Practical – 3

Question 1:

We want to develop a program that can do the following:

· Prompt the user for input of two integers, which we will call numerator and denominator. For clarity, we are only looking at integers, because this assignment is about rational numbers. A rational number can always be expressed as a quotient of two integers.

· Calculate the floating point division result (e.g. 10/4 = 2.5).

· Calculate the quotient and the remainder (e.g. 10/4 = 2 with a remainder of 2 = 2 2/4).

Code:

Program.cs

```
1   int a, b;
2   int r;
3   double d;
4   Console.WriteLine("Please enter the numerator: ");
5   a = Convert.ToInt32(Console.ReadLine());
6   Console.WriteLine("Please enter the denominator: ");
7   b = Convert.ToInt32(Console.ReadLine());
8   d = (double)a / b;
9   r = a % b;
10  Console.WriteLine($"Integer division result = {(int)d} with a remainder {r}");
11  Console.WriteLine($"Floating point division result = {d}");
12  Console.WriteLine($"The result as a mixed fraction is {(int)d} {r}/{b}");
```

Output:

```
Please enter the numerator:
14
Please enter the denominator:
4
Integer division result = 3 with a remainder 2
Floating point division result = 3.5
The result as a mixed fraction is 3 2/4
```

Question 2:

1. Read a string from the keyboard and print the length of the string, with a label.

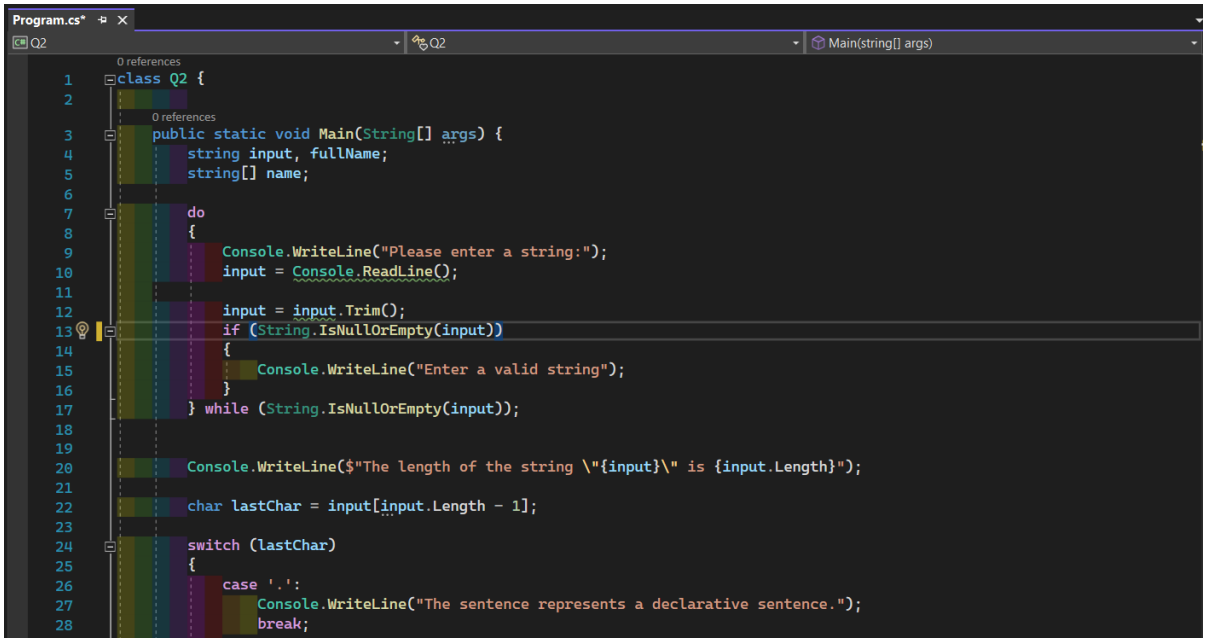
2. Read a sentence (string) from a line of input, and print whether it represents a declarative sentence (i.e. ending in a period), interrogatory sentence (ending in a question mark), or an exclamation (ending in exclamation point) or is not a sentence (anything else).

It makes sense to only make small changes at once and build up to final code. First you might just code it to check if a sentence is declarative or not. Then remember you can test further cases with else if (...).

3. Read a whole name from a single line of user input. Do not ask for first and last names to be entered on separate lines! Assume first and last names are separated by a space (no middle name). Print last name first followed by a comma and a space, followed by the first name. For example, if the input is "Marcel Proust", the output is "Proust, Marcel".

4. Improve the previous part, so it also allows a single name without spaces, like “Socrates”, and prints the original without change. If there are two parts of the name, it should work as in the original version.

Code:



```
            case '!':
                Console.WriteLine("The sentence represents exclamation");
                break;
            case '?':
                Console.WriteLine("The sentence represents an interrogatory sentence");
                break;
            default:
                Console.WriteLine("The sentence is a normal sentence");
                break;
        }

        do
        {
            Console.WriteLine("\n\nEnter your full name:");
            fullName = Console.ReadLine();
            fullName = fullName.Trim();
            if (String.IsNullOrEmpty(fullName))
            {
                Console.WriteLine("Enter a valid name");
            }
        } while (String.IsNullOrEmpty(fullName));



        name = fullName.Split(' ');

        if (name.Length >= 2)
        {
            Console.WriteLine($"Name: {name[1]}, {name[0]}");
        }
        else if (name.Length == 1)
        {
            Console.WriteLine($"Name: {name[0]}");
        }
    }
}
```

Output:

```
Please enter a string:
Hello!
The length of the string "Hello!" is 6
The sentence represents exclamation


Enter your full name:
Vedant Joshi
Name: Joshi, Vedant
```

```
Please enter a string:
Hello?
The length of the string "Hello?" is 6
The sentence represents an interrogatory sentence


Enter your full name:
Vedant
Name: Vedant
```

Question 3: Enumeration Sample with bit flags (C# Microsoft.com)

Program.cs

Q3 | FlagsEnumExample | Main()

```
1   [Flags]
2
    10 references
3   public enum Days
4
5   {
6       None = 0b_0000_0000,
7       Monday = 0b_0000_0001,
8       Tuesday = 0b_0000_0010,
9       Wednesday = 0b_0000_0100,
10      Thursday = 0b_0000_1000,
11      Friday = 0b_0001_0000,
12      Saturday = 0b_0010_0000,
13      Sunday = 0b_0100_0000,
14      Weekend = Saturday | Sunday
15  }
16
    0 references
17  public class FlagsEnumExample
18
19  {
20
        0 references
21      public static void Main()
22
23      {
24
25          Days meetingDays = Days.Monday | Days.Wednesday | Days.Friday;
26          Console.WriteLine(meetingDays);
27
28          Days workingFromHomeDays = Days.Thursday | Days.Friday;
29          Console.WriteLine($"Join a meeting by phone on {meetingDays & workingFromHomeDays}");
30
31          bool isMeetingOnTuesday = (meetingDays & Days.Tuesday) == Days.Tuesday;
32          Console.WriteLine($"Is there a meeting on Tuesday: {isMeetingOnTuesday}");
33
34          var a = (Days)25;
35          Console.WriteLine(a);
36      }
37
38  }
```

Output:

```
Microsoft Visual Studio Debug Console
Monday, Wednesday, Friday
Join a meeting by phone on Friday
Is there a meeting on Tuesday: False
Monday, Thursday, Friday
```