

CS 115 - Introduction to Programming in Python

Lab 02

Lab Objectives: Strings, Loops, Nested Loops

Instructions: For this assignment, you can use your favorite IDE (Spyder or Jupyter recommended). Upload your solutions as a single .zip file to the Lab01 assignment for your section in Moodle before the end of your lab session. Use the following naming convention:

SS_Lab01_Surname_FirstName.zip where SS is the section number 01, 02, 03, ..., & Surname is your family name, & FirstName is first name. You must attend the lab Zoom session. You must show and explain your solutions to your TA during your lab session and must answer their questions to get your grade by the end of your lab session (the week of Oct 18).

Students who do not attend the lab Zoom session but submit will get 0.

1. Write a program, `Lab02_yourname_Q1.py`, that continually generates and sums up random numbers between 1 and 20 and stops when the sum exceeds 1000. It should stop early if more than 100 numbers have been generated.

Sample Run 1: Sum of 101 random ints in [1..20] is 992	Sample Run 2: Sum of 96 random ints in [1..20] is 1004
Sample Run 3: Sum of 94 random ints in [1..20] is 1005	Sample Run 2: Sum of 89 random ints in [1..20] is 1009

2. Write a program, `Lab02_yourname_Q2.py`, that prompts the user for two positive integers, representing the lower and upper bounds of a range. It finds and displays all integers in that range, inclusive, which are pretty numbers. A pretty number is a number such that the sum of its digits to the power of the number of digits of the number (e.g. it is 3 for 3-digit numbers) is equal to itself. For example, 153 is a pretty number because $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27$. However, 15 is not a Pretty Number because $15 \neq 1^2 + 5^2$ ($15 \neq 26$). **Hint:** Think about type conversions!

Your program should validate input for the given integers. See sample runs below.

Sample Run 1: (User inputs are red) enter first int > 0 [0 to quit]: -10 You must enter a positive int, try again.... enter first int > 0 [0 to quit]: -40 You must enter a positive int, try again.... enter first int > 0 [0 to quit]: 100 enter second int > 0 [0 to quit]: 370 153 is a pretty number 370 is a pretty number	Sample Run 2: (User inputs are red) enter first int > 0 [0 to quit]: 10000 enter second int > 0 [0 to quit]: 300 370 is a pretty number 371 is a pretty number 407 is a pretty number 1634 is a pretty number 8208 is a pretty number 9474 is a pretty number
---	--

3. Write a program, `Lab02_yourname_Q3.py`, that prompts the user to enter a string until an empty is entered. It reports whether the input string is a mirror. A string is a mirror string if its first half is the mirror of its second half, that is, the characters in its first half appear in the same order in the reverse of its second half.

Sample Run: (User inputs are red)

Enter a string (empty to quit): **abccba**

It is a mirror string

Enter a string (empty to quit): **abdcba**

It is a mirror string

Enter a string (empty to quit): **abcde**

It is NOT a mirror string

Enter a string (empty to quit): **abcdeba**

It is NOT a mirror string

Enter a string (empty to quit): **aabcccyccbaa**

It is a mirror string

Enter a string (empty to quit): **abababbababa**

It is a mirror string

Enter a string (empty to quit): **abbbabbba**

It is a mirror string

Enter a string (empty to quit): **abbbbba**

It is a mirror string

enter a string (empty to quit): **abbbabba**

It is NOT a mirror string

Enter a string (empty to quit):

bye!