

## CS100M Lab 2

When you have completed the lab, show this sheet and any associated programs to your lab instructor, who will record that you have completed the lab. If you do not finish this exercise during the lab, show the instructor what you have done at the end of the lab and be sure to complete it in the next few days.

If you have any questions, **ask** your lab instructor or a consultant immediately! They are in the lab to help you learn the material.

## CMS: Course Management System

You will use CMS to submit homework, find your scores, and view grading comments. If you had *pre*-enrolled, then your student information had been entered into CMS already and you are ready to use CMS. If you added CS100M very recently, then you must identify yourself in section to ask your instructor to add your information into the system.

Try to log in into CMS now. Use the course webpage to access CMS. *If you are not able to log in you must tell your section instructor now.* Homework must be submitted through CMS.

### 1 Leap year and leap day

Since the earth takes slightly more than 365 days to revolve around the sun, our Gregorian calendar needs to make up those few hours by adding an extra day to the year (February 29th) periodically. These years with an added day are called leap years. The extra day is called a leap day. A year is a leap year if it is divisible by four with the exception of century years that are not divisible by 400. For example, the years 1992 and 2000 are leap years, but the years 1993 and 2100 are not.

The program below is an attempt to calculate the number of leap days between January 1st, 1900, and December 31st in a year that the user inputs. However, the program does not always compute the correct number of leap days. Find the error(s) and make your correction(s) *on paper*. (Note: this is good practice for exams—you will write exams without the aid of a computer.) After your pen-and-paper analysis, type up the program and test it with several different input. Call the script `leapDays.m`. To learn what a built-in function does, type `help <functionname>`, e.g., `help ceil`.

```
% Count the number of leap year days between January 1, 1900, and
% December 31 of the inputted year in the 21st century, inclusive.

numberOfLeapDays= 0;          % the number of leap days

year= input('Enter a year from 1900 to 2099, inclusive: ');

% calculate the number of leap days
if (year < 2000)
    numberOfLeapDays= ceil((year - 1900) / 4);
else
    numberOfLeapDays= floor((year - 1900) / 4) + 1;
end

fprintf('# of leap year days from 1/1/1900 to 12/31/%d = %d\n', ...
        year, numberOfLeapDays);
```

Test your program with several different input. Finally, change your program to calculate the number of leap days from January 1st, 1900, to any year up to the end of the 22nd century. The last year would then be 2199.

## 2 Function Evaluation

Write three different programs (scripts) to determine in which quadrant a user-input value of  $A$  degrees belong. Assume that the user may enter any non-negative number. For example,  $725^\circ$  is the same, and should be treated, as  $5^\circ$ . (Hint: the function `mod` that you saw last week might be useful.) To avoid ambiguity, we use the following convention:

$$\text{Quadrant is } \begin{cases} 1 & \text{if } 0 \leq A < 90 \\ 2 & \text{if } 90 \leq A < 180 \\ 3 & \text{if } 180 \leq A < 270 \\ 4 & \text{if } 270 \leq A < 360 \end{cases}$$

Print the result. In the first script use four *separate* `if` statements (4 separate `if-end` constructs) and call the program `angle1.m`. In the second script, use a *single* `if-elseif-else-end` construction for the evaluation and call it `angle2.m`. In the third script, use *nesting* without using the `elseif` clause and call it `angle3.m`. Pay close attention to the differences among the three programs.

## 3 Submitting text files in CMS

The files that you create using MATLAB's editor window are *plain text files*, i.e., there is no formatting information associated with any character in the file. For example, there is no *italic*, **bold**, or other character format. You can create plain text files using applications such as Notepad or Wordpad. MATLAB's Editor also produces plain text files. Be sure to submit plain text files whenever we ask you to submit MATLAB files with the filename extension `.m`.

To make sure that you understand the difference between plain text files and files with formatting, create two files, one using *Notepad* and the other using *Microsoft Word*, containing the same text. You can use any text, but here's a suggestion:

```
"There's nothing sillier in the world, I say, than being a devil in  
despair." (Mephistopheles)
```

```
Faust -- Johann Wolfgang Von Goethe.
```

Save both files in their default format. Use *Windows Explorer* to find out their sizes (in bytes). Which one is smaller? Now close both files and then try to open the *Microsoft Word* file using *Notepad*, a plain text editor. What do you see?

If you have time left in the lab ...

Go to the course website and click on the **CMS** link. *Read the instructions if you haven't done so already. In particular, note that you must form your partner group **before** you submit any file.*

Submit your solution to any one of the Project 1 questions in CMS. If you work with a partner, *do this only if you've already registered your group in CMS*. You can do this even if your solution isn't complete yet. You can submit another file later; the later submission will overwrite the previous file. If you have questions/problems with submitting files in CMS, *ask now!* You must submit your Project 1 in CMS on time.

**Please delete your files from the computer before leaving the lab!**