**Name :** Rahul H Prajapati **PRN :** 2019033800125516

**Assignment - 3**

**GitHub Link : “https://github.com/RahulPr2408/.Net-Lab\_Ass\_3.git”**

**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 :**

using System;

namespace Code\_1

{

class Program

{

static void Main(string[] args)

{

int numerator;

int denominator;

Console.Write("Please enter the numerator : ");

numerator = int.Parse(Console.ReadLine());

Console.Write("Please enter the denominato : ");

denominator = int.Parse(Console.ReadLine());

int quotient = numerator / denominator;

int remainder = numerator % denominator;

Console.WriteLine($"Integer division result = {quotient} with a remainder = {remainder}");

double fraction\_quotient = Math.Round((double)numerator/denominator);

Console.WriteLine($"Floating point division result : {fraction\_quotient}");

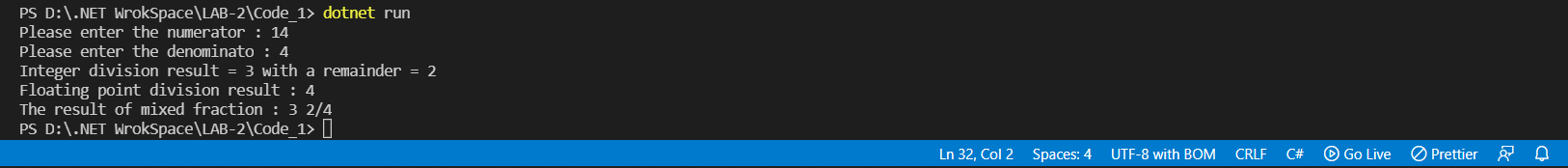
Console.WriteLine($"The result of mixed fraction : {quotient} {remainder}/{denominator}");

}

}

}

**Output :**

****

**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 (...).**

1. **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".**
2. **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 :**

using System;

namespace Code\_2

{

class Program

{

static void Main(string[] args)

{

// find length of a String

Console.Write("Enter a string to find Length : ");

String str1 = Console.ReadLine();

int length = str1.Length;

Console.WriteLine($"The length of {str1} = {length}");

// check sentences

String str2;

Console.Write("Enter a Sentence : ");

str2 = Console.ReadLine();

checkSentence(str2);

// full name

String str3;

Console.Write("Enter Your Full name (without Middle name) : ");

str3 = Console.ReadLine();

FullName(str3);

}

public static void checkSentence(String str) {

if(str.EndsWith(".")) {

Console.WriteLine("The sentence is Declarative");

}

else if (str.EndsWith("?")) {

Console.WriteLine("The sentence is Interogatory");

}

else if (str.EndsWith("!")) {

Console.WriteLine("The sentence is Exclaimatory");

}

else {

Console.WriteLine("This is not a sentence ");

}

}

public static void FullName(String str) {

int index = str.IndexOf(" ");

String firstName = str.Substring(0,index);

String lastName = str.Substring(index);

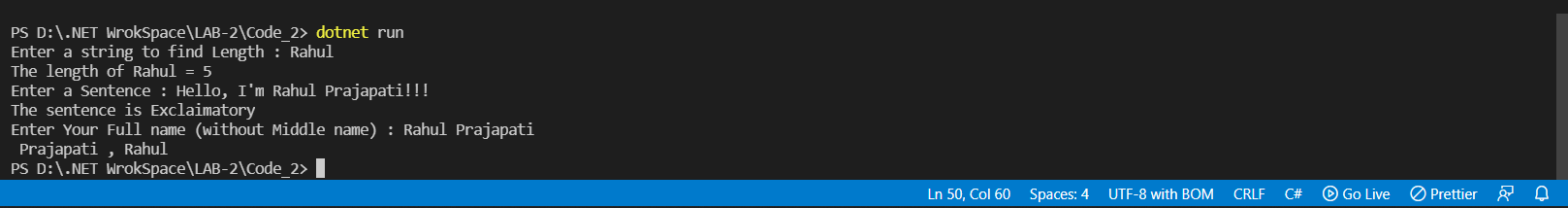
Console.WriteLine($"{lastName} , {firstName}");

}

}

}

**Output :**

****

**Question 3:**

**Enumeration Sample with bit flags**

**Code :**

using System;

namespace Code\_3

{

public enum Days

{

None = 0b\_0000\_0000, // 0

Monday = 0b\_0000\_0001, // 1

Tuesday = 0b\_0000\_0010, // 2

Wednesday = 0b\_0000\_0100, // 4

Thursday = 0b\_0000\_1000, // 8

Friday = 0b\_0001\_0000, // 16

Saturday = 0b\_0010\_0000, // 32

Sunday = 0b\_0100\_0000, // 64

Weekend = Saturday | Sunday

}

class Program

{

static void Main(string[] args)

{

Days meetingDays = Days.Monday | Days.Wednesday | Days.Friday;

Console.WriteLine(meetingDays);

// Output:

// Monday, Wednesday, Friday

Days workingFromHomeDays = Days.Thursday | Days.Friday;

Console.WriteLine($"Join a meeting by phone on {meetingDays & workingFromHomeDays}");

// Output:

// Join a meeting by phone on Friday

bool isMeetingOnTuesday = (meetingDays & Days.Tuesday) == Days.Tuesday;

Console.WriteLine($"Is there a meeting on Tuesday: {isMeetingOnTuesday}");

// Output:

// Is there a meeting on Tuesday: False

var a = (Days)37;

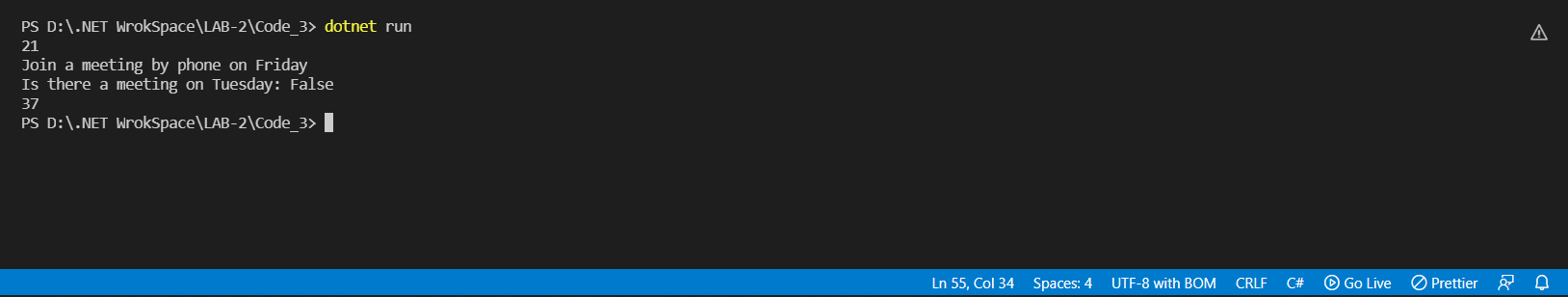
Console.WriteLine(a);

}

}

}

**Output :**

****