**Lab 5 – Lists (Part II)**

These exercises are to be completed using Notepad++ and Anaconda Prompt.

**Q4: Spelling Check [ \*\* ]**

You are given a file called q4.py. Inside the file, you’re given a list called COMMON\_WORDS that contains 5000 commonly used English words. Define a function called check\_spelling() that checks for misspellings. The function takes in a string that represents a piece of text. It returns a list of words from the text that are possibly misspelled, i.e., a list of words that are not found in the 5000 commonly used English words.

For example, check\_spelling("I studdy at Singapore Managment Univercity") should return ["studdy", "Managment", "Univercity"].

**Q5: Tax Calculation [ \*\*\* ]**

Let us now revisit the tax calculation task. Recall that we have the following tax rates:

|  |  |  |
| --- | --- | --- |
| **Chargeable Income** | **Income Tax Rate (%)** | **Gross Tax Payable ($)** |
| First $20,000  Next $10,000 | 0  2 | 0  200 |
| First $30,000  Next $10,000 | -  3.50 | 200  350 |
| First $40,000  Next $40,000 | -  7 | 550  2,800 |
| First $80,000  Next $40,000 | -  11.5 | 3,350  4,600 |
| First $120,000  Next $40,000 | -  15 | 7,950  6,000 |
| First $160,000  Next $40,000 | -  18 | 13,950  7,200 |
| First $200,000  Next $40,000 | -  19 | 21,150  7,600 |
| First $240,000  Next $40,000 | -  19.5 | 28,750  7,800 |
| First $280,000  Next $40,000 | -  20 | 36,550  8,000 |
| First $320,000  In excess of $320,000 | -  22 | 44,550 |

The information above can be stored inside a list of tuples as shown below:

TAX\_INFO = [

(20000, 0, 0.02),

(30000, 200, 0.035),

(40000, 550, 0.07),

(80000, 3350, 0.115),

(120000, 7950, 0.15),

(160000, 13950, 0.18),

(200000, 21150, 0.19),

(240000, 28750, 0.195),

(280000, 36550, 0.2),

(320000, 44550, 0.22)

]

You can see that each element of this list is a tuple with three values: (1) an amount of chargeable income, (2) the payable tax for that amount, and (3) the tax rate for extra income above that amount. For example, the tuple (30000, 200, 0.035) indicates that for the first $30,000 of chargeable income, $200 is charged as tax, and for any additional income above $30,000 (and below the next threshold of $40,000), a tax rate of 3.5% is applied.

The list above is given in q5.py. Implement a function called calculate\_tax() inside q5.py that takes in a number representing the taxable income of a person. The function returns the amount of tax that person has to pay. Some test cases have been given in q5.py.

**Q6: More on Lists [ \*\*\* ]**

In all the questions below, you can assume that the lists passed to the functions (i.e., the parameters) do not contain any duplicate elements. You can also assume that the lists passed to the functions are not empty.

1. Define a function called get\_all\_combinations(). The function takes in two lists. The first list is called str\_list and contains a sequence of strings. The second list is called num\_list and contains a sequence of numbers. The two lists may have different lengths. The function returns a list of tuples, where each tuple is a combination of an element from str\_list and an element from num\_list. The returned list should contain all possible combinations.

For example, get\_all\_combinations(["a", "b"], [1, 2, 3]) should return [("a", 1), ("a", 2), ("a", 3), ("b", 1), ("b", 2), ("b", 3)].

1. Define a function called get\_larger\_numbers(). The function takes in two lists of numbers, num\_list1 and num\_list2. The function returns all the numbers in num\_list1 that are larger than all the numbers in num\_list2.

For example, if num\_list1 is [4, 6, 10] and num\_list2 is [1, 3, 5], then the function should return [6, 10]. This is because 4 is not larger than all the numbers in num\_list2, but 6 and 10 are both larger than all the numbers in num\_list2.

1. Define a function called get\_non\_common\_strings(). The function takes in two lists of strings, str\_list1 and str\_list2. The function returns a list of strings that can be found in either str\_list1 or str\_list2, BUT NOT in both.

For example, if str\_list1 is ["a", "b", "c", "d"], str\_list2 is ["b", "d", "e", "f"], then this function returns ["a", "c", "e", "f"].