Dictionaries

Dictionaries

Reading Challenge 1. Describe (in words, written on paper or in a text file) what kind of dictionary the following code snippets define. They are *all* valid!

```
phonebook = dict()
phonebook["John␣Paul"] = 07924316548
phonebook["Fionnuala␣Johnson"] = 07439532325
```

```
shopping_list = {}
shopping_list["tinned␣food"] = ["tomatoes", "beans", "tuna"]
shopping_list["vegetables"] = ["yam", "potato", "tomatoes"]
```

```
doubled_numbers = {2:4, 3:6, 8:16, 25:50}
```

```
hotel = {}
floor_1 = {101: "Mr.␣Akeju", 103: "Dr.␣Praise␣Adeimo"}
floor_2 = {202: "Dr.␣Kitty␣Meeks"}
floor_3 = {} # This floor is empty!

hotel["floor␣1"] = floor_1
hotel["floor␣2"] = floor_2
hotel["floor␣3"] = floor_3

hotel["floor␣4"] = {401: "Dr.␣Tim␣Storer", 402: "Joe␣
    Gallagher", 405: "Nicola␣Sturgeon"}
```

Problem 2. Write a dictionary which takes strings as their keys from user input (using the "input" function), and makes the value the number of times the user has inputted that thing. So, if the user inputs 'a', then 'b', and then ''', the dictionary should be `{'a': 2, 'b': 1}`.

Problem 3. Write a function that prints the keys and values of a dictionary prettily.

Problem 4. Write a function of your choosing which takes and returns a number (such as `def double(x): return x*2`).

Write another function which checks to see whether the input has been seen before. If it has not, use the function to calculate the result and *save the calculation in a dictionary*, where the key is the input and the value is the result. If it has been seen already, look up the result in the dictionary. For the double function given as an example, you should be able to build the relevant dictionary from the reading problems at the beginning of this section.

Once you have solved this problem, consider other functions you could have written. Are there situations where this technique could make your programs more elegant or quicker? What are those situations? Discuss with a tutor if you can get their attention.

This technique is usually referred to as Dynamic Programming.

Problem 5. Using a dictionary, where keys are strings and values are lists:

1. Implement a directed graph
2. Write a function which randomly traverses the graph for `n` nodes, where `n` is an argument to your function.

Hint: the keys are the names of the nodes, and the edges are adjacency lists.

Problem 6. A Markov chain is a graph with probabilities assigned to all of their edges. When Markov chains are traversed, we move from one node to an adjacent node according to the probability on the edge.

Before continuing, it would be wise to familiarize yourself with Markov Chains at this link.

Using a dictionary where the keys are strings, as in problem 5, but the values are *pairs of adjacent nodes and their relevant probabilities*:

1. Implement a Markov Chain
2. Write a function which randomly traverses the Markov Chain.