EECS 10: Assignment 2

Prepared by: Prof. Rainer Doemer

June 30, 2017

Due Monday July 10, 2017 at 11 pm

1 Homework Problem 1: Compute the approximate value of e^x [25 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:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \ldots + \frac{1}{n!}x^n + \ldots$$

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 output should look as follows:

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

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

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.

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

`e.c`,
`e.txt` and,
`e.script`.

2 Bonus Problem [5 Points]

For the above problem of calculating e^x, minimize the number of multiplication operations used in your approximation, which will make your implementation more efficient. In other words, avoid any duplication of multiplications.

With your optimizations applied, your program should still calculate the same values as above (with the same precision).

To submit your improved program, use the same files as above, i.e. **`e.c`**, **`e.txt`**, and **`e.script`**.

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 today's 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 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 (1-31)       d=10
Month (1-12)     m=7
Year (1582-2014) y=2017
```

We assume that the user will always enter proper input values, e.g. the day number 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 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 $\lfloor \frac{a}{b} \rfloor = a/b$

4. Output the numerical and textual results: Use the following format:

```
For the date 6/29/2017, the day of the week is 5.
This is a Thursday.
```

The output of the Zeller's congruence equation is a numerical value in the range from 0 to 6, where 0 represents Saturday, 1 means Sunday, and so on. To print the result in textual format for human understanding, you have to convert the numerical value to a text string.
   Hint: you may use seven **`if`** statements to create this textual 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:

7/10/2017 (the deadline for this assignment),
7/4/2017 (this year's Independence Day), and
10/04/1965 (the first day of classes at UCI).

For the second part of this assignment, name your files as

`weekday.c`,
`weekday.txt` and,
`weekday.script`.

4 Submission

Submission for these files is the same as for last week's assignment. The only difference is that you work in a directory called **`hw2`**. Put all the files for assignment 2 in the **`hw2`** directory and run the **`~eecs10/bin/turnin.sh`** command to submit your homework.