# Python Exercises

Rui Leite - 2022

## Exercises that don't use Modules

1. Given a list **L** with numbers extract the sub-list of the members that are on even positions (consider the position as the one that is used by python).

2. Consider that **L** is a list, write an instruction in python that shift all elements to the right moving the last to the first position.

   Example: if L=[4,7,4,3,1] then after executing the instruction L should be [1,4,7,4,3]

3. Given a list **L** with distinct numbers write an instruction in python that computes the second highest member.

4. Consider a list **L** with distinct numbers, write an one line instruction in python that swap the maximum with the minimum of **L**.

   Example: if L=[3,1,5,2,8,7] so after executing the instruction L should be [3,8,5,2,1,7].

5. Consider that we have two lists, **L** and **S** with the same size. Suppose also that the members of **S** are all of type **bool**. Write an instruction in python that output the sub-list of **L** whose corresponding members of **S** are **True**.

   Example: if L=[4,7,2,3] and S=[True, False, False, True] then the instruction should output [4,3]

6. Write an instruction in python to create a list with the integers from 1 to 100 (including 100) and create a variable **X** to store it.

7. Consider that you want to creates a list with the perfect squares from 1 to $100^2$ (including this last). Which instruction do you use inside the loop present bellow?

```
x=[]
for i in range(100):
    ?????
```

8. Write a one line instruction in python that create the same list mentioned in the last exercise.

   (a) using the function map

   (b) using list comprehension

9. Given a list **L** with numbers write an instruction in python that counts the number of times the maximum value occurs in **L**.

10. Given a list **L** with numbers write an instruction to determine if **L** has a memeber which is exactly 1/5 of the sum of all members.

11. A prime number is a integer (positive) with exatly 2 divisors. A method to find if **n** is prime consists to test all numbers from 1 to **n** and test if it divides **n**. Write a function (named prime) that receives a parameter **n** and returns True if is prime and False otherwise. (**hint: remember that the operator % computes the remainder**)

12. Note that the divisors of **n** can be organized in pairs (d,n/d).

    Example: 3 divides 12 and 12/3=4 also divides 12. To avoid counting a pair more than once we can restrict to represent (d,n/d) only in the case when d≤n/d.

    Using this information implement a more efficient function to test if **n** is a prime.

13. Suppose that you want to sort a list **L** with numbers without applying the sort function. An algorithm to do so is the following

```
create a new list X (starting empty)
While L is not empty
    remove the minimum value of L
    and insert it in the end of X
```

Write a function (named insertSort) that implements this algorithm

14. Consider that **L** is a list with integers. Write a segment of code in python that move all odd elements to the begining of the list and all even elements to the end of the same.

    Example: if L=[4,3,6,7,1,10,8] then after executing the instructions in the code segment L should be [3,7,1,4,6,10,8]

15. We want to define M as a "matrix" 8×8 filled with zeros representing it as a list of lists. The main list contains 8 lists each one with 8 zeros. We used the following instruction to create the list

    ```
    M = [ [0] * 8 ] * 8
    ```

    However this doesn't work, see what happens to M when we do M[0][0]=5.

    Find an explanation for what you observe.

    Write a different intruction that do what we want (initialize the "matrix" with zeros).

16. Consider that **L** is a list with type str. Write an instruction in python thst convert every value to upper case. (**hint:** Use the function **map**)

17. Write an instruction in python that checks if a list of numbers is sorted.

18. A moving average is a way to smooth data from a time serie. It consists of a moving window (of size k) for which the average is computed.

    Example: Consider the serie **S**=[3,6,2,5,7,8] and **k**=3 (moving window size) the moving average is (3+6+2)/3; (6+2+5)/3; (2+5+7)/3; (5+7+8)/3

    Write an instruction in python that computes the moving average of serie **S** considering the window size **k**=3.

19. Write an expression to compute the moving average of S for a given k (k<len(S)).

20. Consider a list **L** with members of basic types. Define a function (named uniq) in python to test that all members are unique (no item can occur in the list more than once).

21. Given to matrices **A** and **B** represented as list of lists with the same geometry (shape), define a function (named mSum) that accept the two matrices and return the matrix sum **A**+**B**.

    (a) implement function mSum using loops

    (b) implement function mSum using list comprehension

22. Implement an analogous function to mSum bit for matrix multiplication (call this last mMult).

23. Consider the following segment of python code where L has a list of numbers and is defined before

    ```
    S=[0]
    i=0
    while i < len(L):
         S.append(S[-1]+L[i])
    ```

    (a) What is wrong in this code? Correct it.

    (b) Implement the same using list comprehension.

24. Write an instruction in python that computes

    $$\sum_{i=1}^{100} \frac{1+i}{i^2}$$

25. Write an instruction in python that computes

    $$\sum_{i=1}^{20} \sum_{j=i}^{50} \frac{2+i^2}{i^3}$$

26. Suppose that **x** and **y** are variables with a numerical value and consider the following segment of code in python

    ```
    y = y - x
    x = x + y
    y = x - y
    ```

    (a) What is the effect of the code on the given variables?

    (b) Do the same using just a one line instruction.

27. Consider that **P1** and **P2** are points in a 2-dimensional space. Define a function (named dist) that computes the euclidian distance between **P1** and **P2**. (note that $\sqrt{x} = x^{0.5}$)

28. Consider that we have a list of lists that represent n points (from 2-dimensional space), write a function (named diameter) that uses the function defined in the previous question (dist) and computes maximum distance along every pair of points.

29. Consider that **words** is a list whose elements are of type str. Write a segment of instructions in python that creates a dictionary with the word as key and the number of occurences as data.

    **hint:** If you provide to function **dict** a list of tuples in the form (<key>,<data>) it constructs a dictionary with these entries.

    Example: dict([(”a”,3),(”b”,7)]) gives {”a”:3, ”b”:7}

30. Consider that you have stored the data about which items were bought on each customer ”basket”. Each of these sales can be also called as transaction. Suppose that you have **Tr** with a list of lists (not every list have the same number of elements) like the one in the following example

    [ [1,3,4], [1,4], [3,5,6], [1,2,4] ] to represent the case where the first transaction have items 1 2 and 3, the second transaction have items 1 and 4 and so on.

    For each of the following question write the appropriate instructions in python to produce the answer:

    (a) Define a list (named **S**) with the complete set of items mentioned in the transactions stored in **Tr**

    (b) Count the number of transactions where each element mentioned in **Tr** appears

    (c) For each pair of items count the number of transaction that contain both

    (d) Create a new structure of the form of list of lists where each member of the main list represent the transactions of one of the items coded as 1 if the corresponding transction contain that item and 0 otherwise. Considering the example given the asked structure is the following

$$[\ \overbrace{[1,1,0,1]}^{item\ 1}\ ,\ \overbrace{[0,0,0,1]}^{item\ 2}\ ,\ \overbrace{[1,0,1,0]}^{item\ 3}\ ,\ \overbrace{[1,1,0,1]}^{item\ 4}\ ,\ \overbrace{[0,0,1,0]}^{item\ 5}\ ,\ \overbrace{[0,0,1,0]}^{item\ 5}\ ]$$

## Exercises that use Numpy Module

Don't forget to do

**import numpy as np**

31. Consider that **a** and **b** are numpy arrays, both with only one row (row vectors), and with the same number of elements. Write an instruction in python that computes

$$\sum_{i=0}^{|a|-1} a_i \times b_i$$

32. Suppose that **x** is a numpy array with just one row filled with numbers. Write an instruction that compute the sum of all elements of **x** that are $> 5$.

33. Suppose that **x** is a numpy array with just one row filled with numbers. Write an instruction that counts the elements of **x** that are $> 5$.

34. Consider that you have a discrete random variable **X** assuming values in V=$\{v_1, ..., v_k\}$ and where the probability $\forall_{i \in \{1,...,k\}}\ P(X = v_i) = p_i$.

    The entropy (H) of **X** is defined as

$$H(x) = \sum_{i=1}^{k} -log_2(p_i) \times p_i$$

    where $log_2(0) \times 0$ is defined as 0

    Suppose that **P** is a list with the probabilities of the different values that **X** can assume and so **sum(P)=1**.

    Write an instruction in python that computes the entropy of **X**.

    (note that **P** should appear in the instruction and not **X**)

35. Write an instruction in python that creates a variable **M** with a 2-dimensional array with 4 rows and 3 columns with elements drawn randomly from a given vector **V** (1-dimensional numpy array) where any element of **V** has the same probability to be selected.

36. Consider that **M** is 2-dimensional numpy with 2 columns and that **W** is an one row array where the number of elements is the same as the number of rows of **M**.

    Write an instruction in python to scale each row of **M** using the corresponding value of **W**.

37. Consider a variation of the previous question but now the goal is to scale the columns of **M** and so the number of elements of **W** is equal to the number of columns of **M**.

    Write the instruction that do the scaling of the columns of **M** using the values of **W**.

38. Write an instruction in python that given **k** and **n** integers $\geq 0$ creates a numpy array with the following pattern

$$\begin{bmatrix} 1 & n+1 & 2n+1 & \dots & (k-1)n+1 \\ 2 & n+2 & 2n+2 & \dots & (k-1)n+2 \\ \dots & \dots & \dots & \dots & \dots \\ n & 2n & 3n & \dots & kn \end{bmatrix}$$

39. Write an instruction in python that given **k** and **n** integers $\geq 0$ creates a numpy array with the following pattern

$$\begin{bmatrix} 1^1 & 2^1 & 3^1 & \dots & k^1 \\ 2^2 & 3^2 & 4^2 & \dots & (k+1)^2 \\ \dots & \dots & \dots & \dots & \dots \\ n^n & (n+1)^n & (n+2)^n & \dots & (k+n)^n \end{bmatrix}$$

40. Consider that **M** is numpy array with **n** rows and **k** columns and **V** is an array with 1 row and **k** columns (a vector of size **k**).

    Write an instruction in python that result in a sub-array whose rows are different from **V**.

41. Given 2-dimensional array stored in **M** write an instruction in python that returns the sub array with all rows and columns except those that contains the maximum of **M**.

42. Given 2-dimensional array stored in **M** write an instruction in python that returns the sub array with all rows and columns except those where the corresponding elements are all greater than the average of all elements of **M**.

43. Given a square array stored in **M** write an instruction in python that computes its determinant.

    (you can list the linear algebra functions of numpy executing the instruction dir(np.linalg)

44. Consider that you have an array with **n** rows and **n+1** columns that represents the coefficients of a linear equation system where the last column is independent term.

Write an instruction in python that solves this linear equation system.

As a test for your answer you can consider the following example

$$\begin{cases} 3x + 5y + z = 3 \\ 7x - 2y + 4z = 4 \\ -6x + 3y + 2z = 2 \end{cases}$$

Where the corresponding **M** is defined with `np.array([[3,5,1,3],[7,-2,4,4],[-6,3,2,2]])`

45. Consider that a speaker challenge his audiance, consisting with 30 people, with the following bet.

"I bet that at least two of you have the same birthday".

What are the chances that the speaker wins the bet?

We want to use python to simulate a bet. All we need is to take 30 random samples from a set with 365 values (with replacement) and check if we have any repeated value.

(a) Write an instruction in python to simulate one of this bets.
**hint:** use functions np.random.choice and np.unique

(b) Write an instruction in python to do 1000 repetitions of that bet and count the number of wins.

There is a problem whose

46. Given a 1-dimensional array **V** with integers write an instruction in python that tests if there are repeated non zero elements.

Note that zero doesn't count to find repetitions and so if V=np.array([0,1,0,0,3]) then the result if False.

47. Given a 9x9 array **M** whose elements are integers between 0 and 9, representing a sudoku that is being solved (0 means an empty cell), write a segment of python code that tests if **M** is valid (no repeated values on a row, ...).

48. Consider that **M** is 2-dimensional array where the elements are integers from 0 to 7. Write an instruction in python that relocate every row with all zeros before all rows with at least one non zero element.

    **Example:**

$$\begin{bmatrix} 3 & 2 & 0 & 5 \\ 4 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 2 & 0 & 5 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

49. Consider a 1-dimensional array **V** with 1 row filled with integers $\geq 0$. Write a function (named **compRight**) that relocates all zeros to the left and replace all consecutive equal values ($\neq 0$) by the values 0 and the sum of those values.

    **Examples:**

    [2,2,0,0] → [0,0,0,4]

    [2,1,4,4] → [0,2,1,8]

    [4,4,1,1] → [0,0,8,2]

50. Consider a 2-dimensional array **M** with 4 rows and 4 columns filled with integers $\geq 0$. Generalize the function that you have created for the previous question to 4 functions: **mvRight**, **mvLeft** **mvUp** and **mvDown**.

    **mv Right** results by applying **compRight** to each row of **M**.

    The other 3 functions are the analogous versions for the other 3 directions.

    **Example:**

$$\begin{bmatrix} 0 & 2 & 0 & 2 \\ 4 & 2 & 0 & 0 \\ 4 & 0 & 0 & 4 \\ 0 & 2 & 2 & 8 \end{bmatrix} \rightarrow \text{(using } \mathbf{mvRight}\text{)} \quad \begin{bmatrix} 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 8 \\ 0 & 0 & 4 & 8 \end{bmatrix}$$

## Exercises that use pandas Module

(to complete soon)

## Exercises About Algorithm Implementation

(to complete soon)