BAN 601 Homework 2

Edit your submission in this word document, attaching the screenshots of the codes used for each question. Include narrative descriptions, outputs screenshot, or short answers when requested.

**Task 1**

Given a list (x=[56,32,21,8,56,21,35]), write Python code to calculate the index of the duplicated number from right to left

**Task 2**

Given a list x (x=[56,32,8,21,35]), please write python code to calculate the third largest number in this list

**Task 3**

**Given a list of integers, write a program to find the three largest numbers in the list without using built-in sorting functions. Return the numbers in a list. If there are fewer than three distinct numbers in the list, return a message stating 'Not enough distinct numbers'.**

* Input: [5, 23, 8, 23, 12, 5, 8]
* Output: [23, 12, 8]
* Input: [7, 5, 5]
* Output: 'Not enough distinct numbers'

**Task 4**

Sum all values in the list objects only if they are of type tuple.

lst=[[1,2,3],(4,5,6),2,"2"]

**Task 5:**

Generate random indexes using uniform distribution and count the number of random tries before successfully accessing an element in the list

lst = [2,3,4]

Hint: you may use the following:

while condition:

try:

…

except:

…

**Task 6**

We have the following key value pairs stored in two lists regarding our products

Keys=[‘length’,’width’,’depth’,’weight’,’reliability’]

Values=[6,7,8,9,’good’]

Inquire about the key 'depth'. If it doesn't exist, return 'feature not found'

**Task 7**

Combine the values of common keys by multiplying them

Suppose

d1 = {'a': 100, 'b': 200, 'c':300}

d2 = {'a': 3, 'b': 2, 'd':4}

The expected output should be

{'a': 300, 'b': 400,'c': 300, 'd': 4,}

**Task 8**

We store cargo in some location identified with 3 different numbers (x, y, and height) in the format of a tuple. For the example, the cargo variable looks like this:

Cargo={(30,23,12): {"a":12,"b":13,"c":15},(36,3,122):{"b":15,"d":3,"e":5}, (36,3,12):{"a":25,"c":30,"f":5}}

It says that at location (30,23,12), there are 12 “a” cargo, 13 “b” cargo, and 15 “c” cargo. Note that the value of the outer dictionary is itself a dictionary.

We also have a price chart stored in the format of a dictionary, showing the price of different types of cargo. For example:

price={"a":100,"b":200,"c":300,"d":450,"e":700,"f":650}

Which says that the price of “a” is $100.

Calculate the total value of cargo stored at the x-coordinate of 36

**Task 9**

Please write code to identify the last recurring character in a string, its positions, and the number of times it recurs. If no recurring characters are found, return a message stating so. Also, if there are multiple characters that recur for the maximum number of times, return all of them with their positions (input = "XDZYTUVZWV", "XDZZTUVZWT", "XDYUWV")

For example:

* input = "XDZYTUVZWV", output = {'Z': [2, 7], 'V': [6, 9], "times\_recurring": 2}
* input = "XDZZTUVZWT", output = {"Z": [2, 3, 7], "T": [4, 9], "times\_recurring": 3}
* input = "XDYUWV", output = "No recurring characters found."

**Task 10**

Design a function that takes in two strings ('what', 'where'). For each string, it should remove characters that appear in the other string. After the removal, it should concatenate the first 50% of the modified first string with the last 50% of the modified second string. If either of the modified strings has less than 2 characters, return a message indicating which string is too short.

For example:

string\_1 = "programming", string\_2 = "language", the function should return `prorue`.

string\_1 = "hello", string\_2 = "world", the function should return `howorld`.

string\_1 = "a", string\_2 = "b", the function should return "String 1 is too short after modifications."

Hint: Consider using string slicing for effective concatenation.