LAB EXERCISE 5 – ANSWERS TUGBERK GOC - 115200084

1)

# 1) Calculate average of a list  
  
givenList = [1, 2, 3, 4]  
  
average = 0  
  
for i in range(0, 4):  
 average = average + givenList[i]  
  
average = average / len(givenList)  
  
print(average)

2)

# 2) Display name of the day with given specific index  
  
days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']  
  
takenIndex = input("Please, enter the index of the array between 1-7.\n")  
  
print(days[int(takenIndex) - 1])

3)

**Rounding errors**

One of the reasons floating point numbers can be tricky is due to non-obvious differences between binary (how data is stored) and decimal (how we think) numbers. Consider the fraction 1/10. In decimal, this is easily represented as 0.1, and we are used to thinking of 0.1 as an easily representable number. However, in binary, 0.1 is represented by the infinite sequence: 0.00011001100110011… Because of this, when we assign 0.1 to a floating point number, we’ll run into precision problems. In this question, we can see that if we multiply 100 \* 0.1, it is going to be 10.000000190734863281250. In computer, we have to think that how to store data in memory because we can calculate math problems with using stored data. And also, I want to point out another think that if we directly try to calculate 1/10, we find a lot of zeros. If we write 1./10 to teach computer one is float, we find 0.10000000000000000555. These are important points.

REFERENCE:

* https://www.learncpp.com/cpp-tutorial/25-floating-point-numbers/