

INFS1201-Computer Programming

Lab 09

The lab consists of three parts:

- Part 1 consists of fundamental exercises.
- Part 2 consists of four intermediate exercises.
- Part 3 consists of further exercises to think about.
- **Submission of all exercises is expected by the end date of the dropbox.**

Always start your code by the following comments:

```
##  
#firstName lastName - StudentID  
#Week11 lab09  
#Exercise XXX
```

Part I

Exercise 1

In this exercise we will implement a simple restaurant menu management system.

To speed up the process we will start with an initial menu which is a dictionary where the keys are the dishes and the values are their prices:

```
menuDict={'soup':12, 'salad':15,  
          'hummus':10, 'fattoush':13, 'tabbouleh':17,  
          'musakhan':32, 'maqluba':28, 'mansaf':40,  
          'kibbeh':52, 'kebab':45, 'mandi':34, 'biryani':25,  
          'pizza':19, 'burger':22, 'fries':12,  
          'kunefe':15, 'cheesecake':23, 'brownie':21}
```

You have now to implement the following functionalities as functions:

1. view menu: shows the dishes, one per line with the price on the same line
2. add new item: add a new dish with its price, unless the dish is already in the menu
3. update existing item: change the price of a dish, unless the dish is not on the menu!
4. delete item: removes a dish from the menu, unless the dish is not on the menu!
5. Exit: exit the program

The main program is a loop that asks the user which operation he wants to perform until the user wants to exit the program.

Sample run:

```
*****
Welcome to restaurant management system
1: view menu
2: add new item
3: update existing item
4: delete item
0: exit
Which operations do you want to perform (0 to 4)? 2
What item would you like to add? karak
What is the price of karak? 5
*****

Welcome to restaurant management system
1: view menu
2: add new item
3: update existing item
4: delete item
0: exit
Which operations do you want to perform (0 to 4)? 3
What item would you like to update? kara
kara is not in the menu. If you want to add it, select operation 2.
*****

Welcome to restaurant management system
1: view menu
2: add new item
3: update existing item
4: delete item
0: exit
Which operations do you want to perform (0 to 4)? 3
What item would you like to update? karak
What is the new price of karak? 3
*****

Welcome to restaurant management system
1: view menu
2: add new item
3: update existing item
4: delete item
0: exit
Which operations do you want to perform (0 to 4)? 1
soup      12
salad     15
hummus    10
fattoush  13
tabbouleh 17
musakhan  32
maqluba   28
mansaf    40
kibbeh    52
kebab     45
mandi     34
biryani   25
pizza     19
burger    22
```

fries	12	
kunefe	15	
cheesecake		23
brownie	21	
karak	3	

Welcome to restaurant management system

1: view menu
2: add new item
3: update existing item
4: delete item
0: exit

Which operations do you want to perform (0 to 4)? 4

What item would you like to remove? karak

Welcome to restaurant management system

1: view menu
2: add new item
3: update existing item
4: delete item
0: exit

Which operations do you want to perform (0 to 4)? 0

Goodbye!

Exercise 2

Using the same dictionary as before, we want to take the order of customers. This is the same exercise as Week7 exercise 8, but we use dictionaries instead of lists.

Ask the customer which item he would like to order (make sure they are on the menu), how many he wants to order and store this information in a dictionary. When the customer is done display a summary of the order and the total bill.

Sample run:

```
What would you like to order? karak
We do not sell karak. Please choose something else.
What would you like to order? soup
How much soup would you like? 2
What would you like to order? mansaf
How much mansaf would you like? 1
What would you like to order? brownie
How much brownie would you like? 2
What would you like to order? exit
Here is a summary of your order:
- soup : 2
- mansaf : 1
- brownie : 2
Total bill: 106QAR.
```

Exercise 3

Write a function `randomLetter()` that returns randomly a lower case letter.

You should check the `random` module and in particular the `choice` function. You might also be interested in reading about `ord()` and `chr()` functions of python if you would like to use `ascii`.

To check your function run it several times and check that it returns different letters each time.

Exercise 4

MCQ are a very popular form of assessment because they can be automatically graded...and the students can choose an answer at random if they don't know.

The goal of this exercise is to write a function `scoreMCQ(attempt, correct)`, where `attempt` is a string that contains the answers of the student and `correct` is a string with the correct answers.

The function should return the number of correct answers.

Check first that both strings have the same length, otherwise raise an exception.

If both strings are empty the score should be 0 of course.

To test your code, you can copy paste the following assert statements as your main function:

```
assert(scoreMCQ("", "") == 0), "First test failed"
assert(scoreMCQ("TFTF", "TTTT") == 2), "Second test failed"
assert(scoreMCQ("TTTT", "TTTT") == 4), "Third test failed"
assert(scoreMCQ("CCCC", "ABCD") == 1), "Fourth test failed"
print("All tests succesfull")
```

Part II

Exercise 5

Assume you have a dictionary similar to exercises 1 and 2, and you would like to find the most expensive item.

Write a function `maxMenu(menu)` that takes a dictionary as input and returns the tuple corresponding to the most expensive item on the menu with its price. If you call this function in the main menu with the same dictionary given in exercise 1, you should get an output as below:

The most expensive dish is kibbeh and it costs 52 QAR.

Exercise 6

This exercise is again about computing the semester GPA, however we assume now that the grades of the semester are stored in a dictionary.

The dictionary has tuples (courseCode, courseCredit) as keys and the grade as value.

Here is an example of such dictionary:

```
grades={('COMM1020',3):92, ('INFS1201',4):78,  
        ('INFT1201',4):96, ('MATH1030',3):86}
```

You can reuse the function `grade2GP(grade)` from last week.

The output should simply be the semester GPA as below:

Your semester GPA is: 3.4642857142857144

Exercise 7

In week 8 exercise 8, we saw how to generate a list of dice rolls and then get the longest streak of 6.

In this exercise, we just generate dice rolls (based on a number of iterations entered by the user) and

they we display the number of times we get each face (between 1 and 6).

Here are two sample runs:

```
How many times do you want to roll the dice? 10  
Output of dice rolls: [6, 6, 2, 1, 4, 1, 5, 3, 4, 5]  
Frequency of dice rolls: {1: 2, 2: 1, 3: 1, 4: 2, 5: 2, 6: 2}  
>>>  
How many times do you want to roll the dice? 100  
Output of dice rolls: [5, 1, 3, 2, 2, 4, 4, 5, 6, 4, 4, 4, 3, 2, 4, 6, 3,  
5, 5, 5, 1, 2, 1, 4, 4, 4, 4, 2, 6, 3, 2, 6, 4, 4, 3, 5, 4, 2, 3, 3, 5, 4,  
3, 2, 3, 6, 2, 3, 3, 1, 6, 1, 1, 3, 6, 4, 4, 4, 5, 3, 5, 2, 5, 4, 4, 1, 4,  
3, 6, 1, 4, 6, 2, 5, 5, 6, 2, 1, 5, 5, 3, 3, 5, 2, 5, 5, 3, 1, 6, 1, 4, 5,  
6, 3, 4, 2, 1, 4, 6, 2]  
Frequency of dice rolls: {1: 12, 2: 15, 3: 18, 4: 24, 5: 18, 6: 13}
```

Exercise 8

In Qatar, you can get notes of 1, 5, 10, 50, 100, 200 or 500 QAR. The goal of this exercise is to write a `makeChange(amount)` function, that takes an amount as input and returns a dictionary that contains the smallest number of notes that can make this amount. The dictionary should only contain the notes that are used to make the amount, we don't want to see the other notes that are not used.

Here is a sample run:

```
Enter an amount: 232  
{200: 1, 10: 3, 1: 2}
```

Part III

Exercise 9

In exercise 4 we saw how to automatically grade MCQs.

A nice instructor realized that his students were performing poorly at an MCQ... so he decides to manipulate the answer key in a way to guarantee the best average grade for his students.

To simplify, we will assume in this problem that MCQs are only T/F questions, and we have a list of students' attempts. The goal is to find the "correct" answer that will lead to the best average for the class.

To clarify, you are given the list of students' attempts and your goal is to maximize their score by manipulating the answer key.

As a simple example, if you have four students and they have only one question to answer, and their answers are ['T', 'T', 'T', 'F'] then you set the answer key to be 'T' as this will lead to an average score of 0.75 because the scores would be [1,1,1,0]. If you choose 'F' as answer key the average would be only 0.25.

If the same students have two questions to answer and they answer ['TF', 'TT', 'TF', 'FF'] then you set the answer key to be 'TF' as this will lead to an average score of 1.5 because the scores would be [2,1,2,1]. If you choose another answer key the average score would always be lower than 1.5. For example, if you choose 'TT', then the scores would be [1,2,1,0] which gives an average of 1.

Now an actual sample run. If we have the following list of answers by 6 students:

```
attempts=['TFTTFFFTFT', 'FFFTTFFFTFT', 'TFTFFFFFTFT', 'TFTFFFFFTFT', 'TFTTTTTTFT',  
, 'TTTTTTTTTT']
```

Then the output of the program should be the following:

```
Arranged 'correct' answers: TFTTFFFTFT  
Arranged scores: [10, 8, 8, 8, 8, 6]  
Best average: 8.0
```


Exercise 10

This exercise builds on exercise 3. We want to generate a number of letters at random, then find the frequency of the letters generated and the most frequent letter (the mode).

This exercise is similar to exercise 7, except that we don't start from a list, we generate the frequencies on the fly and we don't know for sure the output (not all letters appear always). The output should be ordered in alphabetical order of the keys. There might be more than one mode, you need to display them all.

See below sample runs:

```
How many letters do you want to draw at random? 10
```

```
Frequency of letters: {'a': 2, 'b': 1, 'd': 1, 'e': 2, 'k': 1, 'o': 1, 'u': 2}
```

```
Most frequent letter(s): a e u
```

```
>>>
```

```
How many letters do you want to draw at random? 100
```

```
Frequency of letters: {'a': 4, 'b': 4, 'c': 2, 'd': 5, 'e': 1, 'f': 4, 'g': 8, 'h': 4, 'i': 5, 'j': 2, 'k': 3, 'l': 6, 'm': 6, 'n': 4, 'o': 4, 'p': 4, 'q': 3, 'r': 3, 's': 5, 't': 6, 'u': 4, 'v': 2, 'w': 3, 'x': 3, 'y': 1, 'z': 4}
```

```
Most frequent letter(s): g
```