DATA STRUCTURES AND ALGORITHMS

# ASSIGNMENT #2

## Name: Muhammad Roll Number: K17-2421 SEC:B

Please Note: Most codes are user independent. Multiple possible values are taken to test the functions.

# Question 1:

# (Question 6 in book exercise)

Write a short Python function that takes a positive integer n and returns the sum of the squares of all the odd positive integers smaller than n.

Code:

def OddPowerSum(n):

t = 0

for a in range(n):

if a%2 != 0:

t = t + a\*a

return t

print(OddPowerSum(12))

print(OddPowerSum(5))

Output:

286

10

# Question 2:

# (Question 7 in book exercise)

Give a single command that computes the sum from Exercise R-1.6, relying on Python’s comprehension syntax and the built-in sum function.

Code:

oddSum = sum([x\*x for x in range(int(input("Enter Value of n: "))) if x%2 != 0])

print(oddSum)

On input of 5, it returned

10

On input of 12, it returned,

286

# Question 3:

# (Question 8 in book exercise)

Give a single command that computes the sum from Exercise R-1.6, relying on Python’s comprehension syntax and the built-in sum function.

Code:

s = [1,2,3,4,5,6,7,8,9,10]

for n in s: #this loop prints in forward order

print(s[n-1])

for m in range(-len(s),0): #this loop prints in reverse order

print(s[m+len(s)])

|  |  |
| --- | --- |
| Ouput of First **FOR** loop | Output of Second **FOR** loop |
| 1  2  3  4  5  6  7  8  9  10 | 1  2  3  4  5  6  7  8  9  10 |

# Question 4:

# (Question 9 in book exercise)

What parameters should be sent to the range constructor, to produce a range with values 50, 60, 70, 80?

Code:

range( 50 , 90 , 10 )

# Question 5:

# (Question 10 in book exercise)

What parameters should be sent to the range constructor, to produce a range with values 8, 6, 4, 2, 0,−2,−4,−6,−8?

Code:

range( 8, -9, -2 )

# Question 6:

# (Question 11 in book exercise)

Demonstrate how to use Python’s list comprehension syntax to produce the list [1, 2, 4, 8, 16, 32, 64, 128, 256].

Code:

power = [2\*\*x for x in range(9)]

print( power )

Output:

[1, 2, 4, 8, 16, 32, 64, 128, 256]

# Question 7:

# (Question 12 in book exercise)

Python’s random module includes a function choice(data) that returns a random element from a non-empty sequence. The random module includes a more basic function randrange, with parameterization similar to the built-in range function, that return a random choice from the given range. Using only the randrange function, implement your own version of the choice function.

Code:

import random

s = [1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10]

def choice\_re( some\_list ):

return some\_list[random.randrange ( 0 , len(some\_list) ) ]

print(choice\_re( s ) )

**Every time when this code is executed, this function returns a different value**

Output:

7