CS 111 – Introduction to Computer Science – Fall 2017

Lab Assignment #9

Lists [*] *(30pts)*

Due Date: at 11:59pm on Saturday, April 21.

This lab provides an opportunity to work with the list container. You will write several simple functions using lists.

⚠ Before getting started with the lab, copy the entire `lab9` folder from the course folder (`H:\Compsci\givens\cs111`) to your `U:\cs111` folder.

List Things

📝 Create a new file `listThings.py` and add the following functions, then use another file to test the functions with the requested lists.

- `product(values)`
  Returns the product of all the elements in the list or 1 if the list is empty. Does not alter the list.

- `countOdds(values)`
  Returns the number of odd numbers in the list. Does not alter the list.

- `squares(values)`
  Returns a new list containing the squares of each value in the original list. For instance, if the list contains
  `[8, 12, -3, 5, -2]`,
  then the function returns
  `[64, 144, 9, 25, 4]`.
  Does not alter the original list.

- `computeAltSum(values)`
  Computes and returns the *alternating sum* of the integer elements in the list. For instance, if the list contains
  `[8, 12, -3, 5, -2]`,
  then the sum is calculated as
  `8 - 12 + (-3) - 5 + (-2)`.
  Does not alter the list.

[*] Based on the labs of Dr. Rance Necaise

- `replaceNegatives(values)`
  Alters the list by replacing each negative value in the list to 0. For instance, if the list contains
  `[8, 12, -3, 5, -2]`,
  then the list is altered to be
  `[8, 12, 0, 5, 0]`.
  Does not return anything.

- `shiftRight(values)`
  Shifts all the elements one position to the right and moves the last element to the first position. For instance, if the lsit contains
  `[8, 12, -3, 5, -2]`,
  then the list is altered to be
  `[-2, 8, 12, -3, 5]`.
  Does not return anything.

Your functions must meet the following requirements.

- Include an appropriate file prolog at the top of your source file.
- Include a comment immediately above each of the two functions that you write to briefly describe the purpose of the function.
- Use meaningful variable names.
- Include appropriate comments throughout the function definitions.
- The module will not contain a *main* function. Nor should it contain any executable code outside of the functions. To test your functions, use the module `testListThings.py`.

■

Testing

Complete the tester `testListThings` to test your `listthings` module. Use the following lists to to test your functions.

```
list1 = [10,17,5,-3,9,-27,5]
list2 = [8, 12, -3, 5, -2]
list3 = []
```

You should expect the following results:

	`list1`	`list2`	`list3`
`product`	3098250	2880	1
`countOdds`	6	2	0
`squares`	[100, 289, 25, 9, 81, 729, 25]	[64, 144, 9, 25, 4]	[ ]
`computeAltSum`	42	-14	0
`replaceNegatives`*	[10, 17, 5, 0, 9, 0, 5]	[8, 12, 0, 5, 0]	[ ]
`shiftRight`*	[5, 10, 17, 5, -3, 9, -27]	[-2, 8, 12, -3, 5]	[ ]

*These functions do not return anything, but alter the list.

Finishing Up

When you are finished with the lab, you need to show me that your code runs and correctly computes the solution for each part of the lab. Also, you need to submit the source files for grading. To submit the files, find the lab assignment on Canvas and upload the two files:

- `listThings.py`
- `testListThings.py`

Remember, all of the files must be named exactly as indicated above, with the same case and with no spaces or special characters.