

COMP 1800 – Fall 2016

Homework 5: Functions

(40 points)

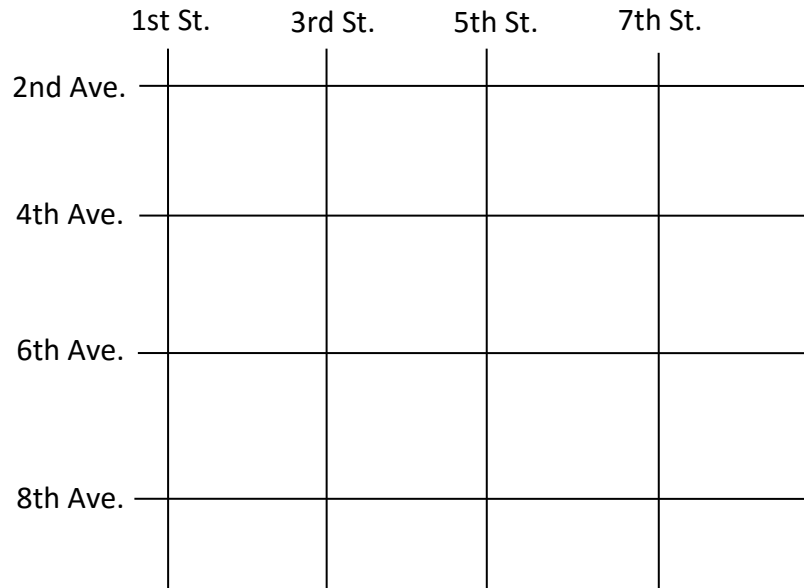
Number of People: Individual. Feel free to ask me for help, or visit the Computer Science Learning Center (http://www.memphis.edu/cs/current_students/csclc.php).

Due: Tues., Nov. 29 by 5:30 pm

Submission: Zip your Python source code files and submit the zip file to the proper folder on eCourseware (<https://elearn.memphis.edu>).

Grader: TA, Swaroop Goli (ssgoli@memphis.edu). Questions about grading? Please contact him first!

Suppose you live in a city where the roads form a grid as shown below. “Streets” run north-south and are numbered 1st St., 3rd St., 5th St., etc. from west to east. “Avenues” run west-east and are numbered 2nd Ave., 4th Ave., 6th Ave., etc. from north to south. This city is quite large; the highest-numbered street is 99th St., and the highest-numbered avenue is 98th Ave.



Assume that the roads are all equally spaced, 1000 ft apart. Also assume that you cannot cut across any blocks when traveling. For example, to get from the corner of 1st and 6th to the corner of 3rd and 4th, you would have to travel north on 1st and east on 4th, or east on 6th and north on 3rd. Both routes have a total traveling distance of 2000 ft.

1. (4 pts) Write a function **getDistance(s1, a1, s2, a2)** that returns the distance traveled when you go from the corner of street **s1** and avenue **a1** to the corner of street **s2** and avenue **a2**. For example, calling **getDistance(1, 6, 3, 4)** would compute the distance traveled from 1st St. and 6th Ave. to 3rd St. and 4th Ave. This should return a value of 2000.0. (Hint: Consider the horizontal and vertical distances separately.)
2. (6 pts) Write a function **ordinal(n)** that returns a string representing the *n*th ordinal number. For example, calling **ordinal(1)** should return the string “1st”, calling **ordinal(2)** should return the string “2nd”, calling **ordinal(3)** should return the string “3rd”, and so on. Your function should work for any value of *n* between 1 and 99, inclusive. (Hint: The most straightforward way to do this is to write a multibranch conditional with 99 branches. Please don’t do this ☺ Note that with a few exceptions, you can determine the appropriate suffix to add based on the ones (rightmost) digit of the number. How can you isolate the ones digit? The % operator might be handy...)

3. (20 pts) Write a function **getDirections(s1, a1, s2, a2)** that prints step-by-step directions for traveling from the corner of street **s1** and avenue **a1** to the corner of street **s2** and avenue **a2**. Include the direction and distance traveled, as well as any turns involved. Within this function, you can call your previously written **ordinal** function to format the street/avenue names nicely. For example, calling **getDirections(1, 6, 3, 4)** should display something like this:

Directions from 1st and 6th to 3rd and 4th:

Take 6th Ave. east for 1000.0 ft until you get to 3rd St.
Turn left onto 3rd St.
Take 3rd St. north for 1000.0 ft until you get to 4th Ave.
You have arrived at your destination!

(Hint: This may seem very challenging, but it's not too bad! Consider the horizontal and vertical directions separately. The direction of the turn will depend on your horizontal and vertical directions. For example, traveling west and then south involves a left turn. What other combinations are possible?)

4. (10 pts) Within the same file where you've written your functions, write a complete mapping program that allows the user to enter a starting street/avenue and an ending street/avenue. Include error checking to ensure that the street is an odd number between 1 and 99, and the avenue is an even number between 2 and 98.

The program should display directions between those points (call your **getDirections** function), as well as the total distance traveled (call your **getDistance** function). Also allow the user to repeat the program if desired.

Here's an example of what your program might look like when you run it. The underlined portions indicate what you type in as the program is running.

Starting street: 0
Street must be a positive odd number between 1 and 99!
Starting street: 87
Starting avenue: 144
Avenue must be a positive even number between 2 and 98!
Starting avenue: 20
Ending street: 1
Ending avenue: 8

Directions from 87th and 20th to 1st and 8th:

Take 20th Ave. west for 43000.0 ft until you get to 1st St.
Turn right onto 1st St.
Take 1st St. north for 6000.0 ft until you get to 8th Ave.
You have arrived at your destination!

Total distance traveled: 49000.0 ft

Enter 1 to get more directions, any other key to exit... 1
Starting street: 3
Starting avenue: 4
Ending street: 3
Ending avenue: 8

Directions from 3rd and 4th to 3rd and 8th:

Take 3rd St. south for 2000.0 ft until you get to 8th Ave.
You have arrived at your destination!

Total distance traveled: 2000.0 ft

Enter 1 to get more directions, any other key to exit... 2