I

T1204

Data Structures & Algorithms



**Tutorial 01 Python Overview**

1. Complete the following Python function, minmax(data)such that it takes a sequence of one or more numbers, and returns the smallest and largest numbers, in the form of a tuple of length two.

(**NOTE**: Do not use the built-in min or max in implementing your solution.)

|  |
| --- |
| **def** minmax(data):  small = big = data[0]  **for i in range(len(data)):**  **if data[i] < small :**  small = data[i]  **elif data [i] > big:**  big = data[i]    **return (small, big)** |

1. Write a short Python function sum\_of\_squares(n) that takes a positive integer n and returns the sum of the squares of all the positive integers smaller than or equal to n.

Def sum\_of\_squares(n):

Sum = 0

For x in range(0, n+1, 2):

Sq = x\*\*2 # x\*x  
 sum = sum + sq

1. Modify the sum\_of\_squares(n) function written for Qn. 2 such that it now takes a positive integer n and returns the sum of the squares of all the odd positive integers smaller than or equal to n.

(**NOTE**: Do not use the modulus operator in implementing your solution.)

1. What parameters should be sent to the range constructor, to produce a range with values:
   1. 50, 60, 70, 80 range(50, 80+1, 10)
   2. 8, 6, 4, 2, 0, -2, -4, -6, -8 range(8, -8-1, -2)

1. Write a short Python function num\_vowels(text) that counts the number of vowels in a given character string.

Def num\_vowel(text):

Num = 0

Vowels = {‘a’, ‘e’, ‘I’, ‘o’, ‘u’}

Text = text.lower()

For c in text:

If c in vowels:

Num = num + 1

1. Write a Python program that repeatedly reads lines from standard input until an EOFError is raised, and then outputs those lines in reverse order.

Def read\_and\_print\_reverse ():

Inputs = []

While True:

Try:

L = input(“>”)

Inputs = [l] + inputs

Except EoFError:

Break

For I in inputs:

Print(i)

(**NOTE**: A user can indicate end of input by typing ctrl-D).

(**HINTS**: You will need to use the try and except blocks to capture the exception. Use a list to store all the lines then try using the built-in reverse() function to reverse the order.)

AY2019/20 S1 Page 1

I

T1204

Data Structures & Algorithms



Sample Output:

Enter a line (ctrl-D to stop): This is line 1 Enter a line (ctrl-D to stop): This is line 2

Enter a line (ctrl-D to stop): This is line 3

Enter a line (ctrl-D to stop): ^D

This is line 3

This is line 2

This is line 1

Process finished with exit code 0

7. Write a Python function that takes a sequence of numbers and determines if all the numbers are different from each other, i.e. they are distinct.

(**HINT**: The simple solution just checks each number against every other one, but there are more efficient solutions around. For now, just implement the simple solution but make sure you don’t compare a number to itself.)

Def isdistinct(seq):

Distinct = True

For I in range(o, len(seq)):

For I in range(0, len(seq)):

If (i!=j) and (seq[i] == seq [j]):

Distinct = false

Break

Return distinct

***-- End of Tutorial --***

AY2019/20 S1 Page 2