EECS 10: Assignment 2

Prepared by: Che-Wei Chang, Prof. Rainer Doemer

August 12, 2013

Due Monday 19 Aug 2013 at 23:00 pm

1 Homework Problem 1: Compute the approximate value of tan(x) [25 Points]

Write a C program to calculate the value of tan x. The result can be approximated using an infinite sum (Taylor expansion, x is in radian NOT in degree):

1a) Use Taylor series of Sine and Cosine functions below and compute sin(x) and cos(x) for any given value of x. For good precision, use first 5 terms in the Sine or Cosine series.

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} + \ldots$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} + \ldots$$

1b) Use Taylor series of Tangent function below to compute tan(x) for given input value x. Use 7 terms from the Tan series as shown below.

$$\tan(x) = x + \frac{x^3}{3} + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \frac{62}{2835}x^9 + \frac{1382}{155925}x^{11} + \frac{21844}{6081075}x^{13} + \ldots$$

1c) Divide Sin(x) by Cos(x) computed in step 1a and print along with the result of Tan(x) computed in Step 1b.

Your program must use only the basic operations such as addition, subtraction, multiplication and division. Also, please follow the same programming style as discussed in Lecture 2.2 for the cosine function (i.e. do not use any loops in your program). In order to make your code simple for higher power terms, precompute x^2, x^3, x^5, x^6 in temporary variables, and use them throughout. Use a maximum of 4 multiplications for each term in all the series above. (Hint: $x^8 = x^5 * x^3; x^{11} = x^6 * x^5$)

When executed, your program output should look as follows:

```
Please enter the real value x (in radians):0.8
Approximately,
Sin(0.800000) = 0.717356,
Cos(0.800000) = 0.696707
Approximately,
Tan(0.800000) = 1.029569,
Sin(0.800000)/Cos(0.800000)=1.029639
```

Note: All variables declared and used need to be of type ”long double”. Use appropriate type specifiers while printing.

You should submit your program code as file **`tan.c`**, a text file **`tan.txt`** briefly explaining how you designed your program, and a typescript **`tan.script`** which shows that you compile your program and run it using the values -0.7, 0.6 and 1.2 as inputs.

Note that for the first part of this assignment, you have to name your files
`tan.c`,

tan.txt and,
tan.script.

2 Bonus: Boundary values checking for tan(x) [5 points]

Tan(x) is almost linear in the range $-\frac{\pi}{2}+\varepsilon$ to $\frac{\pi}{2}-\varepsilon$. As x approaches $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, tan(x) reaches $-\infty$ and ∞ respectively. Hence add boundary check conditions such that tan(x) program in previous problem accepts input "*x*" values only within $-\frac{\pi}{2}+\varepsilon$ to $\frac{\pi}{2}-\varepsilon$.

In this problem, choose π as 3.1416 and ε as 0.2708, so that tan(x) produces valid output for x within range $-1.3 < x < 1.3$. If x is outside this range, calculate and print sin(x) and cos(x), but print the following error message and exit the program with status 1.

```
Tan(x) could not be computed. Enter input x within range -1.3 to  1.3.
```

When executed, your program output should look as follows:

```
Please enter the real value x (in radians):1.4
Approximately,
Sin(1.400000) = 0.985451,
Cos(1.400000) = 0.169975
Tan(x) could not be computed.
Enter input x within range -1.3 to 1.3.
```

NOTE: Include the code for this problem in the same **tan.c** file that you submit for Problem 1. In the **tan.txt** file, explain the logic that you used for checking boundary values. Your **tan.script** should contain program output for x = -2.4, -0.7, 0.6, 1.2, 1.5

3 Homework Problem 2: Calculate the weekday for any date [25 Points]

Zeller's congruence (source: http://en.wikipedia.org/wiki/Zeller's_congruence is an algorithm devised by Christian Zeller to calculate the day of the week for any calendar date. For todays Gregorian calendar, Zeller's congruence is

$$w = (d + \lfloor \frac{(m+1)*26}{10} \rfloor + K + \lfloor \frac{K}{4} \rfloor + \lfloor \frac{J}{4} \rfloor + 5J)\ mod\ 7, \text{where}$$

w is the day of the week (0 means Saturday, 1 means Sunday, 2 means Monday, and so on)
d is the day of the month ($1 \leq d \leq 31$)
m is the month ($1 \leq m \leq 12$), and
y is the year of the calendar date ($1582 \leq y \leq 2014$).
Further, the above equation distinguishes
J as the century (that is, $J = \lfloor \frac{y}{100} \rfloor$) and
K as the year of the century (that is, *K* = y mod 100).

Finally, there is an exception in Zellers congruence for the months of January and February which need to be counted as month 13 and 14, respectively, of the previous year. Thus, if *m*=1 or *m*=2, then we need to add 12 months to the value of *m*, and subtract 1 year from *y* before we feed the values into the above equation.

Your weekday calculation program should contain the following sections:

1. Data input: Let the user enter a valid calendar date in the following format:

```
Please enter a calendar date:
Day     d=19
Month   m=10
Year    y=2009
```

We assume that the user will always enter proper input values, e.g. d will not be greater than 31. Therefore, there is no need to handle any invalid input in your program.

2. Data preprocessing: Handle the exception for the months of January and February.
   That is, if m $<$ 3 then add 12 to m and subtract 1 from y.

3. Computation: Use Zellers congruence
   Hint: The floor function $\lfloor x \rfloor$ is implicit in any integer division.
   That is, if a and b are both integer variables, then $\lfloor \frac{a}{b} \rfloor = a/b$

4.Output the numerical result: Use the following format:

```
For the date 10/19/2009, the day of the week is 2.
This is a Monday.
```

The output of the Zeller's congruence equation is a numerical identifier (0-6). To print the result similar to the format above, you have to convert the numerical identifier to a text string.
   Hint: you may use seven **`if`** statements or a **`switch`** statement to create this output.

You should submit your program code as file **`weekday.c`**, a text file weekday.txt briefly explaining how you designed your program, and a typescript weekday.script which shows that you compile your program and run it. Use the following dates as inputs:
8/19/2013 (the deadline for this assignment),
1/1/2014 (next New Year), and
10/04/1965 (the first day of classes at UCI).

For the second part of this assignment, you have to name your files as
`weekday.c`,
`weekday.txt` and,
`weekday.script`.

4 Submission

Submission for these files will be similar to last week's assignment. The only difference is that you need to create a directory called **`hw2/`**. Put all the files for assignment 2 in that directory and run the **`/ecelib/bin/turnin10`** command to submit your homework.