
{
//....Variable Declaration Field
//....Floating type
float result=1;
float a, b;
float h, sum;
//....Integer type
int i, j;
int n;
//....Invoke clear screen function
clrscr();
//....Input section
//....Input range
printf("\n\n Enter the range -");
printf("\n\n Lower limit a- ");
scanf("%f", &a);

printf("\n\n Upper limit b - ");
scanf("%f", &b);
//....Input no. of Subintervals
printf("\n\n enter number of subintervals - ");
scanf("%d", &n);
//....Calculation and processing section
h=(b-a)/n;
sum=0;
sum=fun(a)+4*fun(a+h)fun(b);
for(i=3;i<n;i+=2)
{
sum+=2*fun(a+(i-1)*h)+4*fun(a+i*h);
}
result =sum*h/3;
//...Output Section
printf("\n\n\n Value of the integral is %6.4f\t",result);
// ... Invoke User Watch Halt Function
printf("\n\n\nPress Enter to Exit");
getch();
}
//... Termination of Main Execution Thread
R.C.E.T. / Department Of Computer Science And Engineering / PSLB Using C Lab
```

```
//... Function Body
float fun(float x)
{
float temp;
temp=1/(1+x*x);
return temp;
}
//... Termination of Function Body
```

Output

Enter the range -
Lower Limit a - 0
Upper Limit b - 6
Enter no of subintervals – 6
Value of the internal is 1.3662
Press Enter to Exit

Experiment No. 35

Aim: Write a Program to find the solution of Differential Equation by Runge kutta method.

Description :

One member of the family of Runge–Kutta methods is so commonly used that it is often referred to as "RK4" or simply as "*the* Runge–Kutta method".

Let an initial value problem be specified as follows.

Then, the RK4 method for this problem is given by the following equations:

where y_{n+1} is the RK4 approximation of $y(t_{n+1})$, and

Thus, the next value (y_{n+1}) is determined by the present value (y_n) plus the product of the size of the interval (h) and an estimated slope. The slope is a weighted average of slopes:

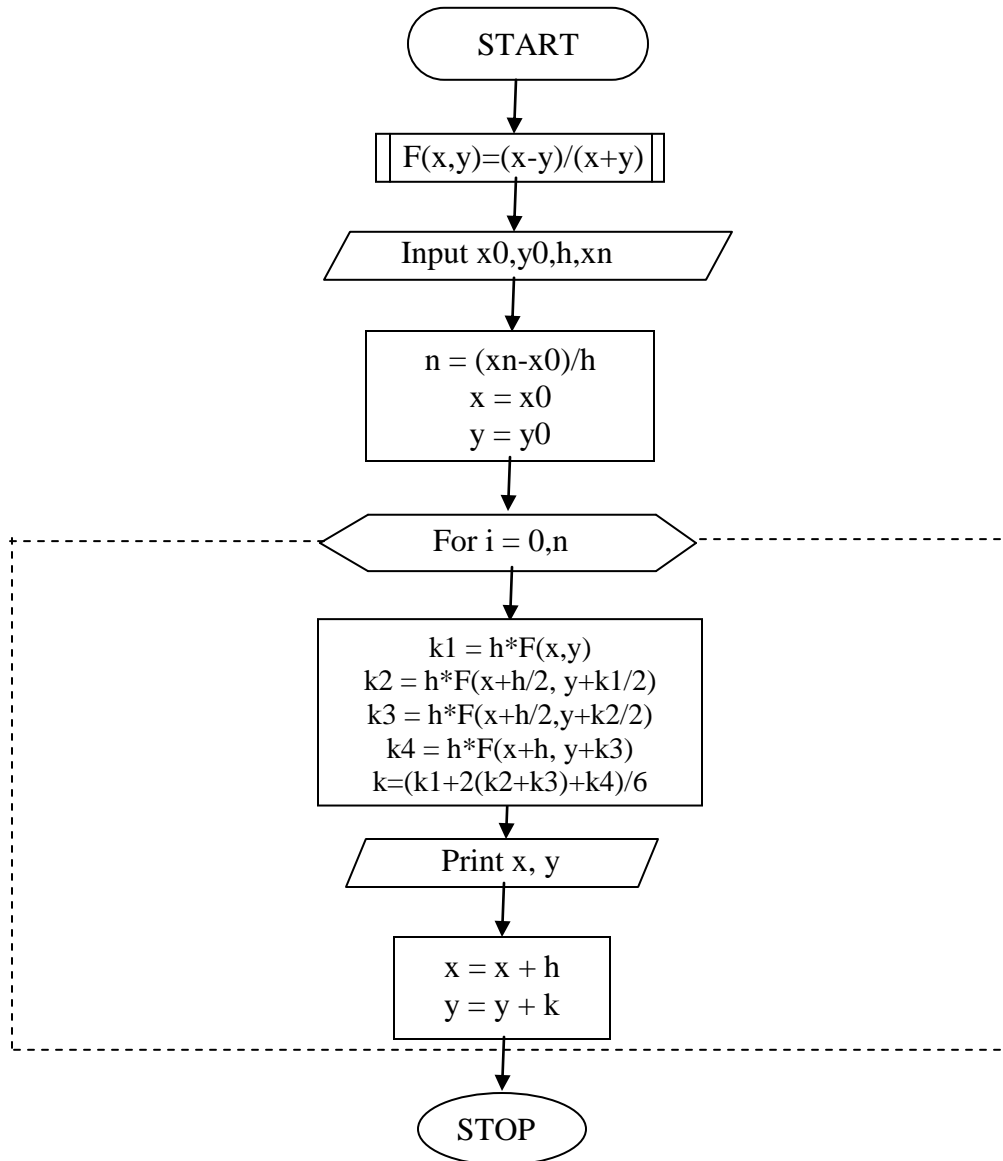
- k_1 is the slope at the beginning of the interval;
- k_2 is the slope at the midpoint of the interval, using slope k_1 to determine the value of y at the point $t_n + h/2$ using Euler's method;
- k_3 is again the slope at the midpoint, but now using the slope k_2 to determine the y -value;
- k_4 is the slope at the end of the interval, with its y -value determined using k_3 .

In averaging the four slopes, greater weight is given to the slopes at the midpoint:

The RK4 method is a fourth-order method, meaning that the error per step is on the order of h^5 , while the total accumulated error has order h^4 .

Algorithm :

1. Function $F(x)=(x-y)/(x+y)$
2. Input x_0,y_0,h,x_n
3. $n=(x_n-x_0)/h$
4. $x=x_0$
5. $y=y_0$
6. For $i=0,n$
7. $k_1 = h * F(x,y)$
8. $k_2 = h * F(x+h/2,y+k_1/2)$
9. $k_3 = h * F(x+h/2,y+k_2/2)$
10. $k_4 = h * F(x+h,y+k_3)$
11. $k = (k_1+(k_2+k_3)*2+k_4)/6$
12. Print x, y
13. $x = x+h$
14. $y = y+k$
15. Next i
16. Stop



Program:

```

/**Program of runge kutta method***/
#include<stdio.h>
#define F(x,y) (x-y)/(x+y)
main()
{
int i,n;
float x0,y0,h,xn,k1,k2,k3,k4,x,y,k;
printf("\nEnter THE VALUES:x0,y0,h,xn:\n");
scanf("%f%f%f%f",&x0,&y0,&h,&xn);
n=(xn-x0)/h;
x=x0;
y=y0;

```



```

for(i=0;i<=n;i++)
{
    K1=h*F(x,y);
    K2=h*F(x+h/2,y+k1/2);
    K3=h*F(x+h/2,y+k2/2);
    K4=h*F(x+h,y+k3);
    K=(k1+2k2+2k3+k4)/6;
    Printf("\nx=%f y=%f",x,y);
    X=x+h;
    Y=y+k;
}
Return;
}

```

Output :

ENTER THE VALUES:x0,y0,h,xn:
0 1 0.02 0.1

```

X=0.000000    y=1.000000
X=0.020000    y=0.980000
X=0.040000    y=0.960816
X=0.060000    y=0.942446
X=0.080000    y=0.924885
X=0.100000    y=0.908128

```

Notation used in Program

- (i) x0 is the initial values of x.
- (ii)y0 is the initial values of y.
- (iii) h is the spacing values of x.
- (iv) xn is the last value of x at which value of y is required.

Experiment No. 36

Aim: Write a Program to find the solution of Differential Equation by Euler's method.

Description :

The Euler forward scheme may be very easy to implement but it can't give accurate solutions. A very small step size is required for any meaningful result. In this scheme, since, the starting point of each sub-interval is used to find the slope of the solution curve, the solution would be correct only if the function is linear. So an improvement over this is to take the arithmetic average of the slopes at x_i and x_{i+1} (that is, at the end points of each sub-interval). The scheme so obtained is called modified Euler's method. It works first by approximating a value to y_{i+1} and then improving it by making use of average slope.

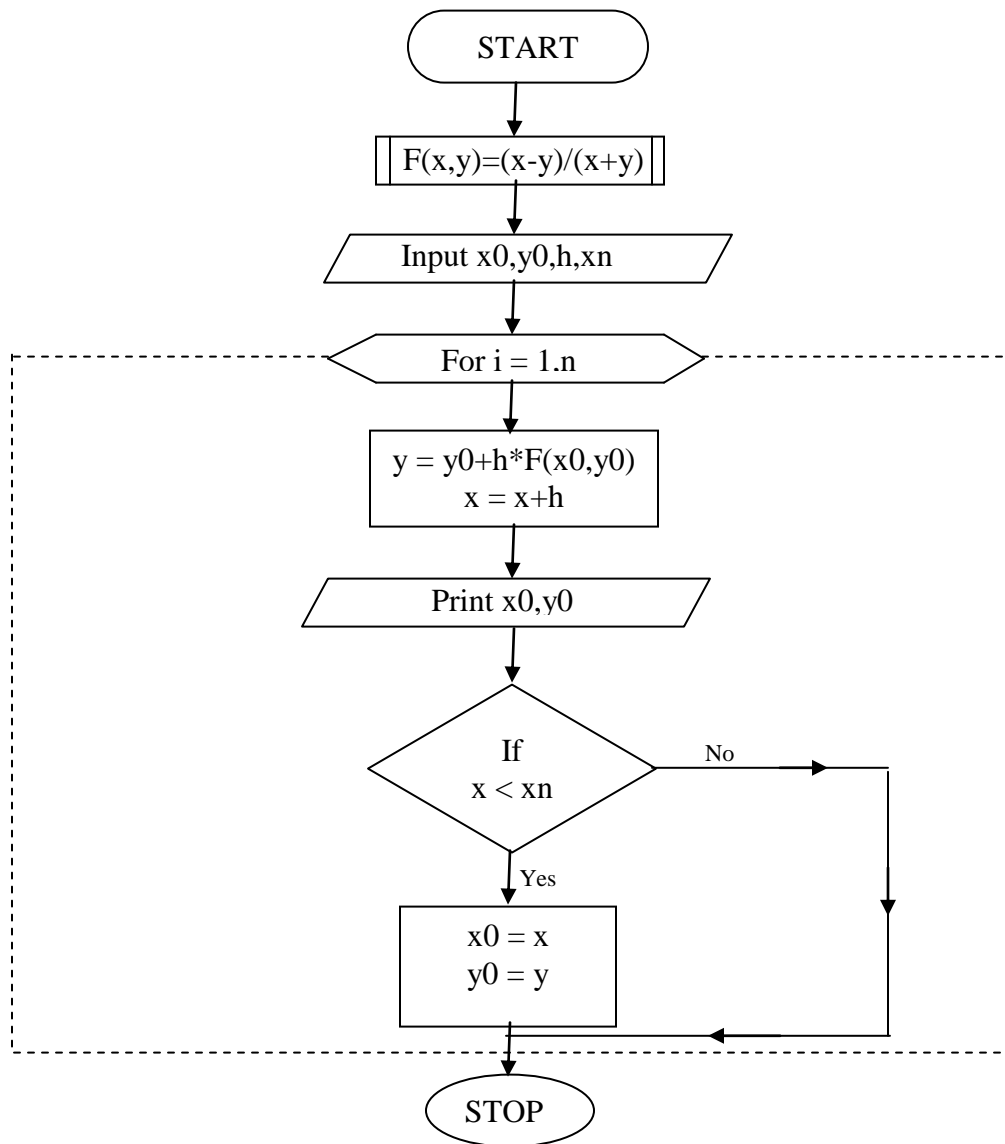
$$\begin{aligned}y_{i+1} &= y_i + h/2 (y'_i + y'_{i+1}) \\ &= y_i + h/2(f(x_i, y_i) + f(x_{i+1}, y_{i+1}))\end{aligned}$$

If Euler's method is used to find the first approximation of y_{i+1} then

$$y_{i+1} = y_i + 0.5h(f_i + f(x_{i+1}, y_i + hf_i))$$

Algorithm :

1. function $f(x,y)=(x-y)/(x+y)$
2. input x_0,y_0,h,x_n
3. $n=((x_n-x_0)/h)+1$
4. for $i=1,n$
5. $y=y_0+h*f(x_0,y_0)$
6. $x=x+h$
7. print x_0,y_0
8. if $x = x_n$ then
 $x_0 = x$
 $y_0 = y$
 else
9. Next i
10. Stop



Program :

```
# include<stdio.h>
# define F(x,y) (x-y) / (x+y)
main ()
{
    int i,n;
    float x0,y0,h,xn,x,y;
    printf("\n Enter the values: x0,y0,h,xn: \n");
    scanf("%f%f%f%f",&x0,&y0,&h,&xn);
    n=(xn-x0)/h+1;
    for(i=1;i<=n;i++)
    {
        y=y0+h*F(x0,y0);
        x=x0+h;
        printf("\n X=%f Y=%f",x0,y0);
        if(x<xn)
        {
            x0=x;
            y0=y;
        }
    }
    return;
}
```

Output :

```
Enter the values: x0,y0,h,xn:
0 1 0.02 0.1
X=0.000000 Y=1.000000
X=0.020000 Y=0.980000
X=0.040000 Y=0.960800
X=0.060000 Y=0.942399
X=0.080000 Y=0.924793
X=0.100000 Y=0.907978
```

Notations used in the Program

- (i) x_0 is the initial value of x.
- (ii) y_0 is the initial value of y.
- (iii) h is spacing value of x.
- (iv) x_n is the last value of x at which value of y is required.

Experiment No. 37

Aim: Write a Program to find the roots of an equation using Newton Raphson Method.

Description:

The Newton-Raphson Method is a powerful method of finding the roots of the function $f(x)$, that is the values of x for which $f(x) = 0$. Suppose that x_{old} is an estimate of the root. Then a better estimate is x_{new} which is defined by:

$$x_{new} = x_{old} - f(x_{old})/g(x_{old})$$

where $g(x) = df(x)/dx$.

This procedure is iterated until convergence is achieved, defined by $|x_{new} - x_{old}| < acc$, where acc is user defined accuracy.

Algorithm :

Step 01. Start of the Program

Step 02. Input the variable x0, n for the task

Step 03. Input Epsilon & delta

Step 04. for i = 1 and repeat if i<=n

Step 05. f0 = f(x0)

Step 06. df0 = df(x1)

Step 07. if | df0| <= delta

- a. Print Slope too small
- b. Print x0 ,f0, df0, i
- c. End of Program

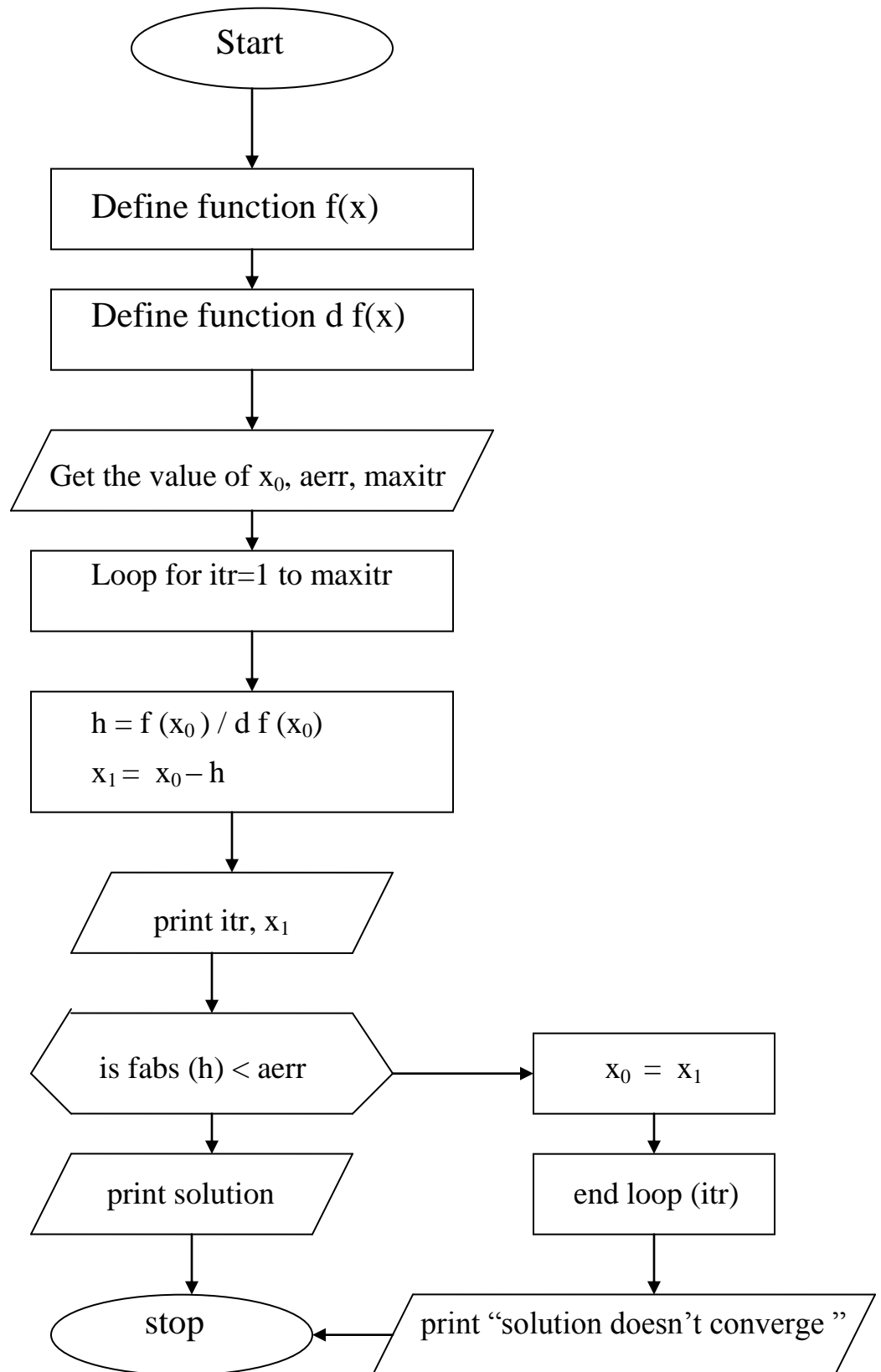
Step 08. x1 = x0-(f0/df0)

Step 09. if |(x1-x0)/x1|< epsilon

- a. Print convergent
- b. Print x1, f(x1), i
- c. End of Program

Step 10. x0 = x1

Step 11. End Loop



Program :

```
//.....including source header files

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

//.....defining formulae

# define f(x)      3*x -cos(x)-1
# define df(x)     3+sin(x)

//.....Function Declaration prototype

void NEW_RAP();

//.....Main Execution Thread
void main()
{
    clrscr();
    printf (“\n Solution by NEWTON RAPHSON method \n”);
    printf (“\n Equation is : “);
    printf (“\n\t\t\t 3*x - cosx - 1=0 \n\n “);
    NEW_RAP();
    getch();
}

//.....Function Declaration

void NEW_RAP()
{
    // Internal Declaration Field
    long float x1,x0;
    long float f0,f1;
    long float df0;

    int i=1;
    int itr;
    float EPS;
    float error;

    // Finding an Approximate ROOT of Given Equation, Having +ve value
    For(x1=0; ; x1 += 0.01)
    {
        F1 = f(x1);
        If(f1>0)
        {
```

```

        break;
    }
}
// Finding an Approximate ROOT of Given Equation, Having -ve value

x0 = x1-0.01;
f0 = f(x0);
printf("Enter the number of iterations : ");
scanf("%d",&itr);

printf("Enter the maximum possible error : ");
scanf("%d",&EPS);

if(fabs(f0)>f1)
    {
        Printf("\n\t The root is near to %.4f\n",x1);
    }

if(f1>fabs(f(x0)))
    {
        Printf("\n\t The root is near to %.4f\n",x0);
    }
X0 = (x0+x1)/2;

For(;i<=itr;i++)

}

{
F0=f(x0);
Df0=df(x0);
X1=x0 - (f0/df0);
Printf("\n\t The %d Approximation to the root is : %f,I,x1");
Error=fabs(x1-x0);
If(error<EPS)
{
    Break;
}
X0=x1;
If(error>EPS)
{
    Printf("\n\n\t NOTE:-");
Printf("The Number Of Iteration are not sufficient ");
}
Printf("\n\n\n\t\t\t-----");
Printf("\n\t\t\t\t\t\t\t\t The Root Is %.4f ",x1);;
Printf("\n\t\t\t\t\t\t\t\t-----");
}

```


Output

Solution by NEWTON RAPHSON method

Equation s is:

$$3*X - \cos X - 1 = 0$$

Enter the number off iterations : 10

Enter The Maximum possible error : .0000001

The Root is near to 0.6100

The 1 approximation ot the root is :0.607102

The 2 approximation ot the root is :0.607102

The 3 approximation ot the root is : 0.607102

The root is 0.607102

Assignment

1. Write a Program to implement simpson's 3/8 rule.

Viva-Voce Questions

1. Explain Newton Raphson Method.
2. Explain Euler's equation.
3. What is Runge Kutta Method.

Experiment No. 38

Aim -Write a programme to calculate the area of a circle, triangle, square, rectangle the various sides are given.

Description: This Program takes required input for calculating the area of a circle, triangle, square, and rectangle.

Algorithm

Step 1 : input the length and width of a rectangle, triangle or square (ex a,b)

Step 2 : input the radius of a circle (ex. r)

Step 3 : calculate the area of triangle using formula $\text{area} = 0.5 * a * b$

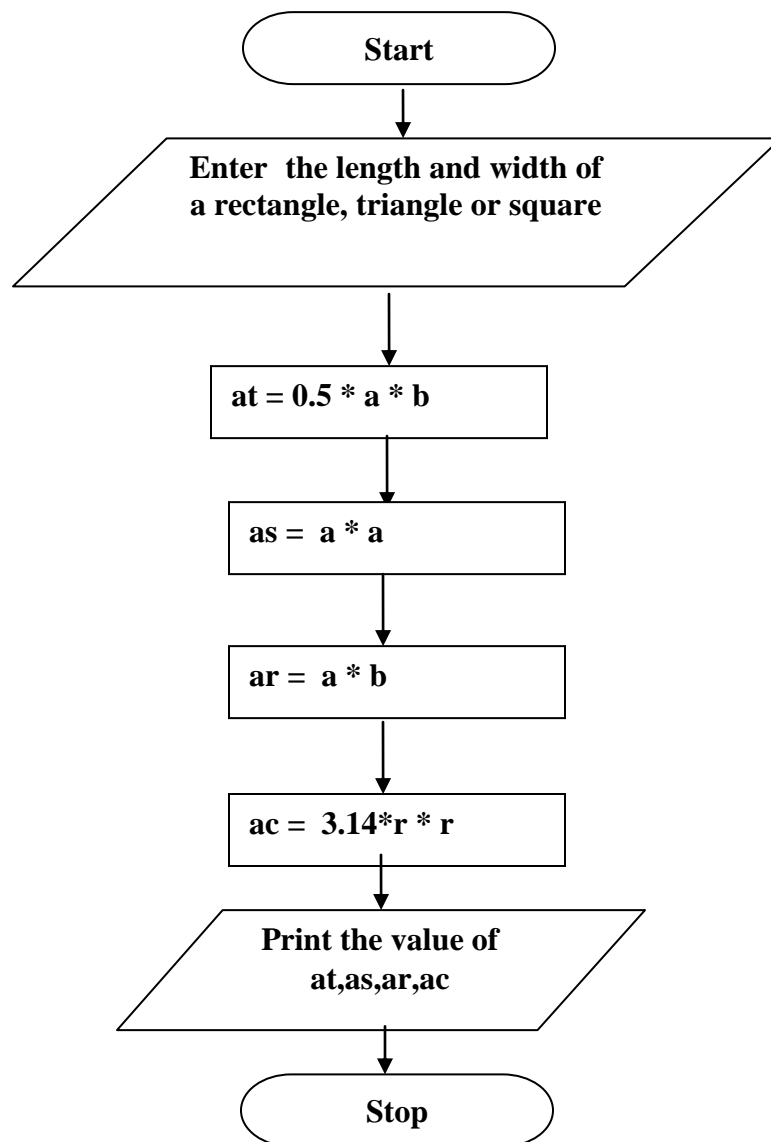
Step 4 : calculate the area of square using formula $\text{area} = a * a$

Step 5 : calculate the area of rectangle using formula $\text{area} = a * b$

Step 6 : calculate the area of circle using formula $\text{area} = 0.5 * a * b$

Step 7 : print all the results.

Step 8 : exit.



Program-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float a,b,r;
    float ac,ar,as,at;
    printf("\n enter sides");
    scanf("%f%f", &a ,&b);
    printf ("\n enter radius");
    scanf("%f",&r);
    at =.5*a*b;
    as = a*a;
    ar = a*b;
    ac =3.14*r*r;
    printf("\n area of triangle =%f", at);
    printf("\n area of rectangle =%f", ar);
    printf("\n area of square =%f", as);
    printf("\n area of circle =%f", ac);
    getch();
}
```

Input :

Enter sides-4.5 , 5

Enter radius- 3.2

Output :

Area of triangle=11.250000

Area of rectangle=22.500000

Area of square=20.25000

Area of circle=32.153603

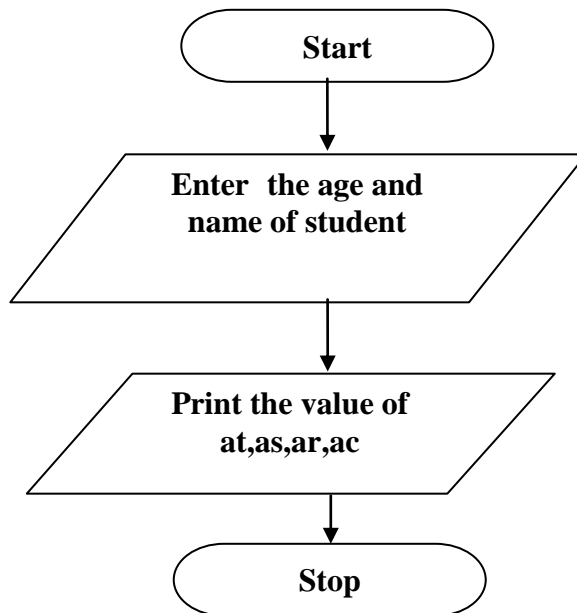
Experiment No. 39

Aim- Write a Program to read name & age of student & display it on a screen

Description: This Program takes name and age as input print the name and age..

Algorithm

- Step 1 : input the age of student (ex age)
- Step 2 : input the name of student(ex. name)
- Step 3 : print the name and age of student.
- Step 4 : exit.



Program-

```
#include<stdio.h>
#include<conio.h>
void main()
{
int age,i;
char name[20];
clrscr();
printf("\n enter age");
scanf("%d", &age);
printf("enter name of the student");
scanf("%s", name);
printf("\n name of the student is %s and age is %d",name,age);
getch();

}
```

Input :

Enter age - 21

Enter Name of the student - shubham

Output :

Name of the student is shubham and age is 21

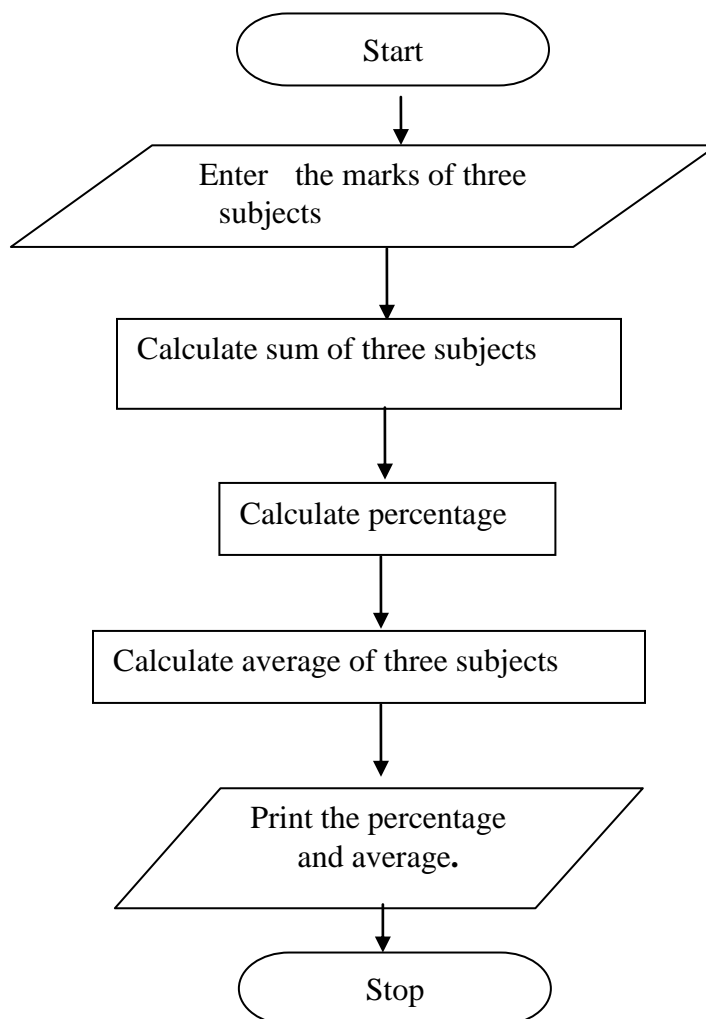
Experiment No. 40

Aim- Write a Program to find the average of three marks of students & calculate the percentage.

Description- This program takes the marks of the student as input and stored the marks in an array and calculate the average of marks.

Algorithm

- Step 1 : input the marks of three subjects. (ex m[3])
- Step 2 : calculate sum of three subjects (ex sum)
- Step 3 : calculate percentage.(ex p)
- Step 4 : calculate average of three subject. (ex av)
- Step 5 : print the percentage and average.
- Step 6 : exit.



Program-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float m[3],sum=0;
    float p,av,n;
    int i;
    printf("Enter marks of three subjects")

    for(i=0;i<3;i++)
    {
        printf("\n enter marks  of %d subject out of 100",+1);
        scanf("%f", &m[i]);
        sum=sum+m[i];
    }
    p =(sum/300)*100;
    av=sum/3;
    printf("\n average marks=%f percent=%f",av,p);
    getch();
}
```

Input :

enter marks of 1 subject out of 100-75

enter marks of 2 subject out of 100-79

enter marks of 3 subject out of 100-71

Output :

average marks =75 percent =75

Experiment No. 41

Aim- Write a Programme to find string is a palindrome or not.

Description- this program takes one string as input and check weather it is palindrome or not.

Algorithm :

Step 1 : input the string. (ex str)

Step 2 : calculate the length of the string (ex l)

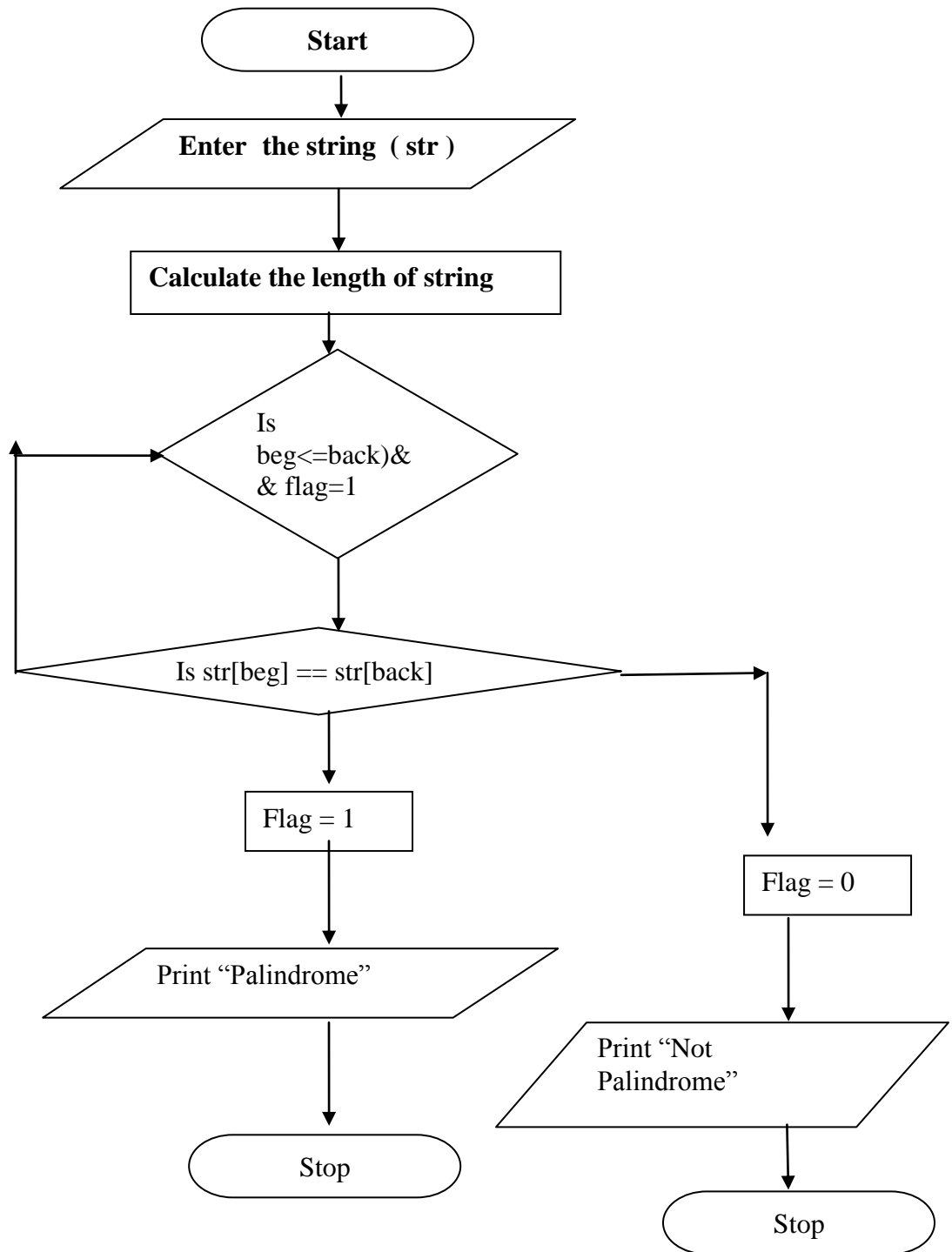
Step 3 : start a loop in which compare first and last character then second
And second last character and so on up to get middle position.

Step 4 : check the flag

If flag = 1 then print the message "String is Palindrome"

Otherwise print the message "String is not Palindrome"

Step 5 : Exit.



Program-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[80];
    int beg,back,l=0;
    int flag;
    printf("\n enter the string");
    scanf("%s", &str);
    while(str[l]!='\0')
        l++;
    flag=1;  beg=0;  back=l-1;
    while((beg<=back)&&flag)
    {
        if (str[beg]==str[back])
            flag=1;
        else
            { flag=0;  break;  }
        beg++;
        back--;
    }
    if (flag)
        printf("yes it is palindrome");
    else
        printf("no it is not palindrome");
    getch();
}
```

Input :

enter the string madam

Output :

yes it is palindrome

Experiment No. 42

Aim: Write a program which has the following options:

- a. Factorial of a number
- b. Prime or not
- c. Odd or even

Description:

In this Program variable **n** represents the number whose factorial is to be found and **f** represents the factorial of **n**. The variable **f** is initialized to **1** and then multiplied by the loop control variable **i**, **n** times. When the value of **i** becomes greater than **n**, the control exits the loop.

Algorithm:

Step1: Read n

Step2: Print 1. for Factorial, 2. for Prime and Not prime, 3. for Odd and Even

Step3: If the user inputs 1

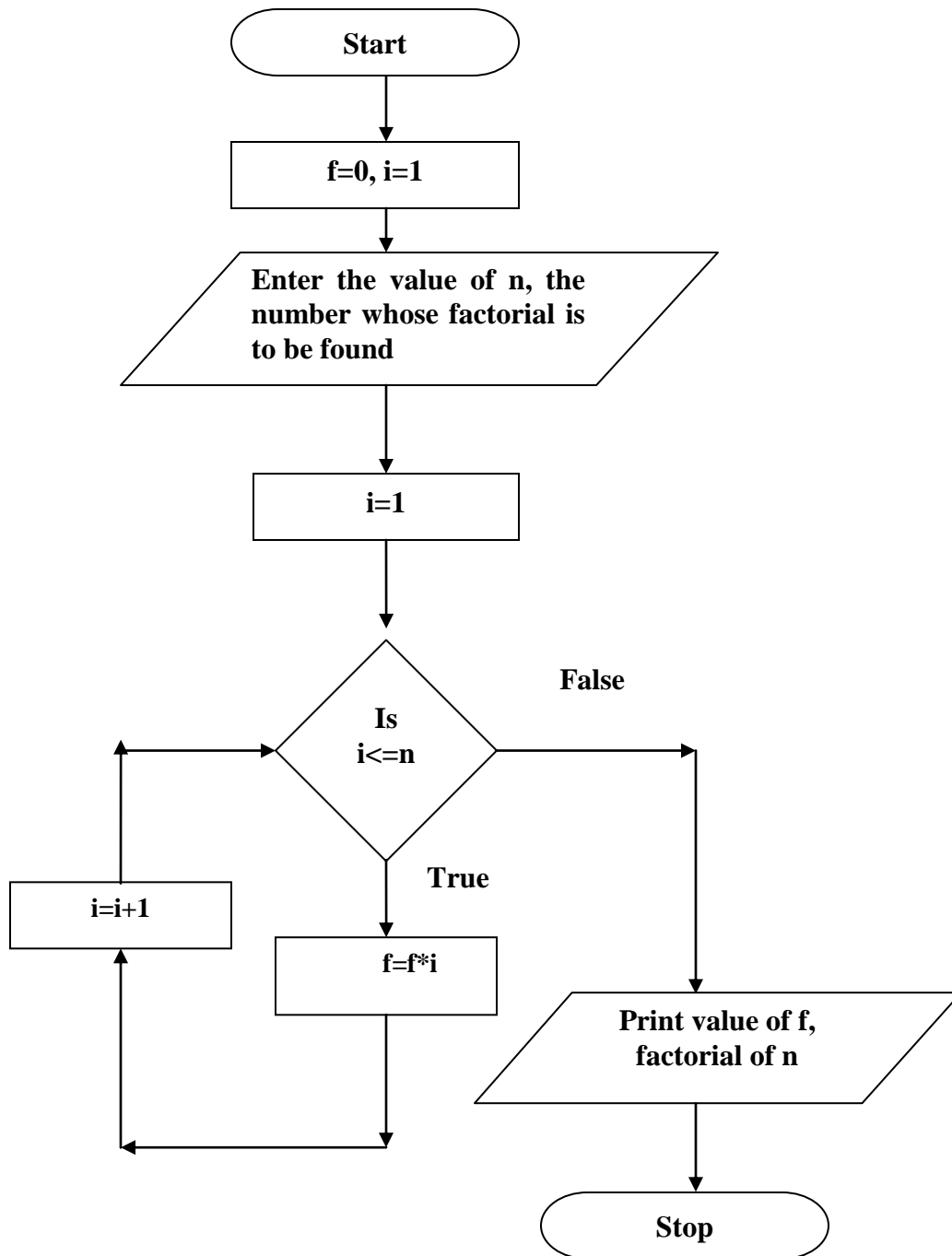
- a. Initialize $f = 1, i = 1$
- b. Repeat steps c, d while $i \leq n$
- c. Multiply the present value of i with f and store it in f .
- d. Increment the current value of i by 1.
- e. Print f

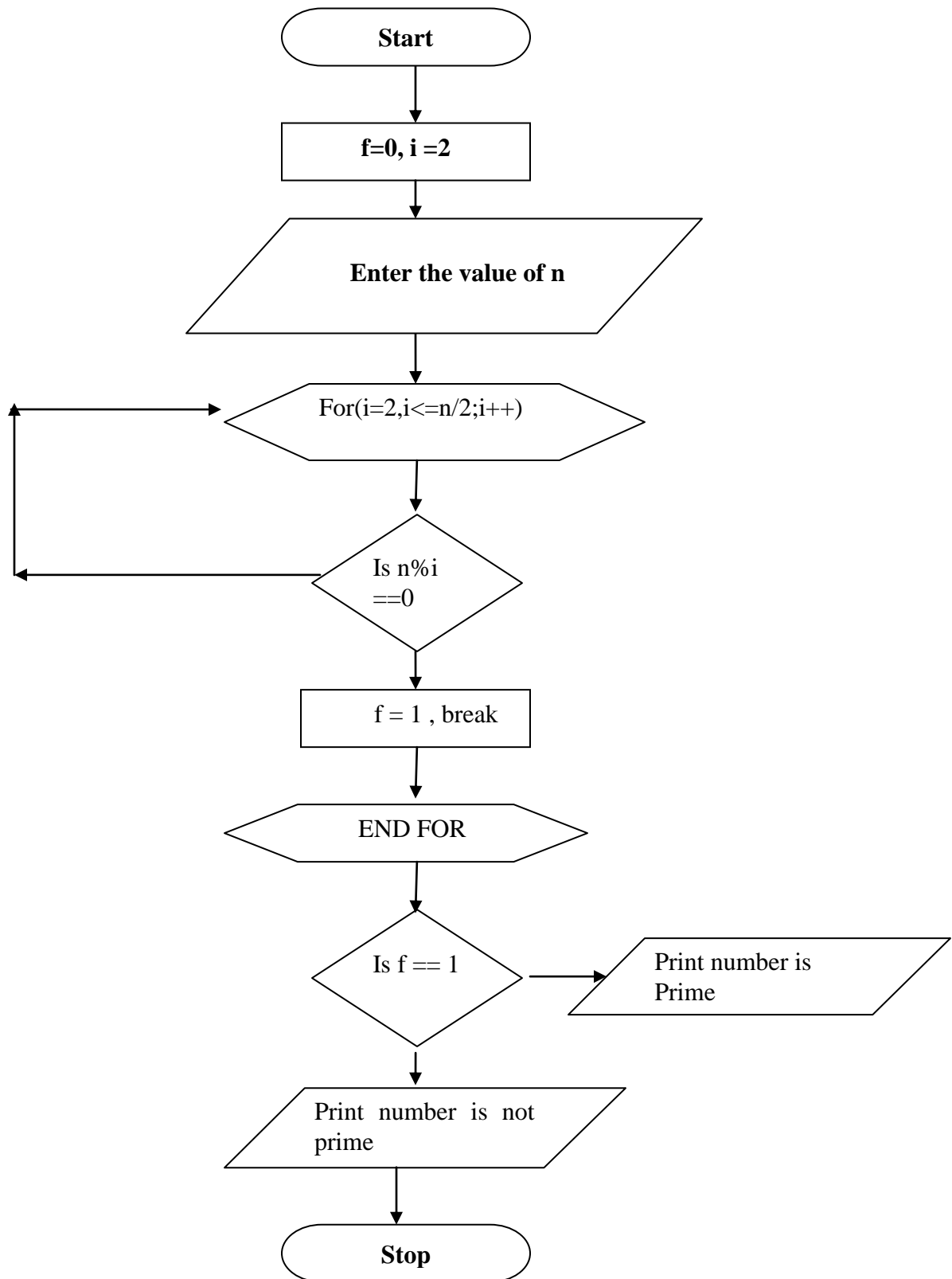
Step4: If the user inputs 2

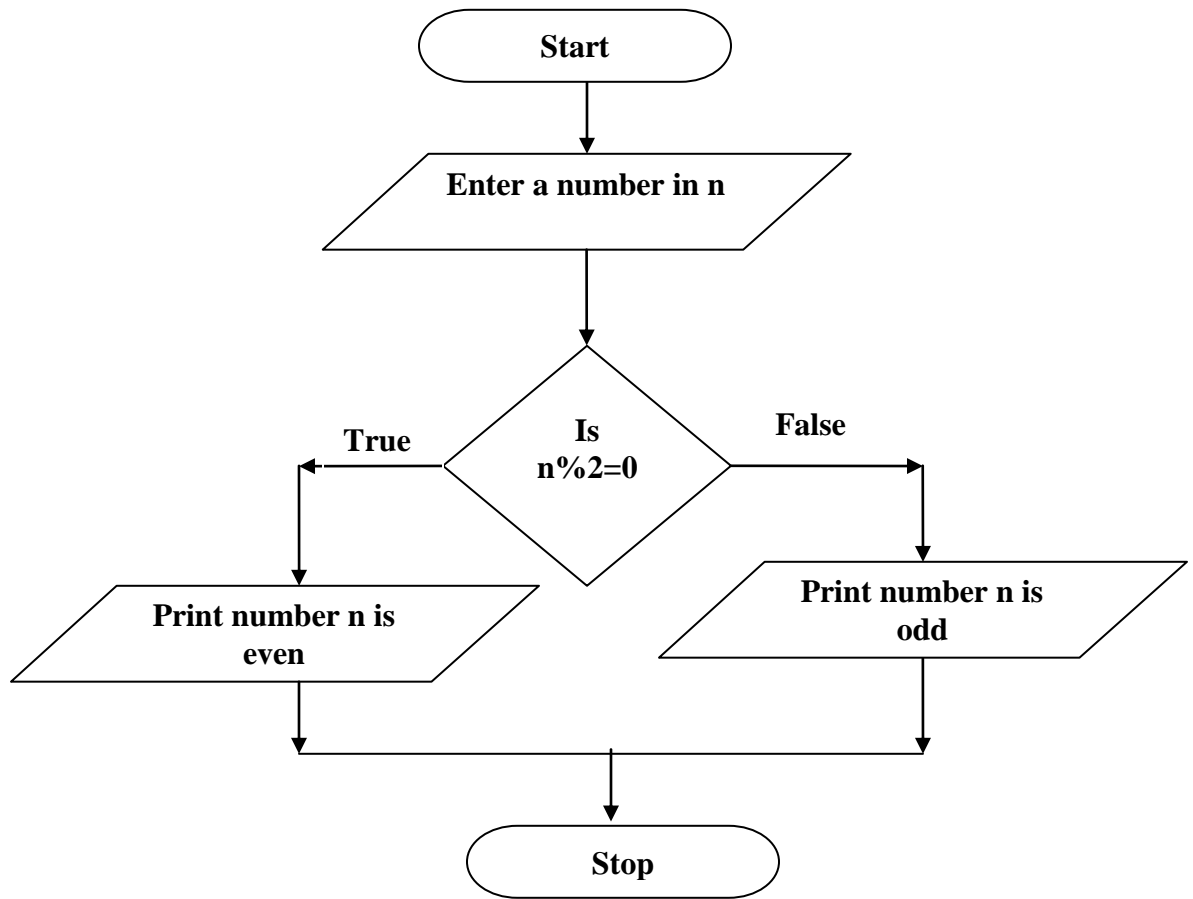
- a. Initialize $f = 0, i = 1$
- b. Repeat steps while $i \leq n$
- c. Divide n by i
- d. If the remainder is zero, increment value of f by 1.
- e. Increment the value of i by 1.
- f. If the value of $f = 2$, print "Number is prime" otherwise goto step h.
- g. Print "Number is not prime".

Step5: If the user inputs 3

- a. Divide n by i .
- b. If the remainder is zero, print "Number is Even" otherwise goto step c.
- c. Print "Number is Odd".







Program:

```
#include<stdio.h>
#include<conio.h>
void main ()
{
    int n, choice, i, j,n;
    long f;
    printf ("Enter any number:\n");
    scanf ("%d", &n);
    printf (" ***** MENU *****\n")
    printf ("1. Find Factorial\n");
    printf ("2. Check prime or not\n");
    printf ("3. Check even or odd\n");
    printf ("Enter your choice:\n");
    scanf ("%d", &choice");
    switch (ch)
    {
        case1:
            f = 1;
            for(i=1;i<=n;i++)
                f = f*i;
            printf ("Factorial of %d is: %ld",n,f);
            break;
        case2:
            f =0;
            for (i=2; i<=n/2; i++)
                if (n%i == 0)
                    {
                        f =1;
                        break;
                    }
            if ( f == 0)
                printf ("Number is Prime");
            else
```



```

                printf ("Number is not prime");
            break;
        case3:
            if(n%2==0)
                printf("\nGiven number is even");
            else
                printf("\nGiven number is odd");
            break;

        getch();
    }

```

Input :

```

Enter a Number:
5
***** MENU *****
1. Find Factorial
2. Check prime or not
3. Check even or odd

1

```

Output:

```

Factorial of 5 = 120

```

Experiment No. 43

Aim: Write a program to calculate the area & perimeter of the rectangle and the area & circumference of the circle. The length and breadth of a rectangle and radius of a circle are input through keyboard.

Description:

This program takes length and breadth of a rectangle, radius of circle as input. Calculates the area of rectangle and circle. Perimeter of rectangle and circumference of circle.

Algorithm:

Step1: Read length (l) and breadth (b) of rectangle.

Step2: Calculate $\text{rec_area} = l * b$

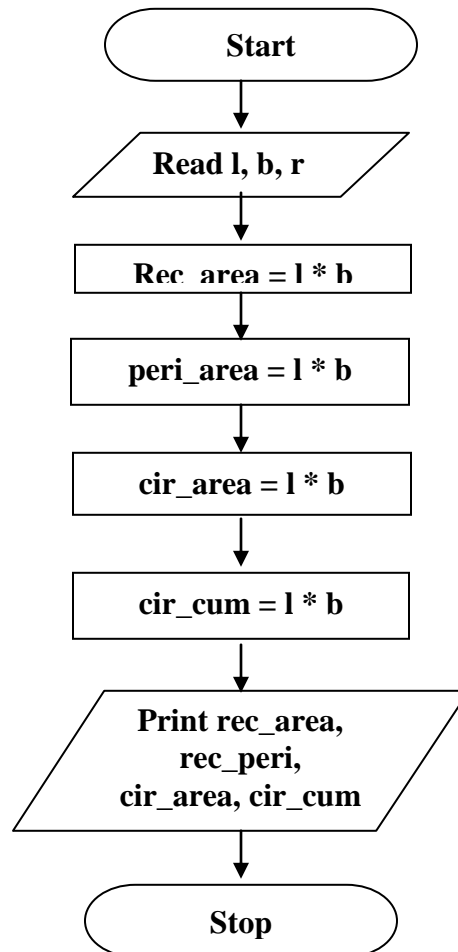
Step3: Calculate $\text{rec_peri} = 2 * (l + b)$

Step4: Read radius (r) of circle.

Step5: Calculate $\text{cir_area} = 3.14 * r * r$

Step6: Calculate $\text{cir_peri} = 2 * 3.14 * r$

Step7: Print the rec_area , rec_peri , cir_area , cir_peri



Program:

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14

void main()
{
    float l, b, r;
    float rec_area, cir_area, rec_peri, cir_cum;
    printf("Enter Length and Breadth of the Rectangle:\n");
    scanf("%d%d", &l, &b);
    rec_area = l * b;
    rec_peri = 2 * (l + b);
    printf("Enter Radius of Circle:\n");
    scanf("%d", &r);
    cir_area = PI * r * r;
    cir_cum = 2 * PI * r;
    printf("Area of Rectangle = %f", rec_area);
    printf("Perimeter of Rectangle = %f", rec_peri);
    printf("Area of Circle = %f", cir_area);
    printf("Circumference of Circle = %f", cir_cum);
    getch ();
}
```

Input:

Enter Length and Breadth of the Rectangle:

15

8

Enter Radius of Circle:

6

Output:

Area of Rectangle = 120

Perimeter of Rectangle = 46

Area of Circle = 113.04

Circumference of Circle =37.68

Experiment No. 44

Aim: Write a program to determine whether the character entered through a keyboard is a capital letter, a small case letter, a digit or a special symbol.

Description:

This program takes a character as an input using `getchar()` function. The functions `isupper`, `islower`, `isdigit` function is used to check the entered character whether it is in uppercase, lowercase, a digit respectively. These functions are used in the “if” statement and returns a non-zero value if the condition is true else zero. The header file required for these functions is “`ctype.h`”

Algorithm:

Step1: Read ch

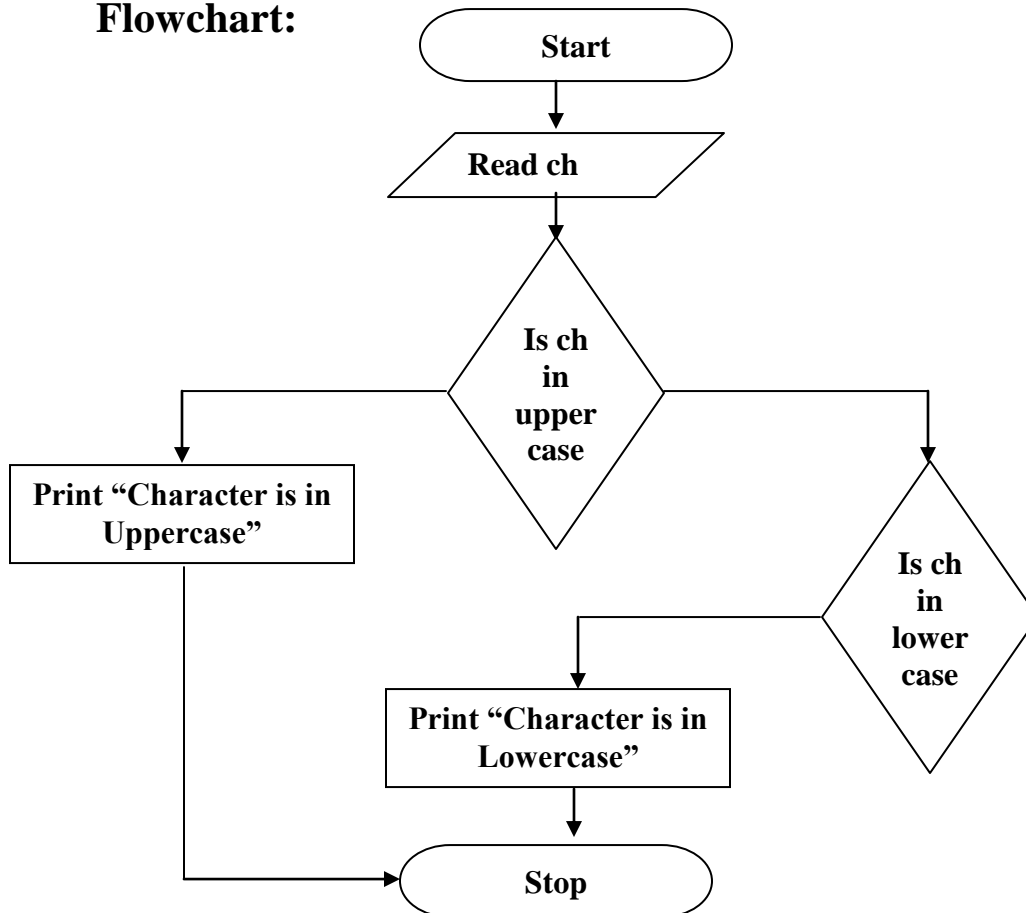
Step 2: Check if ch is in uppercase, print “Character is in Uppercase”, otherwise go to step3.

Step 3: Check if ch is in lowercase, print “Character is in Lowercase”, and otherwise go to step4.

Step4: Check if ch is a digit, print “Charcter is a Digit”, otherwise go to step5.

Step5: Print “Character is a Special symbol”.

Flowchart:



Program:

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
void main()
{
    char ch;
    printf("Enter a character to check whether it is in uppercase, lowercase, a digit or a special
symbol\n");
    getchar(ch);
    if(isupper(ch))
        printf("Character is in Uppercase\n");
    else if(islower(ch))
        printf("Character is in Lowercase\n");
    else if(isdigit(ch))
        printf("Character is a Digit\n");
    else
        printf("Character is a Special Symbol\n");

    getch();
}
```

Input:

Enter a character to check whether it is in uppercase, lowercase, a digit or a special symbol:
B

Output:

Character is in Uppercase

Input:

Enter a character to check whether it is in uppercase, lowercase, a digit or a special symbol
v

Output:

Character is in Lowercase

Input:

Enter a character to check whether it is in uppercase, lowercase, a digit or a special symbol
5

Output:

Character is a Digit

Experiment No. 45

Aim: Write a program to add first seven terms of the following series using looping statements series is

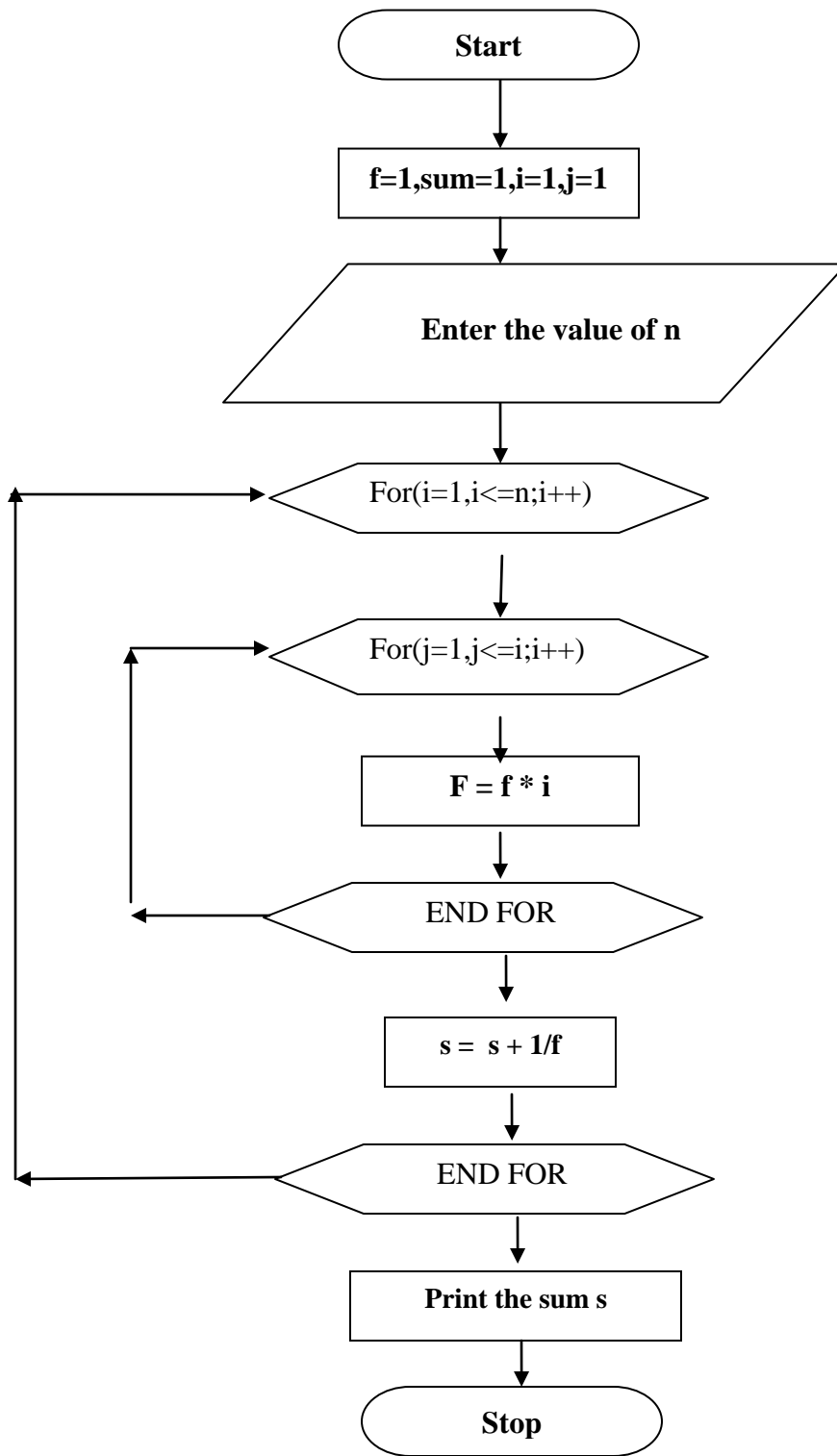
$$S = 1/1! + 2/2! + 3/3! + \dots\dots\dots$$

Description:

In this program ‘**i**’ and ‘**j**’ are the loop variables. Variable ‘**f**’ is initialized to 1 and is used to store the factorial of ‘**i**’ after calculation. ‘**sum**’ is the variable initialized to zero and finally after 7 times executing the loop sum of the series is stored in ‘**sum**’.

Algorithm :

- Step 1 : Initialize the variable f=1,sum=0;
- Step 2 : Initialize i=1,j=1
- Step 3 : Input the value of n
- Step 4 : Repeat step 5 to 9 while i<=7
- Step 5 : Repeat step 6 to 8 while j<=i
- Step 6 : Compute f = f * i
- Step 7 : Compute s = s + 1/f
- Step 8 : increment the value of j by 1.
- Step 9 : increment the value of i by 1.
- Step 10 : Print the value of sum s.
- Step 11 : Exit.



Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,n, j, f =1;
    float sum = 0;
    printf("Enter the value of n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        for (j=1;j<=i;j++)
            f = f * j;
        S = S + 1/f;
    }
    printf("Sum of the Series = %f", sum);
    getch ();
}
```

Input:

Enter the value of n 7

Output:

Sum of Series = 2.721

**CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY,
BHILAI (C.G.)**

Semester: B.E. III Sem.
Subject: Problem Solving & Logic Building
Using C Lab
Total practical Periods: 36
Total Marks in End Semester Exam: 40

Branch: Information Technology
Practical Code : 333322(33)

List of programmes to be executed (but should not be limited to):

1. Write a program to take the radius of a sphere as input and print the volume and surface and surface area of that sphere.
2. Write a program to take a 5-digit number as input and calculate the sum of its digits.
3. Write a program to take three sides of a triangle as input and verify whether the triangle is an isosceles, scalene or an equilateral triangle.
4. Write a program that will take 3 positive integers as input and verify whether or not they form a Pythagorean triplet or not.
5. Write a program to print all the Prime numbers between a given range.
6. Write a program to define a function that will take an integer as argument and return the sum of digits of that integer
7. Write a program to define a macro that can calculate the greater of two of its arguments. Use this macro to calculate the greatest of 4 integers.
8. Write a program to define a recursive function that will print the reverse of its integer argument.
9. Write a program to print the sum of first N even numbers using recursive function.
10. Write a program to sort an array using Bubble sort technique.
11. Write a program that will take the elements of two integer arrays of 5 element each, and insert the common elements of both the array into a third array (Set intersection)
12. Write a program to take 5 names as input and print the longest name.
13. Write a program to define a structure Student that will contain the roll number, name and total marks of a student. The program will ask the user to input the details of 5 students and print the details of all the students whose total marks is greater than a given value.
14. Write a program to define a union Contact that will contain the members Mobile no and E-mail id. Now define a structure Employee that will contain name, roll number, mode of contact (mob/e-mail) and a variable of type Contact as members. The program will ask the user to give the details of two Employees including mode of contact and the contact num/ E-mail. Print the details of both the Employees.
15. Write a program that will ask the user to input a file name and copy the contents of that file into another file.
16. Write a program that will take any number of integers from the command line as argument and print the sum of all those integers.
17. Write a program to process sequential file for payroll data.
18. Write a program to process random file of library data.

List of Equipments/Machine Required:

PCs, C-Compiler

Recommended Books:

Programming in ANSI C – E. Balaguruswamy Tata Mc-Geraw Hill