Instructions: This exam is to be taken in silence, without notes, books, or electronic devices (including "smart" watches or earbuds). The time limit to complete it is the duration of the class period (1 hour and 15 minutes). Answer the following questions and problems, trying to be as clear and as accurate as possible. Take your time to read the statements carefully before trying to answer them. If you need more space, write on the back of your test page and indicate it clearly. When writing code, make sure your special punctuation characters are legible, and your lowercase and uppercase letters are easy to distinguish. As usual, every statement or series of statement is assumed to be in a valid class and method, and you can use the `C.RL()`, `C.W()` and `C.WL()` abbreviations.

____ / **15 pts.** **Problem 1** For each of the following three snippet of code, indicate what would be displayed.

```
int Myst1(int n)
{
    if (n != 0) return n + Myst1(n - 1);
    else return n;
}
Console.WriteLine(Myst1(4));
```

```
void Myst2(int n)
{
    if (n == 0) { Console.WriteLine("Done"); }
    else if (n < 0)
    {
        Console.Write($"{n} ");
        Myst2(-n);
    }
    else
    {
        Console.Write($"{n} ");
        Myst2(-(n - 1));
    }
}
Myst2(3);
```

```
void Myst3(int len)
{
    MystH(0, 1, 1, len);
}
void MystH(int a, int b, int counter, int len)
{
    if (counter <= len)
    {
        Console.Write($"{a} ");
        MystH(b, a + b, counter + 1, len);
    }
}
Myst3(6);
```

____ / **30 pts.** **Problem 2** Suppose that at `filePath` is located a file where each line contains either

- a decimal (e.g., 12.4, -14, 0.34),
- the word "STOP",
- some other string ("Test", "The sky is blue", "Ignore this line", "My file contains").

Write a program that displays the average of the decimals in the file knowing that

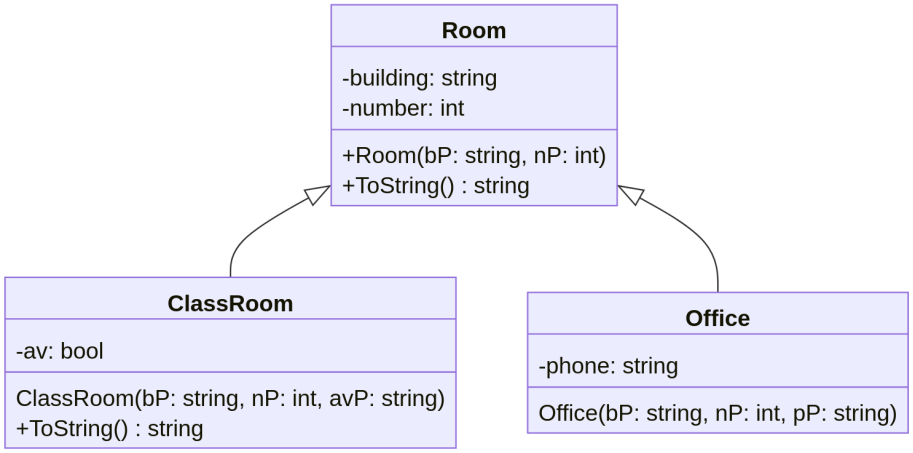
- your program should ignore the values after the line that contains "STOP" if it is present,
- all the other strings should simply be ignored.

For example, for the following two files, "4.0" and "10.0" should be displayed, as (12.48 - 2.48 + 2) / 3 = 4 (13 been ignored) and (15 + 5) / 2 = 10.

```
┌──────────────────┐
│12.48             │
│This is a test    │
│-2.48             │
│2                 │
│STOP              │
│13                │
└──────────────────┘
```

```
┌──────────────────┐
│My file contains  │
│STOP but          │
│averages          │
│15                │
│     and          │
│5                 │
└──────────────────┘
```

____ / 20 pts. **Problem 3** Consider the following diagram and code:



Suppose you are given an implementation of the `Room` class, such that

```
Room test = new Room("UH", 243);
Console.WriteLine(test);
```

displays

```
UH 243
```

1. Write an implementation of the `ClassRoom` class. Your `ToString` method should display the room's building and number, in addition to whether it has AV set-up.

2. Write a `SameBuilding` static method to be placed inside the `Room` class such that

```
Office test1 = new Office("UH", 127, "706 737 1566");
ClassRoom test2 = new ClassRoom("UH", 243, true);
Office test3 = new Office("AH", 122, "706 729 2416");
Console.WriteLine(Room.SameBuilding(test1, test2));
Console.WriteLine(Room.SameBuilding(test2, test3));
```

Would display "true" and "false".

____ / **30 pts.** **Problem 4** Write the `AddRev`, `AddLog` and `AddAndReset` methods (header included, as if they were part of the `Main` method) so that the snippets on the left column would display what is given in the right column.

1. `AddRev` implements the reversible addition: after its execution, the first parameter contains the sum of both parameters, while the second contain their difference.

```
int x = 4, y = 3;
AddRev(ref x, ref y);
C.WL($"x is {x}, y is {y}.");
```

Output window:

```
┌──────────────────┐
│ x is 7, y is 1.  │
└──────────────────┘
```

2. `AddLog` returns the sum of its `int` arguments and "logs" (that is, records) the operation inside a `string`.

```
string lg;
int x = 4, y = 3;
int res = AddLog(x, y, out lg);
C.WL(res + "\n" + lg);
```

Output window:

```
┌──────────────────┐
│ 7                │
│ 4 + 3 = 7        │
└──────────────────┘
```

3. `AddReset` resets to 0 its arguments, and store their sum in a third reference variable.

```
int a = 2, b = 3, c;
AddReset(ref a, ref b, out c);
C.WL($"a={a}, b={b}, c={c}.");
```

Output window:

```
┌──────────────────┐
│a=0, b=0, c=5.    │
└──────────────────┘
```

____ / **10 pts.** **Problem 5** Give the name and signature of one method from the `String` class, one example usage, and explain what your code would do.