ENGINEERING H192
DAILY ASSIGNMENT B11

Many times obtaining a rigorous mathematical solution to a problem can be very difficult, but an approximate result of sufficient accuracy may be obtained by replacing the complicated expression with a series expansion called the Taylor Series. Retaining just enough terms to give the desired accuracy reduces the computation time. The cosine of an angle may be computed by the series expansion,

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \ldots = \sum_{n=1}^{\infty} (-1)^{(n+1)} \frac{x^{2(n-1)}}{2(n-1)!}$$

in which the angle **x** is expressed in radians, **4!** is 4 factorial, and **n** is the term number, 1, 2, 3, etc.

Complete the following **C** language program that has been started for you on the following page. You have been provided with a **main ()** function and must write:

1. A function called **myfactorial ()** which takes an integer argument and returns the argument's factorial as a long integer data type
2. A function called **mycos ()** which takes two arguments. The first is the angle in degrees and the second is the number of terms to use in the Taylor Series expansion

Function prototypes for **myfactorial ()** and **mycos ()** are given.

The main program contains a **for** loop in which the angle varies from **0 to 90** degrees in **5** degree increments. As the loop executes it:

1. Invokes the **mycos ()** function to compute your estimate of sin(x)
2. Invokes the built in **cos ()** function to compute a value for cos(x) against which you can compare the results of **mycos ()**.
3. Prints the angle in degrees and the results of **mycos ()** and **cos ()** to the screen.

Results are displayed by the main program in the Linux **xterm** window. As you complete the logic of the program, add blank lines for legibility and include additional comments.

Experiment with various numbers of terms in the Taylor Series expansion. What number of terms produces results that compare best with the built in **cos ()** function? Do you run into a problem if the number of terms is too high? If so, what do you think may be the cause of the problem? Answer these questions as comments at the end of your program.

Name the C program file **b11.cpp** and the executable file **b11.out**. When the program is running and producing correct results, redirect the output to a text file with the command **b11.out > b11out.txt**. Submit printouts of the program file and the output file along with this sheet.

Name_____Instructor_____Seat_____Hour_____

ENGINEERING H192
DAILY ASSIGNMENT B11

```
/*
    ENG H192 Winter Quarter 2007
    Daily Assignment B11
    Prepared by Scarlet Anne Gray (Replace with your name)
    Calculates cosines of angles from 0 to 90 degrees using
    a Taylor Series expansion of the cos () function
*/

#include <stdio.h>
#include <math.h>
#define PI 3.14159265359   /* PI is a symbolic constant */

/*
    Function prototype for my estimate of cos (x).  The function expects the
    angle in radians and the number of terms to expand the Taylor series as
    arguments.  The function returns a float.
*/
float mycos (float x, int terms);
/*
     Function prototype for factorial function.  The function expects the number
     to determine the factorial of as its argument.  The function returns a long
     integer.
*/
long myfactorial (int fact);

int main ()
{
  int angle_deg, num_terms;
  float angle_rad, my_estimate, built_in;
  printf ("\n ENG H192 Daily Assignment B11-Scarlet Anne Gray\n");
  printf (" This program computes the cosines of angles\n");
  printf (" from 0 to 90 degrees at 5 degree intervals.\n");

  printf ("\n\nNumber of terms in Taylor Series expansion: ");
  scanf ("%d", &num_terms);

  for (angle_deg = 0; angle_deg <= 90; angle_deg = angle_deg + 5)
    {
      angle_rad = PI * angle_deg/180.;
      /* A - call 'mycos()' to calculate my estimate of cos(x) */
      my_estimate = mycos (angle_rad, num_terms);
      /* B - call 'cos()' to estimate cos(x) using the built in function */
      built_in = cos (angle_rad);
      /* C - display the angle, my estimate, and built in estimate */
      printf ("\n Angle = %4d  mycos = %10.6f  cos = %10.6f", angle_deg,
      my_estimate, built_in);
    }
  printf ("\n\n");
}
```

Name_____Instructor_____Seat_____Hour_____