

```

/*#####*/
/* HW03_Samet_Sait_Talayhan_101044044.c */
/* ----- */
/* Created on March 19, 2013, 1:52 PM by Samet Sait Talayhan. */
/* ----- */
/* Description */
/* ----- */
/* An example calculation of sin z function that gets the parameter
 * z and n return the sinus of z for the first n terms.
 * ----- */
/* ----- */
/*#####*/
/*
 * File: main.c
 * Author: neo
 *
 * Created on March 19, 2013, 1:52 PM
 */
/*#####*/
/* Includes */
/*#####*/
#include <stdio.h>
#include <stdlib.h> /* for exit() function */
#include <math.h> /* for math functions i.e pow,sin */

/*#####*/
/* Defines */
/*#####*/
#define PI 3.14159265

/*-----*/
/* Function Implementations */
/*-----*/
/* factorial Function */
/* ----- */
/* Input - int: number
 * Output - double: Result of the factorial value of the given number.
 *-----*/
int
factorial(int numb)
{
    int result = 1;

    if(numb == 0) /* if numb is equal zero, 0! = 1 */
    {
        return 1;
    }
    else if(numb < 0 ) /* Negative factorial does not occur */
    {
        printf("Error Message!\n");
        exit(1);
    }
    else /* Calculate factorial */
    {
        while( numb > 0 )
        {
            result *= numb;
            --numb;
        }
        return result;
    }
}

/*-----*/
/* Function Implementations */
/*-----*/
/* mySinusFunc Function */
/* ----- */

```

```

/* mySinusFunc function that gets the parameter
 * z and n return the sinus of z for the first n terms.
/* Input - int: n , double z
/* Output - double: Approximately sinus value
/*-----*/
double
mySinusFunc(int n, double z)
{
    int count = 0;
    double result = 0; /* return value */

    for(count = 0; count < n; ++count)
    {
        result += pow(-1,count) * pow(z,(2*count+1)) / factorial((2*count+1));
    }

    return result;
}
/*-----*/
/*          Function Implementations          */
/*-----*/
/* calculateMinError Function                */
/*-----*/
/* This function comparing the result, mySinusFunc and "math.h"
/* sin function. And return mininum error terms.
/* Input - double: maxError, double z
/* Output - int: .
/*-----*/
int
calculateMinError(double maxError, double z)
{
    int count = 0;

    while( fabs(mySinusFunc(count,z*PI/180) - sin(z*PI/180))
           / sin(z*PI/180) > maxError )
    {
        count++;
    }
    return count;
}

/*#####*/
/* int main() */
/* ----- */
/* Return */
/* ----- */
/*0 on success */
/*#####*/
int main(int argc, char** argv) {

    FILE *matris1Fptr;
    FILE *matris2Fptr;
    FILE *outputFptr;
    int readNumb1,readNumb2, count=0;
    int n;
    double z;
    double mySinDouble = mySinusFunc(10,30*PI/180);
    double theSinDouble = sin(30*PI/180);

    printf("Enter radian value: ");
    scanf("%lf",&z);

    printf("How many times the loop turn?: ");
    scanf("%d",&n);

    mySinDouble = mySinusFunc(n,z*PI/180);
    theSinDouble = sin(z*PI/180);

```

```
printf("theSin :%f\n",mySinDouble);
printf("mySin :%f\n",theSinDouble);

if( fabs(theSinDouble - mySinDouble) < 0.1 )
    printf("Works Correctly!\n");
else
    printf("Doesnt Work!\n");

/* Test calculateMinError function */
printf("%d\n",calculateMinError(0.001,z));

matris1Fptr = fopen("matris1.txt","r");
matris2Fptr = fopen("matris2.txt","r");
outputFptr = fopen("sumMatrix.txt","w");

fclose(matris1Fptr);
fclose(matris2Fptr);
return (EXIT_SUCCESS);
}
/*#####*/
/*      End of HW03_Samet_Sait_Talayhan_101044044.c      */
/*#####*/
```