



Columbia College
Vancouver, Canada

Introduction to Computer Science and Programming 1
CSCI120

Chapter9: Dictionary and Set

Assignment 9

Note: This document has been designed and developed as part of an initiative for creating an OER (Open Education Resource) package for the course CSCI 120 at Columbia College.

Please contact Alireza.davoodi@gmail.com for any comment, modification, and questions.

Terms of use: Please feel free to customize this document as needed

Last Modified: July 2022



If it is a group assignment, please add the information here

# of Students in the Group:		
Student 1	<i>First name, last name</i>	<i>Student-ID</i>
Student 2	<i>First name, last name</i>	<i>Student-ID</i>
Student 3	<i>First name, last name</i>	<i>Student-ID</i>
Student 4	<i>First name, last name</i>	<i>Student-ID</i>

Requirements

- Please use meaningful name for your variables and functions
- Try to reuse your solutions as much as possible.
- For each of the following problem you need to
 - o Define a function
 - o For all test cases you have already written for your algorithm, write a function call inside the main function
- Define all the functions in one file (all in one)
- Define one main function
- Call the functions inside the main function
- If the function you have implemented for a question is big, please try to break down to multiple functions.
- Do not use methods, functions, statements that we have not covered in the previous lectures.



Problem1

- Write a function which has no input parameter. The function asks the user to enter numbers. The user can enter repeated numbers but if the user entered a number which was already entered, the function will provide an error message to the user and ask the user to enter another one. The user can enter numbers as long as s/he has not entered 0. Once a 0 is entered, the function returns the sum of all distinct numbers the user had entered.

Problem2

- Write a function which lets the user enter English words. The user can keep entering English words as long as the user has not entered the word “exit”. Once the user enters “exit” the function will return and print the list of all distinct words starts with English alphabets. Like:
A: Ali, apple, ...
B: Bob, book
... until Z

Problem3

- Design and implement a function which has one input parameter which is a number which is greater than 50, called num. Then the function will create a dictionary whose keys are 2 and 3 and 4 and 5 and 6 and 7 and 8 and 9. Then the function calculates the values for each of the above keys. The value for a key is all the numbers between 2 and input “num” that are divisible by the key. The function eventually will print the result.
- Hint: Use a dictionary whose values are lists.
- Example:
num = 20
2: [2,4,6,8,10,12,14,16,18,20]
3: [3,6,9,12,15,18]
4: [4,8,12,16,20]
5: [5,10,15,20]
6: [6,12,18]
7: [7, 14]
8: [8, 16]
9: [9, 18]



Problem4

- Write a function which receives a list and returns a number. In the list, all numbers have been repeated twice except one number that is repeated once. The function should return the number that is repeated once and return it.

Problem5

- Write a Python function which has a list of points as its input parameter. Each point is represented by a dictionary with 3 keys and each key has a value. The keys of the point dictionary is "x", "y" and "z". Each point represent a position in the 3D coordination system (space). The function should find and return the point in the input list which has the minimum distance to the center of the coordination system. The center of the coordination system is a point whose values for all the keys are 0 (i.e: center = {"x":0, "y":0, "z":0})
- The distance between two points (x1,y1,z1) and (x2, y2, z2) in coordination system is calculated using the following formula:
$$\text{distance} = \text{Sqrt}((x1-x2)**2 + (y1-y2)**2 + (z1-z2)**2)$$

for example:
point1=(2,5,6) and point2 = (1,1,1) then the distance is
$$\text{distance} = \text{sqrt}((2-1)**2 + (5-1)**2 + (6-1)**2) = \text{sqrt}(1+16+25) = \text{sqrt}(42)$$

```
def main():  
    testFindClosestPointToCenter()  
main()
```

```
def testFindClosestPointToCenter():  
    samplePointList = [{"x":1, "y":4, "z":1}, {"x":3, "y":7, "z":10},  
                        {"x":-1, "y":3, "z":4}, {"x":4, "y":4, "z":2},  
                        {"x":7, "y":-4, "z":0}, {"x":10, "y":4, "z":1}]  
    #Complete code here as a test function
```



Problem6

- Write a Python function which receives 3 lists as its input parameters and combines the lists and remove repeated numbers from the combined list and return the combined list. For instance, if the input is [1,2,3,4,2,3] and [3,4,6,7] and [-1,0,23,4] the result is [1,2,3,4,6,7,-1,0,23]
- Note, the order the lists are combined together does not matter.

Good Luck ☺