EECS 10: Homework 2

Prepared by Weiwei Chen, Prof. Rainer Dömer

July 2, 2012

Due Monday, July 9, 2012 at 11:00pm

1. Part 1: Compute the approximate value of e^x (20 points)

Write a C program to calculate the value of *e* to the power of *x*. The result can be approximated using an infinite sum:

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + ... + \frac{1}{n!}x^n + ...$$

Your program should use only the basic operations such as addition, subtraction, multiplication and division. Also, please follow the same programming style as discussed in Lecture 2 for the cosine function (i.e. do not use any loops in your program).

The goal is to compute the value of e^x such that the result has a precision of 3 decimal places. For example, if the value of $e^{0.9}$ = 2.459603111…, then your program should output
$e^{0.9}$ = 2.459xxxx (where "x" is any digit, no matter whether it is accurate or not).
In your program, you should use as many terms from the above formula as necessary to just achieve the above mentioned precision for the three values given below.

$e^{0.4}$ = 1.491xxx
$e^{0.7}$ = 2.013xxx
$e^{1.0}$ = 2.718xxx

When executed, your program should look like this:

```
Please enter the real value x: 1.0
e to the power of x is approximately 2.718xxx
```

You should submit your program code as file **`e.c`**, a text file **`e.txt`** briefly explaining how you designed your program, and a typescript **`e.script`** which shows that you compile your program and run it using the values 0.4, 0.7 and 1.0 as inputs.

2. Part 2: Calculate the weekday for any date (30 points)

Zeller's congruence (source: http://en.wikipedia.org/wiki/Zeller%27s_congruence) is an algorithm devised by Christian Zeller to calculate the day of the week for any calendar date. For today's Gregorian calendar, Zeller's congruence is

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

w is the day of the week (0 means Saturday, 1 means Sunday, 2 means Monday, …),
d is the day of the month ($1 \le d \le 31$),
m is the month ($1 \le m \le 12$), and
y is the year of the calendar date ($1582 \le y \le 2099$).
Further, the above equation distinguishes

J as the century (that is, $J = \left\lfloor \frac{y}{100} \right\rfloor$) and

K as the year of the century (that is, $K = y \bmod 100$).

Finally, there is an exception in Zeller's 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
   if $m < 3$ then add 12 to m and subtract 1 from y

3. Computation: Use Zeller's 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 $\left\lfloor \frac{a}{b} \right\rfloor = a / b$.

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

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

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:
7/9/2012 (the deadline for this assignment),
11/6/2012 (next Election Day), and
10/04/1965 (the first day of classes at UCI).

3. Bonus Problem [5 Points]

The program output of Part 2 uses a numerical identifier (0 – 6) to represent the computed weekday. In this bonus problem, we will extend our program so that it prints the result as a regular text string, i.e. Saturday, Sunday, Monday, …, or Friday. Thus, given the date listed above, the program should output:

```
For the date 7/9/2012, the day of the week is 2.
This is a Monday.
```

Hint: You may use seven if-statements to create this output.
To submit, use the same files as in Part 2, i.e. **`weekday.c`**, **`weekday.txt`**, and **`weekday.script`**.

4. Submission

Submission for the 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 in the parent directory of hw2/ to submit your homework.