Practice Week 02

Exercise 1: Loops

Write code for each of the following examples. In the case of any **for** loops, also implement the same coding task with a **while** loop.

1. Use a **for** loop to print the numbers 1 to 10 to the screen. Now write this using a while loop.
2. Use a **while** loop to print the temperature along with the statement "The Weather is Fine!" for as long as the temperature is between 20 and 30 degrees. To simulate the weather, you can use the `randint()` function from the *random* module, with the lower bound set to 16 and the upper bound to
3. Why can't you implement this as a for loop?
4. Use a **for** loop to print the days of the week. *Hint:* you might want to make use of `in`, `range` and `len`, as in the lecture notes. How would you implement this as a **while** loop?

Exercise 2: Loops and Input Processing

Write a script that:

1. Prompts the user to enter five comma-separated integer numbers.
2. Prints an error message if the user enters less than 5 numbers.
3. Outputs the sum of these numbers to the screen.
4. Outputs the number of valid values used in producing the cumulative sum.

Note: if any element is NOT a number, you should:

1. Treat it as NaN (Not a Number).
2. Exclude the NaN from the summation.

Hints:

1. As a way to check you have the information entered correctly, if you use the command `type()`, it should return `int` for integer numbers.
2. Any user input is a string. You can split it by any desired delimiter, e.g. by comma, and the result is a list. You will learn about lists next week. For now you can convert it in a tuple by function `tuple()`.

Exercise 3: The mean of a data set

Write a script that computes the mean of a user-supplied tuple that contains at least 5 numbers. For this script you should:

1. Prompts the user to enter at least five comma-separated integer numbers.

2. Check the entry contains numbers. If there are any non-numerical elements treat these element as a NaN.
3. Determine the number of valid numbers.
4. Compute the mean. If there are NaNs, the calculation of the mean should be adjusted, to take only genuine numbers into account.
5. Display the original entry, the mean and the number of valid numbers to the screen.

Reminder: mean is a ratio of summation of all elements to the count of elements.

$$\bar{x} = \frac{\sum x}{n}$$

Exercise 4: The standard deviation of a data set

Expand the script from Exercise 3 to also compute the standard deviation of the numerical elements. If there are less than 3 numerical elements entered, do not calculate the standard deviation. The full output should now be:

1. The original entry
2. The mean
3. The standard deviation, if calculated, otherwise display a message telling the user the standard deviation was not able to be calculated.
4. The number of valid numbers.

Reminder: standard deviation is a squared root of the ratio of summation of squared deviations of all elements to the count of elements minus one.

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$$

Exercise 5: Strings

For this exercise, you are asked to write a program that takes a string input from the user, and returns, as output, the entire message minus the vowels (a,e,i,o,u). E.g. if the input message is

```
Tim thinks this is a good exercise to try.
```

The output should look as follows:

```
Tm thnks ths s gd xrcs t try
```

This is the same exercise as in the lecture. Try to do it differently. You can try using a suitable built-in string function to help you make the coding task a little easier. Think about how you will handle upper or lower cases vowels as well. When storing the vowels, think about using a suitable data type.

Exercise 6: String-palindrome

Write a program that takes, as input, any word or phrase, entered by the user, and determine whether that word or phrase is a palindrome. A palindrome is a word that reads the same in reverse order, e.g.

- glenelg
- Anna

- Amore roma
- Poor Dan is in a droop
- Never odd or even

are palindromes, since if we read them from left to right, they produce the exact same word or phrase as if we read it from right to left. On the other hand, words or phrases such as

- Belinda
- String functions

aren't palindromes, since if we read them from right to left, we'd get something very different!

Hint: You may want to convert everything to one case (lower or upper) and remove white spaces to make things easier.

Exercise 7: Game 21

Write a program that plays as many games of 21 as a player would like. The idea behind 21 is that a player draws cards from a pack and either loses when the cumulative sum of the cards drawn exceed 21, or wins if they are as close to 21 as possible before stopping (they can also be equal to 21 when they stop and still win).

For this simplified version of the game, the cards drawn can take values 2 through to 11. Your code should:

1. Prompt the player to draw one card and display the value of the card drawn.
2. Ask whether they want to draw another card, or finish their game.
3. If the player chooses to finish the game, output the sum of the cards drawn and the value of the next card, should they have chosen to draw another card. If the sum plus the value of the next card exceeds 21, print a congratulations message for stopping in time. Otherwise, print a loser's message.
4. If the user chooses to draw another card, then if the sum of the cards drawn to date plus the newly drawn card exceeds 21, output the sum and a loser's message. Otherwise go to step 2 above.
5. If you would like an extra challenge, then treat the value of 11 as the Ace card. This means that the card can then either take on the value of 11, or the value of 1. If the user draws 11 and the value of 11 will cause them to lose, change the value to 1. Otherwise, keep the value of the card drawn as 11.

Hints:

1. To allow the user to play as many games as they like, as well as allowing them to continue to draw cards within a particular game, you will need to use loops.
2. To exit the loop(s) you could either test the condition as a part of the loop statement itself or use break – the choice is yours.