CS103 – Fall 2025 – Lab07

General Instructions

1. In today's lab, you are required to complete all required exercises during class to receive attendance credit.
2. You are welcome to:
    - Work together with classmates.
    - Search online for help or documentation.
    - Ask the TA for guidance if you are stuck.
3. To receive credit, you must finish and show your solutions to the TA before leaving lab.
4. There are also extra (optional) exercises at the end for those who want to challenge themselves. These are not required for attendance credit but are recommended for practice.

Exercise Instructions

- Make a folder **Lab7** inside your **cs103fa25** folder.
- Create a new python file inside your Lab7 folder (lab07.py).***
-

***Important Note:

This week, we are switching from **Jupyter Notebook** **to** **VSCode** for all Python programming labs and homework.

- Please review the **VSCode primer** before starting this lab to ensure your environment is set up correctly.
- All lab submissions will now be **.py** files instead of notebooks.

Required Exercises

EXERCISE 1:

Write a function "**`cashier`**" that receives an int **`"c"`** for the number of items bought. The function will ask the `price` and `tax rate` of each item and prints out the total cost.

Sample Input

```
c = 2
```

Expected Output

```
For item 1
Enter the price: 10
Enter the tax rate: 0
For item 2
Enter the price: 20
Enter the tax rate: 8
Your total price is 31.6
```

Sample Input 2

```
c = 3
```

Expected Output 2

```
For item 1
Enter the price: 100
Enter the tax rate: 8
For item 2
Enter the price: 20
Enter the tax rate: 0
For item 3
Enter the price: 40
Enter the tax rate: 8
Your total price is 171.2
```

EXERCISE 2: Write the function "**`starLine`**" that takes a positive int "**`n`**" and prints "**`*`**" characters. The output should print **`n`** lines of stars. The number of stars in the line should be equal to the number of line. (For example: line 3 has 3 stars, line 4 has 4 stars… and so on)

Sample Input 1

```
n = 7
```

Expected Output 1

```
*
**
***
****
*****
******
*******
```

Sample Input 2

```
n = 3
```

Expected Output 2

```
*
**
***
```

Sample Input 3

```
n = 1
```

Expected Output 3

```
*
```

EXERCISE 3:

Write a function "**`stringPair`**" that receives two strings "**`s1`**" and "**`s2`**" and prints out each character of the strings side by side. Assume that both strings will have the same number of characters.

Sample Input 1

```
s1 = 'abc'
s2 = 'xyz'
```

Expected Output 1

```
a x
b y
c z
```

Sample Input 2

```
s1 = 'pmtn'
s2 = 'iuea'
```

Expected Output 2

```
p i
m u
t e
n a
```

Extra (Optional) Challenges

These are not required for credit but will help you practice and strengthen your problem-solving skills.

Challenge 1: smallestSum

Write a function **"smallestSum"** that accepts two lists, **L1** and **L2**, of the same length *(assume the lists have the same length all the time)*. The function should find and return the **smallest possible sum** for the entries in the corresponding positions, i.e. the same index in two lists.

Sample Input

```
L1 = [0, 1, 8, 12, 65, 34534]

L2 = [120, 2, 4, 55, 323, 12]
```

Expected Output

```
3 ***Hint: 1+2=3 (the min index for these numbers is 1)
```

Challenge 2: Write a function "**listCompare**" that takes two list **"l1" and "l2"**and compare these lists. The function should display all missing and extra values in l2

Sample Input

```
l1 = ["a", "b", "c", "d", "e"]
l2 = ["b", "d", "e", "f", "g"]
```

Sample Output

```
The missing values: a, c
The extra values: f, g
```

Challenge 3: Heron's Algorithm

Implement Heron's Algorithm to calculate the square root of a float number. Accept tolerance value as 0.000001. (Do not use the script file provided in the class, create your own methods)

Challenge 4: sumUnique(nums)

Given a list of integers nums, write a function sumUnique(nums) that returns the sum of all elements that appear **exactly once**.

- Use only loops and conditionals

Sample Function Calls:

```
Input: nums = [1,2,3,2]
Output: 4
Explanation: 1 and 3 appear exactly once, sum = 1 + 3 = 4
```

To get attendance credit, finish Exercises 1–3.

If you finish early, try the optional challenges!