sOFTWARE 1 PRACTICAL

## Dictionaries

Week 5 – Practical 8

### **Exrcise 1:**

Consider a set of dictionaries representing our online business. One contains the stock and the other one contains the price of each item in pence. An example is given below:

stock = {

"banana": 6,

"apple": 0,

"orange": 32,

"pear": 15

}

prices = {

"banana": 40,

"apple": 20,

"orange": 15,

"pear": 30

}

basket\_1 = {

"banana": 4,

"pear": 3

}

basket\_2 = {

"apple": 1,

"pear": 3

}

1. Write a function basket\_price(basket, stock, prices) which compute the price of a given basket. If the basket contains more items than it is available in stock the function should return -1. Given the example above, the function should return 250 for basket\_1 and -1 for basket\_2 (not enough apples).
2. Write a function checkout(basket, stock, prices) which returns the price of the basket, and update the stock, that is remove the items in the basket from the stock. If the basket contains more items than it is available in stock the function the operation must be cancelled (that is no item is removed from the stock) and the function should return -1. Given the example above, the function should return 250 for basket\_1 and stock should have only 2 bananas and 12 pears left. For basket\_2 the stock should remain unchanged and the function should return -1.
3. Write a function add\_stock(items, stock) where items is a dictionary where keys are fruits, and the values are the number of fruits added to the stock. The parameter stock is the stock to be updated. The function does not return any value. For example, after calling add\_stock({‘apple’:5, ‘kiwi’:10}, stock), the new stock should be:

stock = {

"banana": 6,

**"apple": 5**,

"orange": 32,

"pear": 15

**"kiwi": 10**,

}

1. Write a function price\_increase(increase, prices) where prices is a dictionary where keys are items and values are price in pence, and increase is a float representing a price increase. For example, an increase of 5% is represented by the float 0.05. The function should modify the dictionary prices so the prices are adjusted accordingly. Note the prices should be rounded to the nearest integer. The function returns the updated prices dictionary. For example, after an increase of 20%, prices should be:

prices = {

"banana": 48,

"apple": 24,

"orange": 18,

"pear": 36

}

### **Exercise** 2: *Text on 9 keys – T9 (from seminar)*

T9, which stands for **Text on 9 keys**, is a predictive text technology for mobile phones, specifically those that contain a 3x4 numeric keypad as shown in Figure 1. T9’s objective is to make it easier to type text messages. It allows words to be entered by a single keypress for each letter, as opposed to the multi-tap approach used in conventional mobile phone text entry, in which several letters are associated with each key, and selecting one letter often requires multiple keypresses.



Figure 1: Example of a numeric keypad used for T9

1. In later versions of T9, the order of the words presented adapts to the usage pattern. For instance, in English, 4663 matches "good", "home", "gone", "hood", etc. Such combinations are known as textonyms; e.g., "home" is referred to as a textonym of "good". When the user uses "home" more often than "good", eventually the two words will switch position. Information about common word combinations can also be learned from the user and stored for future predictions.

t9\_keypad = {’2’:[’a’,’b’,’c’], ’3’:[’d’,’e’,’f’],

’4’:[’g’,’h’,’i’], ’5’:[’j’,’k’,’l’],

’6’:[’m’,’n’,’o’], ’7’:[’p’,’q’,’r’,’s’],

’8’:[’t’,’u’,’v’], ’9’:[’w’,’x’,’y’,’z’]}

Write a function extract\_textonyms(vocabulary, keypad) that takes list of string named vocabulary as a parameter. vocabulary contains a series of words that represents the know vocabulary for a user. The second parameter is a keypad representation named keypad similar to the one shown below. The function should return a dictionary (user\_dict) where the keys are digits pattern, and the values list of sub-lists. The sub-list contains only two elements, the first element is a word from the vocabulary with that digits pattern, and the second element is the frequency of the word. For example:

user\_dict={'4663':[['home',5],['good',8],['hood',1]],

'2':[['a',50]],...

}

1. Write a function get\_words(digits, user\_dict) that takes a string of digits such as ‘4663’, the user vocabulary and returns the list of words corresponding to that pattern ordered by decreasing frequencies (in this case [‘good’, ‘home’, ‘hood’,…]).

**Hint**: you may want to search on how to use the method sort from the list type.

1. Write a function add\_words(text, user\_dict, keypad) which update the user\_dict data structure by adding the words contains in text (a list of string) to the user\_dict. If the word already exists in the user\_dict, its occurrence should be increased by the number of times it appears in the text. The function should return the updated user\_dict.