Practical Questions

CSCA08 Fall 2017 – Week 6

Question # 1

a) my_sum = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 +11 + 12 + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20
Write a loop to do the above summation more efficiently

b) my_list = [9 , 8 , 7 , 6 , 5 , 4 , 3 , 2 , 1]
Write a loop to print out each element in the list on a new line

c) x = 1
y = 2

_____

z = x + y
x = y
y = z

Write a line of code on the blank line that will create a loop that will end when y is greater than 100

d) Write a function that takes in a string, my_string, and a single-character string, my_char, and returns the number of times my_char occurs my_string

e) Write a function that takes in a string and returns a string built from every even index from the original string (first, third, fifth, seventh, etc.)

f) Write a function that takes in a string and returns a copy of the original string with all the repeated characters removed

g) Write a function that takes in a string and returns a string that only contains the characters that were repeated in the entered string (e.g., Input: “112355fjshfsdaslhlsfnmbsdjkgf”, output: “15fjshdl”)

h) Write a function that takes in a string of non-negative integers separated by a space and returns the sum of the integers in the string (e.g., Input: “12 23 56”, output: 91)

i) Write a function that takes in a list of string representation of integers and returns a list of those integers (e.g., Input: [‘12’, ‘45’, ‘78’], output: [12, 45, 78])

j) Write a function that takes in a list of integers and returns a True if the given list of integers is in increasing order. (Each integer is strictly larger than the integer before it. Use a while loop for this question).

k) **(Challenge)** Write a function that returns the sum of all elements within an r_list, where an r_list is defined as a list, where every element of the list is either an integer, or an r_list. (e.g., [1, [2, 3, 4], [5, [6], 7], [8], [], 9] is an r_list).

l) https://open.kattis.com/problems/conundrum

m) https://open.kattis.com/problems/quickbrownfox

n) (Challenge)
https://open.kattis.com/problems/closingtheloop

Discussion Questions

Question # 2

What is one benefit of the way that the while loop determines how many iterations to go through compared to how the for loop decides. What is a downside to the way the while loop determines how many iterations to go through. (This is an important point to understand when using while loops)

Question # 3

In lecture, Brian said that anything you can write with one type of loop, you can write with any other type of loop. Going one direction, this is fairly trivially true: We saw how a for loop is really just a simplified version of a specific type of while loop, so turning any for loop into a while loop should be easy. What about the other way? Can every while loop be trivially turned into a for loop? Is there a simple algorithm to do this?

Question # 4, The “Logic Question” **(Challenge)**

Let’s play a game. You are to be blindfolded, and I will place four rooks (the ‘castle’ piece in chess) on the four corners of a chessboard. Each rook can be placed either right side up, or upside down. Your task is to either turn all of the rooks right side up, or all of them upside down.

At each turn, you can pick 2 of the rooks, and inspect them to feel their orientation (assume that you can tell from feeling them which way they’re oriented), at which point you can choose to flip neither, either or both of the rooks. After each turn, the board will be rotated randomly (but always stopping so that one side is facing you. i.e., it will always go through some number of 90 degree turns).

Once you get all four rooks upside down or right side up, you win, your blindfold will be removed, and you will get a big fat juicy challenge point.

Obviously if you play for long enough, you’ll eventually win by random chance. Your challenge is to devise an algorithm to guarantee a win within a set number of turns.