

# Homework 3

## Reading

Read Chapters 4 and 5 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

## Logistics

If you want to do the assignment on your own computer, you will need to download and install Python. There is a link for downloading Python under Content -> Course Information on D2L.

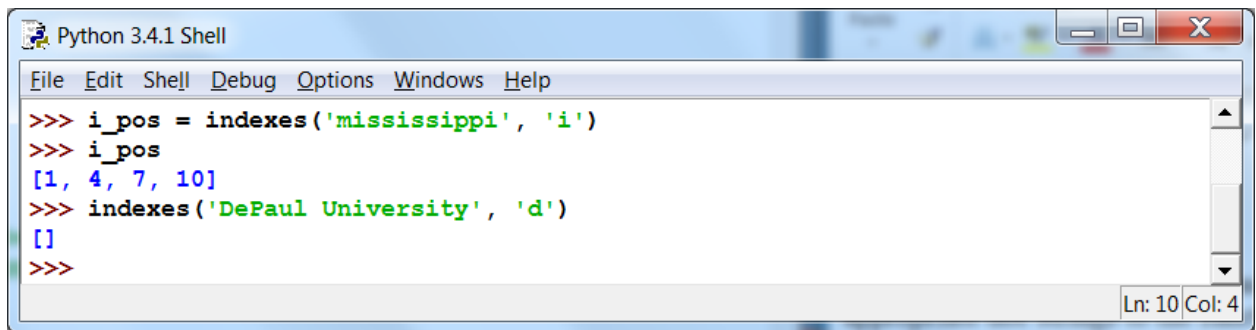
In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in the comment box when you submit on D2L. You must also list their names at the top of the assignment. The total number of collaborators on any assignment **may not exceed two other people**. Please see the Collaboration Guidelines document found under Content -> Course Information on D2L for more details on what is or is not allowed when working on homework.

Remember that **everyone** submitting the assignment must be able to explain **all** of the code submitted, regardless of the type of collaboration that occurred. Anyone submitting code they cannot explain is violating the Academic Integrity policy and will earn a 0 on the assignment.

## Assignment

To begin the assignment, download the **HW3.py** template found on the D2L site. You will need to add appropriate doc strings to the functions there as well as complete the implementation as described below. **Submissions without doc strings will not earn full credit.**

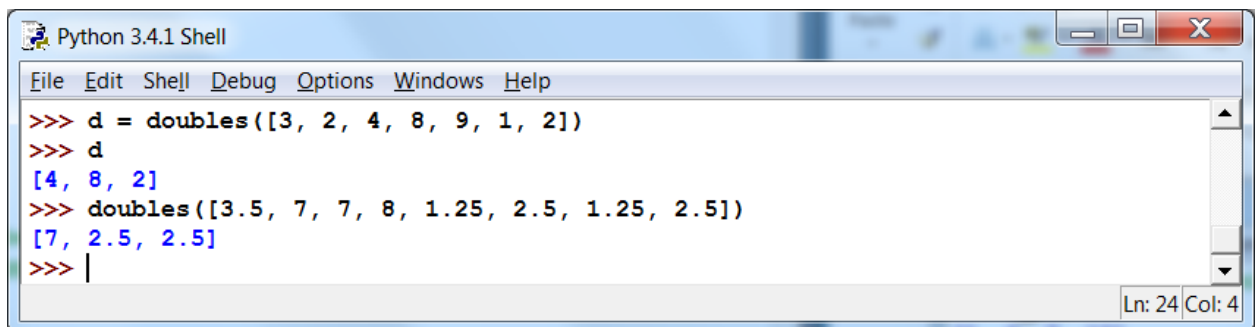
1. Write a function **indexes()** that takes a string and a character (string of length 1) and returns the indices at which the character appears in the string. The indices are returned as a list of integers and must occur in increasing order. Below is an example of how this function might work.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> i_pos = indexes('mississippi', 'i')
>>> i_pos
[1, 4, 7, 10]
>>> indexes('DePaul University', 'd')
[]
>>>
```

Ln: 10 Col: 4

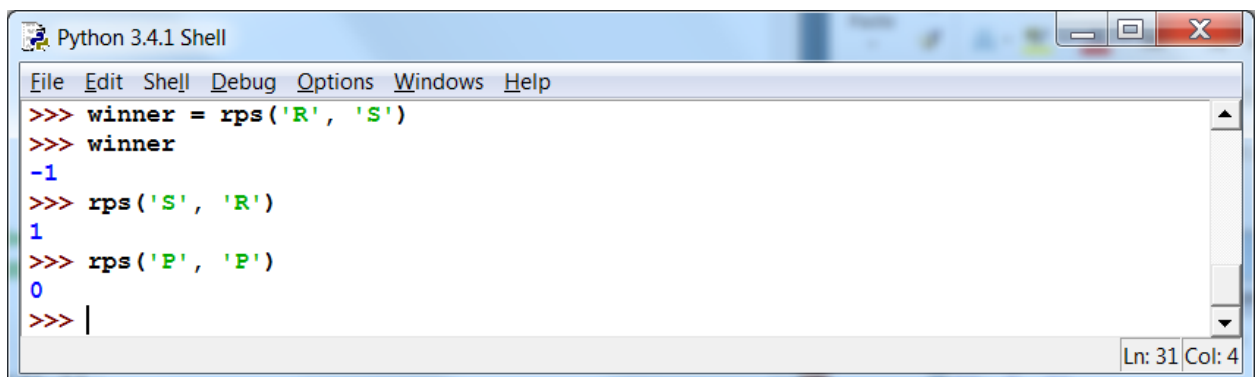
2. Write a function called **doubles()** that takes a a list of numbers as an argument and returns a list of those numbers that are double the value of the previous number in the argument list. The numbers in the output list occur in the same order that they occur in the input list. Below is an example of how this function might work.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> d = doubles([3, 2, 4, 8, 9, 1, 2])
>>> d
[4, 8, 2]
>>> doubles([3.5, 7, 7, 8, 1.25, 2.5, 1.25, 2.5])
[7, 2.5, 2.5]
>>> |
```

Ln: 24 Col: 4

3. Write a function **rps()** that takes two string arguments. Each argument is either 'P', 'R', or 'S' and stand for rock paper scissors. The function returns -1 if the first argument wins the game of rock, paper, scissors. It returns 1 if the second argument wins. It returns 0 if it is a tie. See problem 5.26 in the book for more details. Below are a few examples of how the function works.

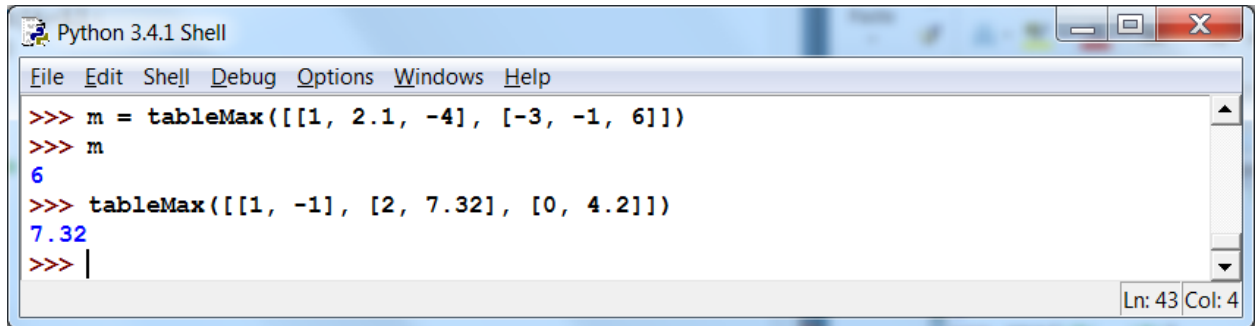


```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> winner = rps('R', 'S')
>>> winner
-1
>>> rps('S', 'R')
1
>>> rps('P', 'P')
0
>>> |
```

Ln: 31 Col: 4

4. Write a function **tableMax()** that takes a 2D table of floats as an argument. (Recall that a table is represented as a list of lists where all the lists have the same size.) The function

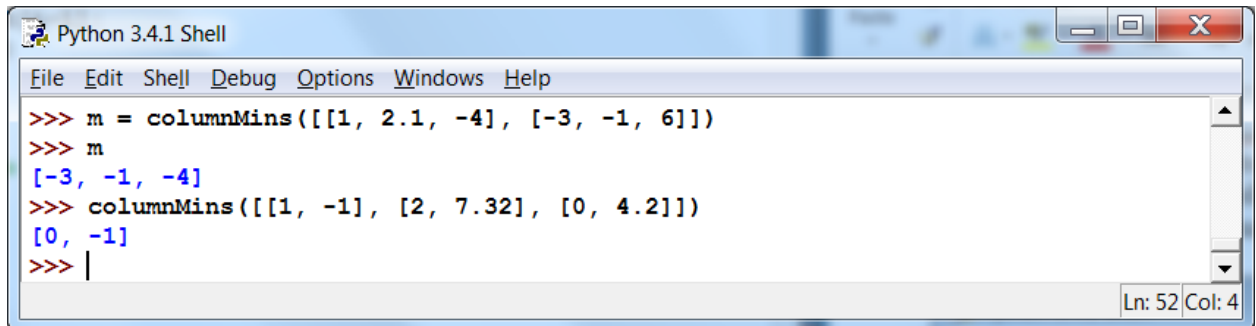
returns the maximum value in the table. Below are a few examples of how the function works.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> m = tableMax([[1, 2.1, -4], [-3, -1, 6]])
>>> m
6
>>> tableMax([[1, -1], [2, 7.32], [0, 4.2]])
7.32
>>> |
```

Ln: 43 Col: 4

5. Write a function **columnMins()** that takes a 2D table of floats as an argument. . (Recall that a table is represented as a list of lists where all the lists have the same size.) The function returns the list of the minimum value in each column. (Note that the output list will be same size as each of the rows of the table.) Below are a few examples of how the function works.



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> m = columnMins([[1, 2.1, -4], [-3, -1, 6]])
>>> m
[-3, -1, -4]
>>> columnMins([[1, -1], [2, 7.32], [0, 4.2]])
[0, -1]
>>> |
```

Ln: 52 Col: 4

## Submitting the assignment

You must submit the assignment using the HW3 folder of the dropbox on D2L. **Submit only a single Python file (HW3.py)** with your solutions found in it. Submissions after the deadline will be automatically rejected by the system and will receive a grade of 0.

## Grading

Each of the five problems is worth 20 points for a total of 100 points.