LAB 2

1. Write a function to return a list of the first n numbers in the Fibonacci string.

2. Write a function that receives a list of numbers and returns a list of the prime numbers found in it.

3. Write a function that receives as parameters two lists a and b and returns: (a intersected with b, a reunited with b, a - b, b - a)

4. Write a function that receives as a parameters a list of musical notes (strings), a list of moves (integers) and a start position (integer). The function will return the song composed by going though the musical notes beginning with the start position and following the moves given as parameter.
        Example : compose(["do", "re", "mi", "fa", "sol"], [1, -3, 4, 2], 2) will return ["mi", "fa", "do", "sol", "re"]

5. Write a function that receives as parameter a matrix and will return the matrix obtained by replacing all the elements under the main diagonal with 0 (zero).

6. Write a function that receives as a parameter a variable number of lists and a whole number x. Return a list containing the items that appear exactly x times in the incoming lists.

   Example: For the [1,2,3], [2,3,4],[4,5,6], [4,1, "test"] and x = 2 lists [1,2,3] # 1 is in list 1 and 4, 2 is in list 1 and 2, 3 is in lists 1 and 2.

7. Write a function that receives as parameter a list of numbers (integers) and will return a tuple with 2 elements. The first element of the tuple will be the number of palindrome numbers found in the list and the second element will be the greatest palindrome number.

8. Write a function that receives a number x, default value equal to 1, a list of strings, and a boolean flag set to True. For each string, generate a list containing the characters that have the ASCII code divisible by x if the flag is

set to True, otherwise it should contain characters that have the ASCII code not divisible by x.

Example: x = 2, ["test", "hello", "lab002"], flag = False will return (["e", "s"], ["e" . Note: The function must return list of lists.

9. Write a function that receives as paramer a matrix which represents the heights of the spectators in a stadium and will return a list of tuples (line, column) each one representing a seat of a spectator which can't see the game. A spectator can't see the game if there is at least one taller spectator standing in front of him. All the seats are occupied. All the seats are at the same level. Row and column indexing starts from 0, beginning with the closest row from the field.

Example:

FIELD

[[1, 2, 3, 2, 1, 1],

[2, 4, 4, 3, 7, 2],

[5, 5, 2, 5, 6, 4],

[6, 6, 7, 6, 7, 5]]

Will return : [(2, 2), (3, 4), (2, 4)]

10. Write a function that receives a variable number of lists and returns a list of tuples as follows: the first tuple contains the first items in the lists, the second element contains the items on the position 2 in the lists, etc. Ex: for lists [1,2,3], [5,6,7], ["a", "b", "c"] return: [(1, 5, "a ") ,(2, 6, "b"), (3,7, "c")].

Note: If input lists do not have the same number of items, missing items will be replaced with None to be able to generate max ([len(x) for x in input_lists]) tuples.

11. Write a function that will order a list of string tuples based on the 3rd character of the 2nd element in the tuple. Example: ('abc', 'bcd'), ('abc', 'zza')] ==> [('abc', 'zza'), ('abc', 'bcd')]

12. Write a function that will receive a list of words as parameter and will return a list of lists of words, grouped by rhyme. Two words rhyme if both of them end with the same 2 letters.

Example:

group_by_rhyme(['ana', 'banana', 'carte', 'arme', 'parte']) will return [['ana', 'banana'], ['carte', 'parte'], ['arme']]