CS103 – Fall 2025

Lab04 Exercises

Objectives:

- To practice working with strings and lists in Python.
- To understand indexing, slicing, and modifying strings.
- To apply functions to real-world style problems.
- To reinforce function writing with conditionals.

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 **Lab4** inside your **cs103fa25** folder.
- Create a new notebook inside your Lab4 folder (lab04.ipynb).

Required Exercises

Exercise 1: reverseString(s)

Write a function reverseString(s) that takes a string and returns the reversed version of it.

Sample Function Call:

```
>>> reverseString("CS103")

"301SC"

>>> reverseString("hello")

"olleh"
```

Exercise 2: countVowels(s)

Write a function countVowels(s) that takes a string and returns the number of vowels (a, e, i, o, u) in the string.

Sample Function Call:

```
>>> countVowels("Computer Science")

6

>>> countVowels("Python")

1
```

Exercise 3: indexLast(numbers)

Write a function indexLast(numbers) that takes a list of integers and returns the index of the last number.

Sample Function Call:

```
>>> indexLast ([10, 20, 30, 40])
3
>>> indexLast ([-5, 5, 10])
2
```

Exercise 4: maxOfThree(a, b, c)

Write a function maxOfThree(a, b, c) that takes three integers and returns the largest of the three.
Do **not** use Python's built-in max() function — solve with if/else.

Sample Function Call:

```
>>> maxOfThree(3, 7, 5)
7
>>> maxOfThree(-2, -8, -1)
-1
```

Extra (Optional) Challenges

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

Challenge 1: palindromeCheck(s)

Write a function `palindromeCheck(s)` that returns True if the string is a palindrome (the same forwards and backwards), and False otherwise.

Sample Function Call:

```
>>> palindromeCheck("madam")
True
>>> palindromeCheck("python")
False
```

Challenge 2: secondLargest(numbers)

Write a function secondLargest(numbers) that returns the **second largest number** in a list of integers.
(Hint: Think about removing the max first, then finding the next max.)

Sample Function Call:

```
>>> secondLargest([10, 20, 30, 40])
30
>>> secondLargest([5, 5, 10])
5
```

Challenge 3: longestIncreasingSubsequence(nums)

Write a function longestIncreasingSubsequence(nums) that finds the length of the longest subsequence where each number is larger than the previous one.
(Hint: This requires nested loops. You do not need to generate the subsequence itself, just return its length.)

Sample Function Calls:

```
>>> longestIncreasingSubsequence([10, 9, 2, 5, 3, 7, 101, 18])
4    # (subsequence: 2, 3, 7, 101)
>>> longestIncreasingSubsequence([0, 1, 0, 3, 2, 3])
4
```

Challenge 4: trapRainWater(heights)

Write a function trapRainWater(heights) that takes a list of non-negative integers where each integer represents an elevation map bar's height. Return the total amount of water that can be trapped after raining.

Sample Function Calls:

```
>>> trapRainWater([0,1,0,2,1,0,1,3,2,1,2,1])
6
>>> trapRainWater([4,2,0,3,2,5])
9
```

To get attendance credit, finish Exercises 1–4.

If you finish early, try the optional challenges!