**ASSIGNMENT NO**.

**PROBLEM STATEMENT:**

Program in C to find the value of sine function for a particular angle given in degree corrected upto four decimal places.

**THEORY:**

The sine of the acute angle is defined with reference to a right angled triangle. It is the ratio of length of perpendicular and the hypotenuse.

Hence , sin Ɵ=

The sine series expansion can be written as:

Sin Ɵ=

Where x is the degree of angle in radian. We will use this infinite series expansion to approximate the value of sine for a given angular value.

Ex: Finding the value of sin 30 by infinite series expansion. Let’s say we are to find the value of sin(30º).

= (0.52) – (0.0234) + (0.0003) – (2.03 x 10-6) + ...

= 0.49689 + ...

~ 0.5.

In this way, we can easily approximate the value of sine function for a given angle upto our desired precision.

**ALGORITHM:**

**Input :** The value of angle in degree to calculate the sine function for, say deg.

**Output :** The value for sine function for given degree of angle.

**Steps :**

1. Print "Enter the value of angle(in degree) : "
2. Input deg
3. Set rad = (Pi/180)\*deg // Pi is a constant which stores the value of pi upto sufficient precision
4. Set sine = rad, sign = -1, index = 3
5. Repeat through step 6 to 13 while(True)

Step 6: Set numer = radindex, denom = 1, j = 1

Step 7: Repeat through step 8 to 9 while (j <= index)

Step 8: Set denom = denom \* j

Step 9: Set j = j + 1

[End of inner while loop]

Step 10: If(Abs(numer/denom) < 0.0001) // Abs is a function which returns

// the absolute value of a number

Then

Step 11: Break

[End of if structure]

Step 12: Set term = (sign \* numer)/denom

Step 13: Set sine = sine + term, index = index + 2, sign = sign \* -1

[End of outer while loop]

Step 14: Set orig = Sin(rad) // Sin is a library function which generates the

// value of sine function for a given angle

Step 15: Set error = Abs(sine - orig)

Step 16: Print "The value of sin(" rad ") as per series : " sine

Step 17: Print "The value of sin(" rad ") as per library function : " orig

Step 18: Print "Absolute error : " error

**SOURCE CODE:**

#include <stdio.h>

#include <math.h>

#define PI acos(-1.0)

int main(){

float deg;

printf("Enter the value of angle (in degree) : ");

scanf("%f", &deg);

float rad = (PI/180) \* deg; // converting degree to radian

float sine = rad; // intiailzing first term

int sign = -1, index = 3;

while(1){

float numer = pow(rad, index); // x^index

float j = 1, denom = 1;

while(j <= index)

denom \*= j++; // index!

if(fabs(numer/denom) < 0.0001) // precision controller

break;

float term = (sign \* numer)/denom;

sine = sine + term, index = index + 2, sign \*= -1;

}

float orig = sin(rad); // value from library function

float error = fabs(orig-sine); // absolute error

printf("The value of sin(%g) as per series : %g", rad, sine);

printf("\nThe value of sin(%g) as per library function : %g", rad, orig);

printf("\nAbsolute error : %g\n", error);

return 0;

}

**INPUT AND OUTPUT:**

**Set 1 :**

Enter the value of angle (in degree) : -30

The value of sin(-0.523599) as per series : -0.500002

The value of sin(-0.523599) as per library function : -0.5

Absolute error : 2.14577e-06

**Set 2 :**

Enter the value of angle (in degree) : 67

The value of sin(1.16937) as per series : 0.920494

The value of sin(1.16937) as per library function : 0.920505

Absolute error : 1.11461e-05

**Set 3 :**

Enter the value of angle (in degree) : 90

The value of sin(1.5708) as per series : 1

The value of sin(1.5708) as per library function : 1

Absolute error : 3.45707e-06

**DISCUSSION:**

1. The computational complexity of this method is very high.
2. The program will depend on the size of datatypes on the machine it runs. Hence it won’t give correct and equal result on every machine.
3. On the extension of above point, this program won’t be able to calculate the sine function for higher values because of type overflows.