Exam #2, on November 1, 2019, will consist of questions taken or inspired from this homework, the previous homework, project #2 and from the labs.

Part I – Questions

Q. 1 - Assume you are given an un-assigned `string` variable `letterGrade`, and an already assigned `float` variable `numberGrade`. Write a small program that assigns "A" to `letterGrade` if `numberGrade` is between 100 and 90 (both included), "B" to `letterGrade` if `numberGrade` is between 90 (excluded) and 80 (included), etc., and "Invalid data" if `numberGrade` is stricly lower than 0 or stricly greater than 100.

Should you use a `switch` statement or a `if ...else if ...else`?

Q. 2 - Given an `int` variable `counter`, write three different statements to decrement its value by 1.

Q. 3 - What will be displayed at the screen by the following program?

```
int x = 3, y = 7;
Console.WriteLine (x++ +" and "+ --y);
```

Q. 4 - What will be displayed at the screen by the following program?

```
int counter = 2;
while (counter != 5)
{
        Console.Write(counter + "\n");
        counter++;
}
```

Q. 5 - What will be displayed at the screen by the following program?

```
int counter = 10;
while (counter != 5) ;
Console.Write(counter + "\n");
counter--;
```

Q. 6 - What will be displayed at the screen by the following program?

```
int counter = 7;
while (counter != 2)
Console.Write(counter + "\n");
counter--;
```

Q. 7 - What is input validation? Name a control structure that can be used to perform it. Why is it important?

Q. 8 - What do we name a variable that is incremented at every iteration of a loop, i.e., that keeps the running total?

Q. 9 - What is a sentinel value?

Q. 10 - Write a program that asks the user to enter a value between 0 and 10, and asks again as long as the user enters integers outside that range.

Q. 11 - Write a small program that asks the user for an integer, and displays "It is positive" if the number entered is posivite, "It is negative" if the number entered is negative, and "Not a number" if the user entered a string that is not an integer.

Part II – Problems

Problem 1

Write a `switch` statement that calculates the number of days in a particular month. You should assume that you are given already assigned `month` and `year` `int` variables, and that your program should set an already declared `int` `numberOfDays` variable to 28, 29, 30 or 31 depending on the month / year combination.

Your program should start with a `switch` matching `month` against certain values, and, if `month` is 2, uses an `if` statement to decide whenever the number of days is 28 or 29. You can use something like

```
switch (month) {
        ⋮
        case (2):
        if ...
        ⋮
        break;
        ⋮
}
```

Problem 2

Read the following code, and answer the following two questions:

```
switch (citizenship) {
        case ("US"):
        case ("CA"):
                if (income > 100)
                        if (age < 21) Console.WriteLine("Go to office A.");
                        else if (age < 60) Console.WriteLine("Go to office B.");
                        else Console.WriteLine("Go to office C.");
                else
                        Console.WriteLine("Go to office D.");
        break;
        case ("DE"):
                if (income > 200 && age > 18) Console.WriteLine("Go to office E.");
                else Console.WriteLine("Go to office F.");
        break;
        case ("FR"):
                if (age <= 18 || income <= 10) Console.WriteLine("Go to office G.");
                else if (income > 200) Console.WriteLine("Go to office H.");
        break;
```

```
        default:
                if (age > 21) Console.WriteLine("Go to office I.");
                else Console.WriteLine("Go to office J.");
        break;
}
```

1. Depending on the values of `age`, `citizenship` and `income`, determine what would be displayed at the screen:

`age`	`citizenship`	`income`	"Go to office…"
18	`"CA"`	80	
21	`"UK"`	150	
18	`"US"`	120	
25	`"FR"`	210	
25	`"DE"`	210	
23	`"AZ"`	150	
21	`"FR"`	8	
18	`"DE"`	200	
62	`"CA"`	120	

2. Give a set of ("legal") values for which nothing would be displayed.

Problem 3

We want to write a program for a coffee machine. The user should chose between three sizes ("Small", "Medium" or "Large") and a type of roast ("Light", "Medium" or "Dark"). Then, the user should be given an amount to pay: a small coffee is $1.00, a medium one is $1.20, a large one is $1.50, and a 10% extra fee is applied for the dark roast. Finally, the user should be asked how much they want to tip ($0.00 being possible, but not negative numbers), and the program should display the total.

For the selection of the coffee, you can ask the user to enter only one letter, the whole string, a number that corresponds to the choice, …, anything that seems appropriate, but you have to perform user input validation on all the data given by the user. Below is an example of running the program where user input is underlined, and hitting "return" is represented by ↵. You do not have to reproduce it exactly, it is just to illustrate the expected behavior.

```
Do you want a small (1), a medium (2) or a large (3) coffee?
S ↵
Do you want a small (1), a medium (2) or a large (3) coffee?
1 ↵
Do you want a (L)ight, a (M)edium or a (D)ark small coffee?
D ↵
```

```
A small dark coffee is $1.10. How much do you want to tip?
-3 ↵
How much do you want to tip?
0.9 ↵
Your total is $2.00.
```

Your program should use, if possible, `do while` loops.