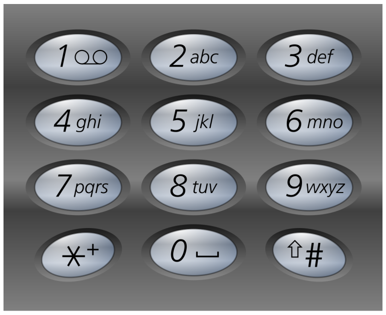
Presentation exercises – Week 11

COMP 202, Winter 2022

Presentation date: Friday, April 1st

Presentation exercise 1

For this question you will write a couple of functions that will allow you to translate a string made of numbers and space characters into a text message (i.e. a string made of letters of the English alphabet and space characters). The idea comes from old telephones keypads like the following one:



To type hello in your phone, you would have pressed the keys in the following order: 44-33-555-555-666. To denote the pauses you have to take between entering each series of numbers, we will be using space characters. So, the string `'44 33 555 555 666'` would be translated into `'hello'`. To represent the mapping between a sequence of digits and the letter that it represents we will be using a dictionary. The dictionary will be mapping integers between 0 and 9 to lists of strings. The position of the string in the letter denotes how many times the integer should be repeated for it to represent such string. So, for instance the dictionary `{2 : ['a', 'b', 'c'], 7 : ['p', 'q', 'r', 's']}` tells us that to represent the string `'a'` we need the integer 2 repeated once (because `'a'` is the first string in the list to which the key `2` maps). On the other hand, to represent the string `'r'` we need to use the integer 7 repeated three times, since `'r'` is the third string in the list to which `7` maps.

Write a function (including docstring) called **`same_chars`** that takes a string as input and returns true if the string contains at least one character and all the characters within the strings are equal.

For example:

```
>>> same_chars('555')
True
>>> same_chars('aaaaa')
True
>>> same_chars('COMP')
False
>>> same_chars('aaAa')
False
```

Then, write a function (including docstring) called **`get_txt_msg`** that takes as input a string containing digits and space characters, as well as a dictionary mapping integers to list of strings. The function returns a string representing the translation obtained from the input string using the input dictionary.

Note that the function should raise a `ValueError` if the input string is not valid. The input string should be consider invalid if:

- It begins or ends with space characters
- It contains two or more consecutive space characters
- It contains characters other than digits or space characters
- It contains sequences built using different different digits
- It contains sequences of digits which cannot be translated using the input dictionary.

For example:

```
>>> map_one = {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'], 0: [' ']}
>>> code = '222 666 3 444 66 4 0 444 7777 0 333 88 66'
>>> msg = get_txt_msg(code, map_one)
>>> msg
'coding is fun'

>>> map_two = {0 : ['a ', 'the ', 'an '], 1 : ['lion ', 'bear ', 'cat '], \
               2 : ['is ', 'are ', 'has '], 3 : ['over ', 'under ', 'below ', 'in '], \
               4 : ['table ', 'forest ', 'tree ', 'cave ']}
>>> s = '00 1 2 33 00 444'
>>> get_txt_msg(s, map_two)
'the lion is under the tree '

>>> d = {2 : ['A', 'B', 'c'], 7 : ['p', 'q', 'r', 'S']}
>>> get_txt_msg('222 2 777', d)
'cAr'

>>> get_txt_msg('227', d)
Traceback (most recent call last):
ValueError: Invalid input string

>>> get_txt_msg('11 2 5', d)
Traceback (most recent call last):
ValueError: Invalid input string

>>> get_txt_msg('77 2222', d)
Traceback (most recent call last):
ValueError: Invalid input string
```

Presentation exercise 2

Write a function **`get_key`** that takes two inputs: a dictionary with integers as values, and a function object. Evaluate the function on the list of values from the dictionary to obtain a value `v`. Return the key of the dictionary that maps to `v`. If there's more than one key, break ties at random. If there's no such value in the dictionary, return **`None`**.

Examples,

```
>>> animals = {'tiger' : 5, 'bat' : 3, 'cow' : 3, 'bear' : 4}
>>> get_key(animals, max)
'tiger'
```

```
>>> random.seed(5)
>>> get_key(animals, min)
'cow'
>>> print(get_key(animals, avg))
None
```

where the function `avg` is defined as follows:

```
def avg(x):
    return sum(x) / len(x)
```

Show examples with at least three other functions you could pass as function objects to `get_key`.

Presentation exercise 3

In this question we will calculate and plot statistics about certain novels. First, navigate to Project Gutenberg (`https://www.gutenberg.org`), a site that offers downloads of out-of-copyright books in .txt format. Choose 3 books from the site (you can consult the top downloaded ones here), and make sure to download in 'Plain Text UTF-8' format. (You can download by saving once you are on the .txt page, or, if in Safari, you can click the 'Plain Text UTF-8' link while holding the option button.)

Now that you have downloaded the three text files, you must analyze the text within. Write the following functions:

- `get_word_count(text)`: Given a string, return a dictionary with the keys being the words and the values being the number of times the word appears in the string.
- `remove_common_words(word_dict)`: Given a dictionary of words (keys) and the number of times they appear (values), remove all key-value pairs from the dictionary where the key is a common word. You should make up a list of at least 15 common words (e.g., 'a', 'the', 'of', 'she', 'he', 'they', etc.).
- `get_word_sequences(words)`: Given a string, return a dictionary where the keys are tuples of length two, with the first element in the tuple being a word in the string, and the second element in the tuple being the word immediately following the first word in the string. The value for each key should be the number of times that sequence of words appears in the string.
  Example:

  ```
  >>> get_word_sequences(["python is good python is great"])
  {('python', 'is'): 2, ('is', 'good'): 1, ('good', 'python'): 1, ('is', 'great'): 1}
  ```

Then, call your functions for each text file that you downloaded, loading in each file as appropriate. Take the 10 most commonly occurring words and 10 most commonly occurring word sequences, and make a bar graph for each, per file. You should thus have 6 plots in total. Show the plots in your presentation and discuss what the differences per text file say about the novels.

Presentation exercise 4

In this question, we ask you to consult the following website which contains a list of Python datasets on a wide variety of topics:

`https://corgis-edu.github.io/corgis/python/`

Choose one of the topics that interests you the most, and download the files provided on the topic webpage. The Usage section shows how to import the data into your code. The Documentation section shows what the function shown in the Usage section does. Typically, it returns a list of dictionaries containing the data of the dataset. Your job is to explore the list of dictionaries. What are the keys and what are the values? Show this in your presentation.

Then, plot three interesting characteristics of the data using Matplotlib. Each plot should be a different type (line plot, bar plot, etc.). Discuss why you chose these characteristics in your presentation.

Presentation exercise 5

This question is a bit longer than others, so you must work on it as a pair and present it together. Make sure to evenly contribute to both the code and the presentation. The presentation should take between 8-10 minutes (approx. 5 minutes per person). Do **not** work on the question by yourself.

Blackjack is a card game played by two players with a regular deck of fifty-two playing cards. Each player is given one card face down and one card face up, and they are asked in turn if they would like another card (to 'hit') or not (to 'stand'). When both players decide to 'stand', their face-down card is revealed, and the player who has a higher card total wins. The value of a card is the number (if between 2 and 10), or 10 (if Jack, Queen or King), or 1 or 11 (player's choice, if Ace).

If a player 'hits' and their total goes above 21, then they 'bust' and automatically lose.

Count an Ace as a 1, unless the total sums to exactly 21.

Write a function `play_blackjack`. The function takes two integer inputs: `hit_limit1` and `hit_limit2`. It simulates a game of Blackjack between two players and returns an integer corresponding to which player wins the game (player 0 or player 1). To simulate a game, first create a list of playing cards. You can choose how to represent the cards (integers, dictionaries, objects of a `Card` class, etc.). Then deal the cards to the players. The players will continue to 'hit' until their total card value is above their `hit_limit` (each player has their own hit limit).

Then, write a function `simulate_games` that takes an integer input `n` and two integer inputs for the hit limits. The function simulates `n` Blackjack games with the given hit limits. It returns a dictionary where the keys are the players and values are how many times each player won.

Finally, write a function `graph_results`. It takes no inputs and returns nothing. It calls the above function several times, each with the same, large `n` (greater than 10000), but with different hit limits each time. The function should then graph the results of the wins per player based on the different hit limits using Matplotlib. Think about the best way to display this information (what type of plot, etc.). Discuss your choice of plot in your presentation, and discuss also the results you obtained from the simulation.