CSM0120
Programming for Scientists

Practical 02
Input, Lists, Dictionaries and Loops

Tasks

Yasir Saleem

yss1@aber.ac.uk

13th October 2023

Task 1:

Write a program that prompts the user to input a number, then asks them to input a second number, and finally prints the sum and product of the two numbers.

Task 2:

Rewrite your times table exercises from Practical 1 using a for-loop instead of a while loop.

i. Print out the two-times-table so that your code produces lines such as

2 times 1 is 2
2 times 2 is 4
…
2 times 12 is 24

ii. Print out all the times tables from 1 to 12 using an inner for loop and an outer for loop. It should look like:

1 times 1 is 1
1 times 2 is 2 …
2 times 1 is 2 …
12 times 12 is 144

iii. Print the times tables in (ii) in reverse, counting down from 12 to 1. It should look like:

12 times 12 is 144
12 times 1 is 12
11 times 12 is 132 …
1 times 1 is 1

Task 3:

Create a variable `towns` and initialize it as an empty list. Prompt the user 5 times (using a `for` loop) to input the name of a town in Wales. Append each town that the user has input into the `towns` list. Sort the list and use another `for` loop to print out the towns in alphabetically sorted order.

Task 4:

Write a program using `for` loop that will create a random DNA string of 100 characters using the letters A, C, G and T only.

You can pick a single random character by using the following code:

```
import random
x = random.choice(["A","C","G","T"])
```

Task 5:

Create a program that prompts the user to type in a sentence as input. Then create a dictionary to store all the unique characters excluding whitespaces. Finally, print out only those characters in the dictionary of unique characters that are not vowels (a, e, i, o, u).

Use a for-loop to go through each character in turn. For example:

```
for character in sentence:
    # do required stuff
```

Hint: **`c not in mylist`** will test if a character **`c`** is not in the list **`mylist`**.

Task 6 (More Challenging):

The function `chr()` generates the character that corresponds to an ASCII numeric value. For instance, "A" is 65, "B" is 66, "Z" is 90 and so on. The following is a program that will generate a random list of 10 letters (`rlist`).

```
import random
letters = []
for x in range(65,91):
    letters.append(chr(x))
rlist = random.sample(letters, 10)
```

Write code to accomplish the following four steps:

i. Print the list of random letters `rlist` to the user and prompt the user to input a word that they can make from the letters in `rlist` (for this task, the word does not need to be a proper. Just some characters that match the random letters in `rlist` should be okay if you could not think of any word).

ii. Get the input word and store it in a variable `input_word`. Check that the `input_word` has at least 3 characters. Print a message "Word is too short" and ask the user to give another input if `input_word` is less than 3 characters. Repeat the same process until a word with a valid length of 3 characters is given as input.

iii. Once the length of `input_word` is valid (i.e., at least 3 characters), check whether `input_word` contains all the characters in `rlist`. Initialize a variable `is_word_acceptable` and set it to `False` as its initial value. Use a `for` loop to go through the `input_word` that the user has input and check that it must contain all characters in `rlist`. If all the characters in `rlist` have been included in the `input_word` then set the `is_word_acceptable` variable to be `True`. (You would be needing to set `is_word_acceptable` to `True` somewhere before the `for` loop, otherwise it will never go to the condition of "Congratulations!" statement mentioned in step (iv)).

iv. At the end, if the `is_word_acceptable` is `True`, print a message "Congratulations!". Otherwise, print a message "Error. Please enter the word from above letters" and go back to step (ii) and restart the same process.

(Hint: use a single `while` loop to include both conditions in (ii) and (iii))