N/B: Include the output/test results from the terminal 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 1a [4%]

Given the following function declaration:

```
let funqy num1 num2 = num1 + num2
let gunqy(num1, num2) = num1 + num2
```

i. What is the output of the following and why?

```
funqy 5 gunqy(3,4)
```

ii. Modify it to fix the problem. Include the fixed code and the output from the terminal as part of your solution.

Question 1b [6%]

Declare a function **`funqy23_5`** with the type **`int -> bool`**, such that **`funqy23_5(n) = true`** exactly when **`n`** is divisible by 2 or divisible by 3 but not divisible by 5. Test with **`funqy23_5(24), funqy23_5(27), funqy23_5(29), funqy23_5(30)`**.
Hint: n is divisible by x when n%x = 0

Question 2a [4%]

Consider the following attempt to declare an F# function that uses the ***fold*** and ***accumulator*** function to **count the number of vowels in a string**:

```
let countNumOfVowels (str : string) =
    let charList = List.ofSeq str
    let accFunc (As, Es, Is, Os, Us) letter =
        match letter with
        | 'a' -> (As + 1, Es, Is, Os, Us)
        | 'e' -> (As, Es + 1, Is, Os, Us)
        | 'i' -> (As, Es, Is + 1, Os, Us)
        | 'o' -> (As, Es, Is, Os + 1, Us)
        | 'u' -> (As, Es, Is, Os, Us + 1)
        | _   -> (As, Es, Is, Os, Us)
    List.fold
```

Fix the issue and print each of the vowels with the corresponding number.
Test your solution with: ***countNumOfVowels*** **"Higher-order functions can take and return functions of any order"**

Question 2b [6%]
Declare a function **`replicateNtimes`** with the type **`int -> string -> string`**. The value of **`replicateNtimes n str`** is the string obtained by concatenating **`n`** copies of **`str`**. The function should raise an exception when the integer argument **`n`** is negative.
For example: **`replicateNtimes 3 "Fun"`** gives **`"FunFunFun"`**
Whereas **`replicateNtimes 0 "EMPTY"`** gives **`""`**.

Question 3a [4%]
Given the following:

```
type Variant =
        | Num of int
        | Text of string
        | Empty
```

Define a function ***printVariant*** such that given a variant, prints out the value or "Empty" if it is an Empty variant. Create 3 instances of the Variant types, use it to call the ***printVariant*** function and include the output from the terminal as part of your solution.

Question 3b [6%]
Given the following declaration that models algebraic expression:

```
type Expr = Num of int
          | Plus of Expr*Expr
          | Times of Expr*Expr
          | Neg of Expr
```

i. Declare a value of Expr that represents the following expression:

```
(6 * 10 + 25 * -2)
```

ii. The following function ***f*** gives a warning "Incomplete pattern matches on this expression". Fix the issue to remove the warning and call the function with the value declared in 3b.i above.

```
let rec f = function
    | Num(n) -> n
    | Plus(x,Neg(y)) -> f(x) - f(y)
```

Question 4a [6%]

Declare a function **`trixtrans`** that acts upon lists of lists and transforms the lists as shown in the figure below such that:

`trixtrans [[1; 2; 3]; [4; 5; 6]; [7; 8; 9]]` gives `[[1; 4; 7]; [2; 5; 8]; [3; 6; 9]]`

$$\begin{bmatrix}[1; & 2; & 3] \\ [4; & 5; & 6] \\ [7; & 8; & 9]\end{bmatrix} \; trixtrans \; = \begin{bmatrix}[1; & 4; & 7] \\ [2; & 5; & 8] \\ [3; & 6; & 9]\end{bmatrix}$$

Question 4b [4%]

Consider the following function declaration:

```
open System.IO
let pclxms24 path =
    Directory.EnumerateFiles(path, "*.fs*",  SearchOption.AllDirectories)
    |> Seq.toList
```

i. What is the result of the following function call? Describe in words what the function does.

```
pclxms24 (".");;
```

ii. Declare a function/value to count the number of elements in the result from 4b.i above and include the output from the terminal as part of your solution.

Question 5a [6%]

Consider the following type definitions to model the Customer and Order part of GTG course project including a list of orders:

```
type Customer =
    | VIACustomer
    | SOSUCustomer

type Order = {
    OrderID : string
    CustomerID : string
    Amount  : float
    Origin : Customer
}
let orders = [
    {OrderID="22401"; CustomerID="1101"; Amount = 245.00; Origin = Customer.VIACustomer}
    {OrderID="22402"; CustomerID="3302"; Amount=245.00; Origin = Customer.VIACustomer}
    {OrderID="22403"; CustomerID="2201"; Amount=245.00; Origin = Customer.SOSUCustomer}
    {OrderID="22404"; CustomerID="1102"; Amount= 255.00; Origin = Customer.VIACustomer}
    {OrderID="22405"; CustomerID="2202"; Amount = 245.00; Origin = Customer.SOSUCustomer}
    {OrderID="22406"; CustomerID="1103"; Amount=500.00; Origin = Customer.VIACustomer}]
```

i). Declare an F# function that takes the list of orders and prints out orders originating from VIACustomer

ii). Declare an F# function/value that takes the list of orders from 5a.i above and calculates the total amount from VIA customers.

Question 5b [4%]

Rewrite the following function to **use pattern matching**, and test with two customers of your choice.

```
let getLunch x =
    let customer, foodChoice = x
    if foodChoice = "veggies" || foodChoice ="fish" ||
       foodChoice = "chicken" then
        sprintf "%s doesn't want red meat" customer
    else
        sprintf "%s wants 'emmm delicious %s" customer foodChoice
```

Question 6 [10%]
This requires that you add the code to your GTG course project and include the results and only the applicable code snippet for the question.
Supposing we were given a new requirement to expand our GTG course project to add new coffee drink products. The **new coffee drink product** is a special coffee of three different cup sizes, **GtgDemi**(9cl), **GtgShort**(24cl) and **GtgTall**(36cl)

Question 6a [4%]
i. Define the data type for the new coffee cup **size**s, a **new coffee record** and the corresponding **price calculation** function and include it in your GTG course project.
Include only the snippet for the data type and price calculation function as the answer for this question.

Question 6b [6%]
ii. Modify your OrderProduct (or OrderDrink) message and ***gtgAgent*** mailboxprocessor of your course project to allow users to order for the newly added coffee drink products.
iii. Post two order drink messages, one for normal coffee and the second for **GtgCoffee** and include your results/output.

Questions 7 - 10 Multi-paradigm programming in Python

Question 7
Define functions, that given a list of integers (`int`) as an argument, returns the sum of all the positive numbers in the list. For example, summing: `[2, -3, 4, -5, 6]` should give `12`

Question 7a [7%]

i. Using the **imperative** paradigm of programming in Python, define a function to sum all the positive numbers in the list.
ii. Using the **functional** paradigm in Python, define a function to sum all the positive numbers in the list.

Question 7b [3%]
Define a **higher-order function using lambda** in Python to print even numbers given a list of natural numbers. Test with: `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

Question 8a [5%]
Using the imperative coding style, implement a function ***group_list(lst, gl)*** that when given a list (lst) and a group length (gl), returns a list of lists with length gl.
Example: `lst = [1, 2,3,4,5,6]`
`group_list(lst, 2)` gives `[ [1, 2], [3, 4], [5,6] ]` while
`group_list(lst, 3)` gives `[ [1, 2, 3], [4, 5, 6] ]`

Question 8b [5%]
Imagine that you want to compute the weekly average time one spends doing exercises. Write a Python program that prompts a user to enter some numbers and when done, prints out the average. The program should keep asking the user to enter a number until he/she enters the number 0, at which point the program should print the average and stop. The output should show the minutes and seconds. For example:

```
Enter a number in minutes: 30
Enter a number in minutes: 20
Enter a number in minutes: 60
Enter a number in minutes: 15
Enter a number in minutes: 0
Your weekly average exercise time is 31 minutes 25 seconds.
```

Question 9
Imagine that we have a requirement to include some accounting features in the GTG Course project. Using the following data representing (product, type, quantity, unit price):

```
# product, type, quantity, unit price
gtg_data2 = [('Coffee', 'Drink', 1015, 15.05),
             ('Juice', 'Drink', 800, 6.05),
             ('Tuna Mini Sandwiches', 'Food', 800, 6.05),
             ('Apple', 'Fruit', 925, 5.15),
             ('Tuna Mini Sandwiches', 'Food', 800, 6.05),
             ('Green Tea', 'Drink', 630, 12.05),
             ('Veggie Sandwiches Mix', 'Food', 800, 6.05),
             ('Banana', 'Fruit', 915, 3.10)]
```

Question 9a [5%]
Define a Python function to filter products of type Drink given the gtg_data2 above.
Requirement: Use Higher-order function.

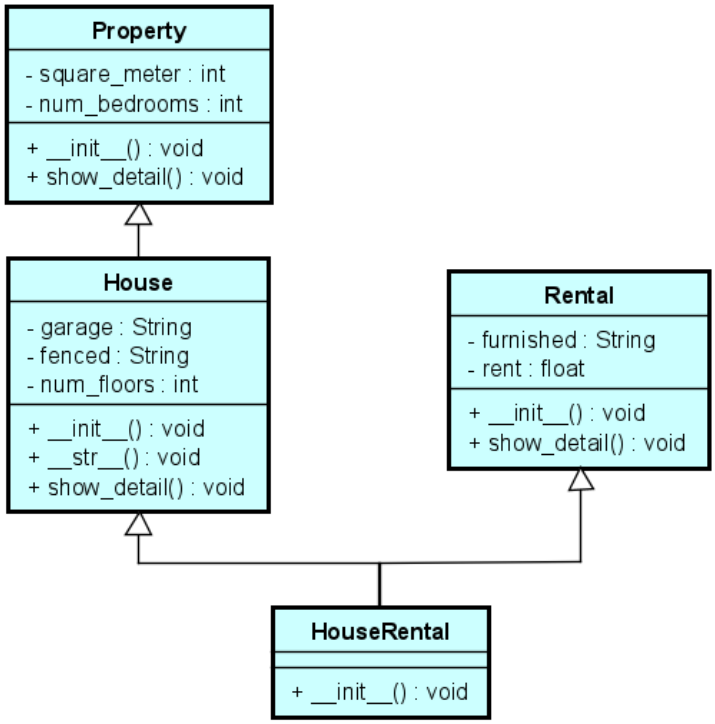
Question 9b [5%]
Using the functional programming paradigm, declare Python functions that returns a list of two tuples, containing the drink and the corresponding subtotal (where subtotal = quantity * unit price) before finally returning the total amount of all the (filtered) drinks.
Requirement: Use Higher-order function.

Question 10a [3%]
Write a Python program to read a user input and compute the tax and discount. The program will begin by reading the cost of a product (Drink, Food or Fruit) ordered at the GTG Café from the user and if the user is a VIA Customer. Compute a 5 percent discount of the cost before tax if the customer is a VIA Customer.

Use the same rate as ***gtgVAT*** from **Sprint 2.2 of the Course Project** for the tax calculation.

The output from your program should include the tax amount, the discount amount, and the grand total for the order including both the tax and the discount. Test your program with inputs for both cases (VIA customer and non-VIA customer)

Question 10b [7%]
Recall from the lessons that Python supports multiple inheritance. Implement the following UML class diagram including the **constructors** (`__init__()` with the necessary parameters/arguments) and specified **methods**. Decide for yourself the rent, when testing your code :-).



Create object of all the classes with examples of your choice and call the ***show_detail()*** method. Include the output as part of your solution.