N/B: Include example output/test results as part of your solution and upload a zip file containing the two projects (F#, Python) or a zip file of .fs(x), .py and/or .txt file.

Questions 1 - 6 Multi-paradigm programming in F#

Question 1

We can decompose a list in F# using pattern matching. Given the following F# declarations:

```
sqrOnlyFirst ls =
    match ls with
      | hd :: _ -> hd * hd
```

Question 1a [4%]
What is the result of executing the function with the following calls below and why?

i. `sqrOnlyFirst [2;4;6]`
ii. `sqrOnlyFirst []`
iii. `Fix it`

Question 1b [6%]
Declare a function **`stringToList`**, that takes a string as an argument and returns the corresponding list of characters (Char).
Your solution must not use any of the built-in conversion features of F#.

Question 2

Question 2a [2%]
Declare an F# data type **`Vector`** for vectors of dimension two, three, four and five.

Question 2b [4%]
Declare a function **`vecLen`** that calculates the length of a vector.

Question 2c [4%]
Declare a function **`vecAdd`** that adds two vectors.

Question 3

Recursive functions and variants.

Question 3a [5%]
Declare a function **`rerun: string -> int -> string`**, so that executing **`rerun s n`** builds a new string by duplicating the string `s` `n` times.
For example: `rerun "code" 3` ➔ `"codecodecode"` and `rerun "code" 0` ➔ `""`.

Question 3b [5%]
Declare a tail-recursive variant of `rerun`.

Question 4

Given the following F# declarations of three functions `f1, f2,` and `f3`:

```
let f1 i j k =
    seq {
         for x in [0 .. i] do
             for y in [0 .. j] do
                 if x+y < k then yield (x,y)
    }

let f2 f p sq =
    seq {
         for x in sq do
            if p x then yield f x
    }

let f3 g sq =
    seq {
         for s in sq do
            yield! g s
    }
```

Question 4a [5%]
What is the value of List.ofSeq (f1 2 2 3).

Question 4b [5%]
Write an alternative declaration of f2 using functions from the Seq library of F#.

Question 5

Let's consider simple expressions, for example 1 + 2 * 3, which can be constructed from integer constants using binary operators. The expressions are modelled using the following F# declaration of the type expr:

```
type expr = | Const of int
            | BinOpr of expr * string * expr
```

where the construct Const generates an integer constant and the operator is given as a string (example: "+") when generating an expression using the constructor BinOpr.

Question 5a [4%]
Write three different values of type `expr`.

Question 5b [6%]
Declare a function `toString: expr -> string`, that gives a string representation for an expression. Put the brackets around every subexpression with operators. For example `(1+(2*3))` is a string representation of the example given above.

Question 6

Question 6 [10%]

Recall from the lessons that F# provides the generic MailboxProcessor class as its implementation of message passing and the actor model.
Write an F# agent/actor to add and get products, for instance in an inventory management system. The product can simply consist of a name and quantity.

Questions 7 - 10 Multi-paradigm programming in Python

Question 7

Question 7a [5%]

Given numLst= [1, 2, 3, 4, 5] we can compute the sum in an imperative way.
Write a Python program such that it uses imperative way to compute the sum of the list.

Question 7b [5%]

Using the imperative way of programming, check for even numbers given a user input:
Write a program, which repeatedly prompts the user for an integer.
If the integer is even, print the integer. If the integer is odd, don't print anything. Print "Bye for now!" and Exit the program if the user enters the integer 123.

Question 8

Question 8a [5%]

Given the following list:

```
data = [1, 2, 3, 4, 5]
```

Write a Python code that adds 10 to every element in the list using **lambda** and the **map()** function. Include the resulting list as part of your solution.

Question 8b [5%]

Using **lambda** and the **filter()** function, write a Python function to find even numbers (**printEvenNumbers**) given a list of natural numbers:
naturalNumbers = [0,1,2,3,4,5,6,7,8,9]

Question 9

Question 9a [5%]

Using the imperative coding style, implement a function groupList that given a list (list) and a group length (glength), returns a list of lists with length glength.
Example: testList = [1,2,3,4,5,6]
groupList(list, 2) gives [[1, 2], [3, 4], [5,6]] while groupList(list, 3) gives [[1, 2, 3], [4, 5, 6]]

Question 9b [5%]

Given the following list:

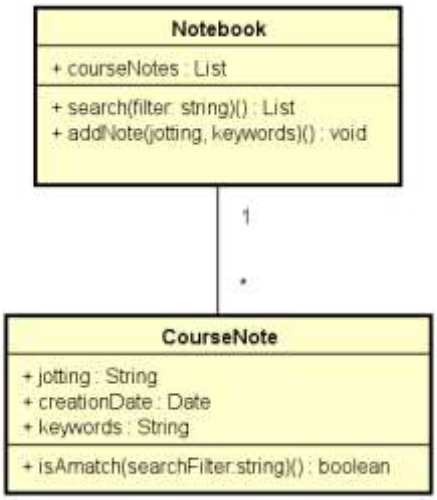
```
dkCities = ["Copenhagen", "Aarhus", "Aalborg", "Horsens", "Odense"]
```

Define a Python function ***filterCities()*** to filter cities that **starts with 'Aa'**.

Question 10

Question 10a [5%]

Implement the following UML class diagram in Python



Question 10b [5%]

Implement the following UML class diagram in Python, Test with your own values and include the output as part your solution