PYTHON WEEK 2 ASSIGNMENT

DATA STRUCTURES

- Please name your py file as the topic of each question and submit them separately.
    - For example, for the first question: "rotate_list.py".
- Note that value between these <> symbols will be dynamic.

1. rotated_list.py

Input:

First line of input says how many index you will slip left the whole data in the list and put the data that will have less than zero index to the rightmost of the list. Second line of the data shows list that you will rotate. Take the list elements from the user input until the user enters "q". Be careful that you print the output like below(in one line).

Output:

"The new order of list is: <list_elements>"

Example:

Input:

2

2 3 4 5 6

Output:

"The new order of list is: 4 5 6 2 3"

Explanation :

2 3 4 5 6 when rotated by 2 elements, it becomes 4 5 6 2 3.

Example:

Input:

3

3 5 7 9 11 13 15 17 19 21

Output:

"The new order of list is: 9 11 13 15 17 19 21 3 5 7"

Explanation :

3 5 7 9 11 13 15 17 19 21

when rotated by 3 elements, it becomes

9 11 13 15 17 19 21 3 5 7

2. x_th_smallest.py

Given a list **list[]** and a number X where X is smaller than the size of list, the task is to find the **Xth smallest** element in the given list. It is given that all list elements are distinct.

Input:

- **The program should ask the user**
- **"Enter the list items: (to stop the program please enter 'q')"**
- **Take inputs for the list items until user enters 'q'.**
- **After that, ask user to take input for :**
- **"Which number do you want to know as Xth smallest number in the list?"**
- **Take input for the number**

Output:

Corresponding to each test case, print the **Xth** smallest element as

"<X>th smallest element in the given list is <number>."

1.Example:

Input:

8 11 5 4 30 17	(These are the list element)
3	(This is the x)

Output:

3rd smallest element in the given list is 8.

2.Example:

Input:

8 11 5 4 30 17	(These are the list element)
4	(This is the x)

Output:

"4th smallest element in the given list is 11."

3. max_distance.py

There is a list with repeated elements, the duty is to find the maximum distance between two occurrences of maximum repeated element in the list and print

"Max Distance between the maximum repeated number (<maximum_repeated_number>) is: <distance>"

Input:

Line of the data shows list that you will find the max distance. Take the list elements from the input of user until the user enters "q".

Output:

For each test case in new line print the maximum distance between two occurrences of an element

Example:

Input

3 5 1 1 2 1 3 4 5 2 4 3 1 7 6

Output

"Max Distance between the maximum repeated number (1) is: 10"

Explanation

list = [3, 5, 1, 1, 2, 1, 3, 4, 5, 2, 4, 3, 1, 7, 6]

Maximum repeated element is 1. There is four times for 1 and

Maximum distance between them is 10.

12(index of last "1") - 2(index of first "1") = 10

4. count_same.py

Find the same character in the data and make a dictionary that shows how many times it occurred in the data for each different character. Take just characters, not punctuations. Be sure that you take lowercase and uppercase of a character as a same character ("a" and "A" are the same character.) and store each character as uppercase.

data = "There Are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some Form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a Passage of Lorem Ipsum, you Need to be sure there isn't anything embarrassing Hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model Sentence structures, to Generate Lorem Ipsum which looKs reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc."

Example:

Input

data(in the question)

Output

word_occurence= {

"K" : 4,

"A": 42,

.

.

.

.

.

}

5. intervals.py

Create a dictionary of keys for ten character of English alphabet(from "a" to "j") and each key has as value a tuple from 0-10 for "a" , 11-20 for "b" and so on respectively.

word_intervals = {

"b" : (11, 12, 13, 14, 15, 16, 17, 18, 19, 20),

"d": (31, 32, 33, 34, 35, 36, 37, 38, 39, 40),

.

.

.

.

.

}

6. alphabetically.py

Sort the dictionary data values below.

Input

```
data =
{
'a": [5, 3, 9, 0, 2, 3, 1],
 'b': [4, 7, 5, 1, 2],
 'c': [8,1, 3, 2, 4]
}
```

7. tuples.py

# 7.1 - Convert a tuple to a string and print.

Input:

tup1 = ('n', 'e', 't', 'h', 'e', 'r', 'l', 'a', 'n', 'd', 's')

Output:

"netherlands"

7.2 - unzip a list of tuples into individual lists and print.

Data:

tup2 = [(3,6), (5,8), (7,4)]

Output:

[(3, 5, 7), (6, 8, 4)]

7.3 - convert a list of tuples into a dictionary and print.

Data:

tup3 = [("a", 1), ("a", 2), ("a", 3), ("b", 1), ("b", 2), ("c", 1)]

Output:

{'b': [1, 2], 'c': [1], 'a': [1, 2, 3]}

8. fibonacci.py

Firstly, search for what Fibonacci number is on the internet. Then, write a code that takes an input from user as a number in order to find Fibonacci numbers until this number and store it as a tuple. Then print these numbers.

9. sets.py

9.1 Create an intersection of sets and print it.

Data:

data1 = (["green", "blue"])
data2 = (["blue", "yellow"])

Output:

{'blue'}

9.2 Create set difference and print them.
Data:

data1 = (["apple", "mango"])
data2 = (["mango", "orange"])
Output:

{'apple'}
{'orange'}

9.3 Find maximum and the minimum value in a set and print them. Do not use built-in functions (min() and max()). Find by iteration(loop) and conditionals.
Data:

data1 = ([5, 10, 3, 15, 2, 20])

Output:

20
2