Date-

<u>Assignment No.</u> :

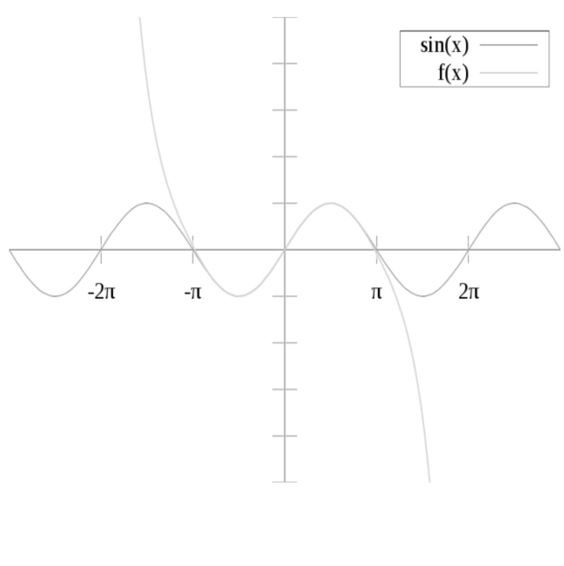
<u>Problem Statement:</u>

Program in C to find the value of sin function with the sin series and to determine the error in between the value by series and value by library function.

<u>Theory:</u>

In mathematics, a Taylor series is a representation of a function as an infinite sum of terms that are calculated from the values of the function's derivatives at a single point. A function can be approximated by using a finite number of terms of its Taylor series. Taylor's theorem gives quantitative estimates on the error introduced by the use of such an approximation. The polynomial formed by taking some initial terms of the Taylor series is called a Taylor polynomial. The Taylor series of a function is the limit of that function's Taylor polynomials as the degree increases, provided that the limit exists. A function may not be equal to its Taylor series, even if its Taylor series converges at every point. A function that is equal to its Taylor series in an open interval (or a disc in the complex plane) is known as an analytic function in that interval.

Pictured on the right is an accurate approximation of sin x around the point x = 0. The pink curve is a polynomial of degree seven:



$$\text{Sinx} \approx x - x^3/3! + x^5/5! + \ldots..$$

The error in this approximation is no more than $|x|^9/9!$. In particular, for $-1 < x < 1$, the error is less than 0.000003.

In contrast, also shown is a picture of the natural logarithm function log(1 + x) and some of its Taylor polynomials around a = 0. These approximations converge to the function only in the region $-1 < x \leq 1$; outside of this region the higher-degree Taylor polynomials are worse approximations for the function. This is similar to Runge's phenomenon.

The error incurred in approximating a function by its nth-degree Taylor polynomial is called the remainder or residual and is denoted by the function $R_n(x)$. Taylor's theorem can be used to obtain a bound on the size of the remainder.

Algorithm:

Input specification: The magnitude of the angle for which sin series is to be calculated say, X.

Output specification: 1. Value by sin series and value by library function.

2. Error calculation by taking the mod value of the difference between library function value and sin series value.

Steps:

Step 1: [Starting of Do-While loop]
Set t= (3.141593*x)/180
Step 2: Set sine=t
Step 3: Set sign=-1
Step 4: Set index=3
Step 5: Repeat through step 6 to step 15 While(numer /denom<0.00001)
Step 6: Set numer=pow(t,index)
Step 7: Set denom=1
Step 8: Repeat step 9 to 10 For j=1 to index
Step 9: Set denom=denom*j
Step 10: Set i=i+1
[End of For loop]
Step 11: Set term=(sign*numer)/denom
Step 12: Set sine=sine+term

Step 13: Set index=index+2
Step 14: Set sign=sign*-1
Step 15: Set c=sin(t) //sin(t) is a inbuilt library function which is used to
//calculate the value of sin function corresponding to degree t
Step 16: Set error=sine-c
Step 17: Print "Value of sine"x
Step 18: Print "As per series="sine
Step 19: Print "As per library function "c
Step 20: Print "Error ="error
Step 21: Print "Do you want to continue:(Y/N)?"
Step 22: Input ch
Step 23: If(ch=='n' OR ch=='N') Then
Step 24: Stop
Step 25: Repeat through step 1 to step 24 While (ch=='y' OR ch=='Y')
[End of Do-While loop]
Step 26: End

Source Code:

```
#include<stdio.h>
#include<math.h>
int main()
{
	char ch;
	do
	{
		int i, j, denom, index, sign;
		float term, sum, x, sine, numer, error, c,t;
		printf("Enter the value of x:");
		scanf("%f",&x);
		t= (3.141593*x)/180;
		sine=t;
		sign=-1;
		index=3;
		while(1)
```

```
        {
                numer=pow(t,index);
                denom=1;
                for(j=1;j<=index;j++)
                denom=denom*j;
                term=(sign*numer)/denom;
                if(numer /denom<0.00001)
                        break;
                sine=sine+term;
                index=index+2;
                sign=sign*-1;
        }
        c=sin(t);
        error=fabs(sine-c);
        printf("\nValue of sine %f",x);
        printf("\nAs per series=%f",sine);
        printf("\nAs per library function %f", c);
        printf("\nError =%f",error);
        printf("\nDo you want to continue:(Y/N)?");
        fflush(stdin);
        scanf(" %c",& ch);
        if(ch=='n'|| ch=='N')
                return 0;
    } while (ch=='y'|| ch=='Y');
    return 0;
}
```

Input & Output:

Set 1:

```
Enter the value of x:45

Value of sine 45.000000
As per series=0.707107
As per library function 0.707107
Error =0.000000
Do you want to continue:(Y/N)?
```

Set 2:

```
Enter the value of x:87

Value of sine 87.000000
As per series=0.998632
As per library function 0.998630
Error =0.000002
Do you want to continue:(Y/N)?
```

Set 3:

```
Enter the value of x:360

Value of sine 360.000000
As per series=-459.409424
As per library function 0.000001
Error =459.409424
Do you want to continue:(Y/N)?
```

Discussion:

1. We are going to assume that the series representation will converge to *f(x)* on $-L \le x \le L$. We will be looking into whether or not it will actually converge in a later section.
2. The series representation will not involve powers of sine (again contrasting this with Taylor Series) but instead will involve sine's with different arguments.
3. In set 3 when the input angle crosses 90 the error becomes higher. It is because the higher terms for which the sin function needs to be calculated for that angle are ignored. That's why when x=180 or 360 then error is very high.