Data Handling for Business Analytics
Exercise number 2, Due Lecture #4

Test your code with different inputs and see how it behaves.

Unless stated otherwise, the inputs should come from the user, i.e., using `input()` func.

Please submit all the solutions in a **single .py** file (separated by #comments). Make sure that your **'.py'** file is **ready to run**, that is, loading it via Spyder and running it must work.

Do not send WORD files. Do not send print screens. Do not add your output.

Please name your .py file based on the following pattern: Ex(*number*)_ **ID**.py.
For example, if my ID is 1234 and this is exercise number 3, the name of the .py file should be: Ex3_1234.py.

Once done, upload your .py file to **Moodle.**

Feel free to ask questions regarding the HW/course material (via Moodle). Good luck!

1. Write a script that assigns (**hardcoded**, i.e. no input from the user) a list *l* and an element *elem* and prints *True* whether the element appears in the list and *False* otherwise.

2. Write a script that assigns (**hardcoded**, i.e. no input from the user) a list of strings to a variable called *l*. Once *l* is set, find the first string that contains the character 'a' and print it to the screen (if none of the strings contain the letter 'a', nothing should be printed to the screen). For example:
   Input (hardcoded):
   l = ['Ben','Noa','Banana']
   Output:
   'Noa'

3. Write a script that receives a *line* of text and a positive integer *i* (both are taken from the user), and return the i^{th} word in the *line*, where *i*==1 is the first word. You may assume *i* is no more than the number of words in *line*. For example:
   Input from the user:
   Trying is the first step towards success
   1
   Output:
   Trying

4. Write a script that assigns (**hardcoded**, i.e. no input from the user) a list of numbers to a variable called *l* (you may assume that *l* contains at least one item). Once *l* is set, find the minimal number (without using the built-in function min) in it and print this number to the screen.

5. Write a script that receives two different non-negative integers $n \neq m$ from the user and prints the **sum** of all the **odd** integers between them, but, **excluding**

them *(*note that the order in which the user enters the numbers should not affect the output):

i. using **for** loop.
ii. using **while** loop.

Input from the user:
9
3
Output:
12

(Comment: the output is 12 since the odd numbers between 3 and 9 (not including them) are 5 and 7)

6. Write a script that assigns (**hardcoded**, i.e. no input from the user) two lists to *l1, l2*. The output should be a new interleaved list. That is, the **first** element in the *new* list is the first number in the first list; the **second** element in the *new* list is the first number in the second list; the **third** element in the *new* list is the second number in the first list; the **fourth** element in the *new* list is the second number in the second list; and so on. If one of the lists is shorter than the other, the rest of the longer list will be copied sequentially to the end of the new list. For example:
Input (hardcoded):
l1 = [1,2,3]
l2 = [5,6,7,8,9,0]
Output:
[1, 5, 2, 6, 3, 7, 8, 9, 0]

Input (hardcoded):
l1 = [1,2,3,4]
l2 = [ ]
Output:
[1, 2, 3, 4]

Challenge (not mandatory but a good warmup for next lesson):

Write a script that receives (integer) numbers from the user until a negative number is given. Maintain a list of all the numbers (without the negative). If the negative number is even, print a list with all the numbers in **even** indices. Otherwise, print a **sorted** list of all the numbers in **odd** indices.
Hint: start by initializing an empty list **l=[]**