| NAME | RAJ SANJAY JADHAV |
| --- | --- |
| CLASS | BE-3-CSE |
| BATCH | B |
| ROLL NO | 512024 |
| PRN | 2019033800129402 |

**PRACTICAL-3**

**GIT REPOSITORY :** [**PRACTICAL-3**](https://github.com/Rajjadhav1710/.NET_Practical/tree/main/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:**

**using System;**

**class DoTheMath {**

**static void Main(){**

**int numerator,denominator;**

**Console.WriteLine("\nPlease enter the numerator?");**

**numerator=Convert.ToInt32(Console.ReadLine());**

**Console.WriteLine("Please enter the denominator");**

**denominator=Convert.ToInt32(Console.ReadLine());**

**Console.WriteLine("\nInteger division result = "+numerator/denominator+" with a remainder "+numerator%denominator);**

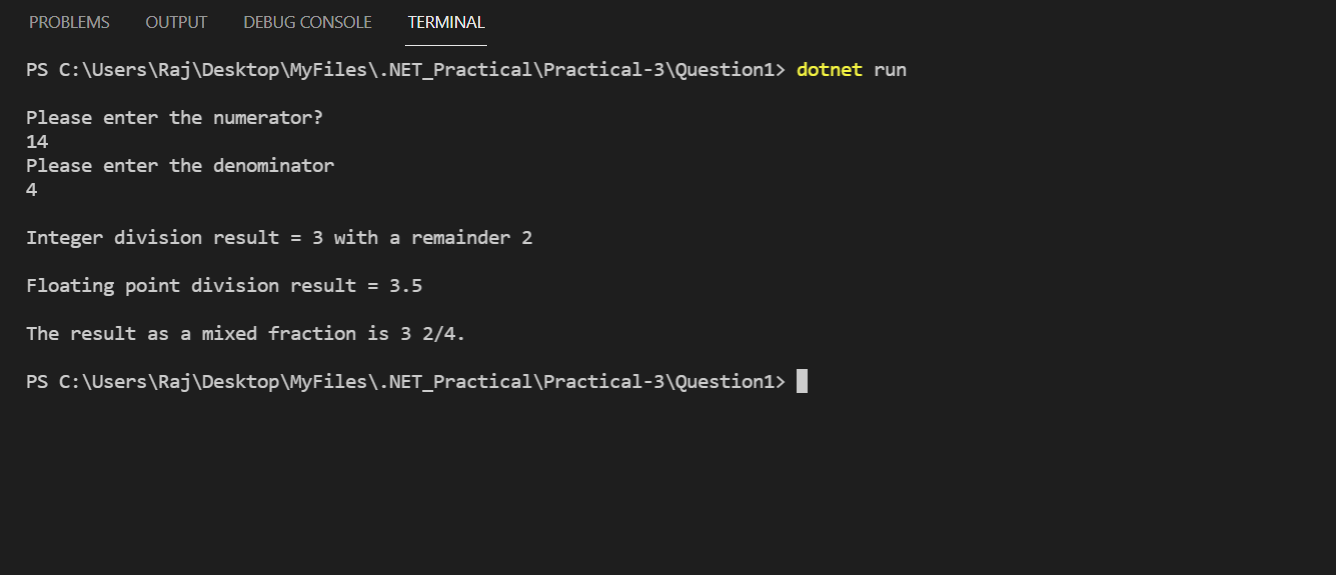
**Console.WriteLine("\nFloating point division result = "+((double)numerator)/denominator);**

**Console.WriteLine("\nThe result as a mixed fraction is "+numerator/denominator+" "+numerator%denominator+"/"+denominator+".\n");**

**}**

**}**

**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 (...).

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:**

**using System;**

**class DoTheMath {**

**static void Main(){**

**Console.WriteLine("\nGive Me A String To Find Length:");**

**string userString=""+Console.ReadLine();**

**Console.WriteLine("\nLength Of The String:"+userString.Length);**

**Console.WriteLine("\nGive Me A Sentence To Identify It's Type:");**

**string userSentence=""+Console.ReadLine();**

**if(userSentence[userSentence.Length-1]=='.'){**

**Console.WriteLine("\nDeclarative Sentence.");**

**}else if(userSentence[userSentence.Length-1]=='?'){**

**Console.WriteLine("\nInterrogatory Sentence.");**

**}else if(userSentence[userSentence.Length-1]=='!'){**

**Console.WriteLine("\nExclamatory Sentence.");**

**}else{**

**Console.WriteLine("\nNot A Sentence.");**

**}**

**Console.WriteLine("\nGive Me Your Full Name:");**

**string fullName=""+Console.ReadLine();**

**string[] splittedName=fullName.Split(' ');**

**if(splittedName.Length==1)**

**Console.WriteLine("\nYour Name: "+splittedName[0]);**

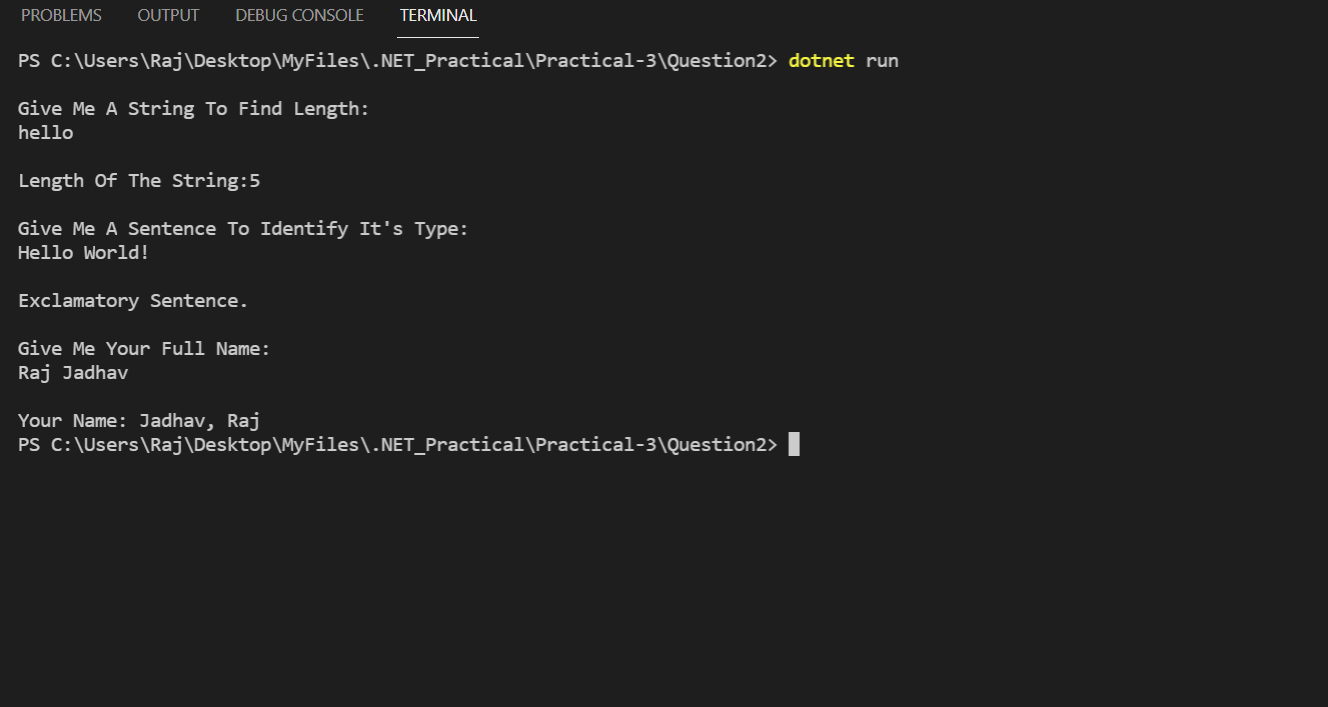
**else**

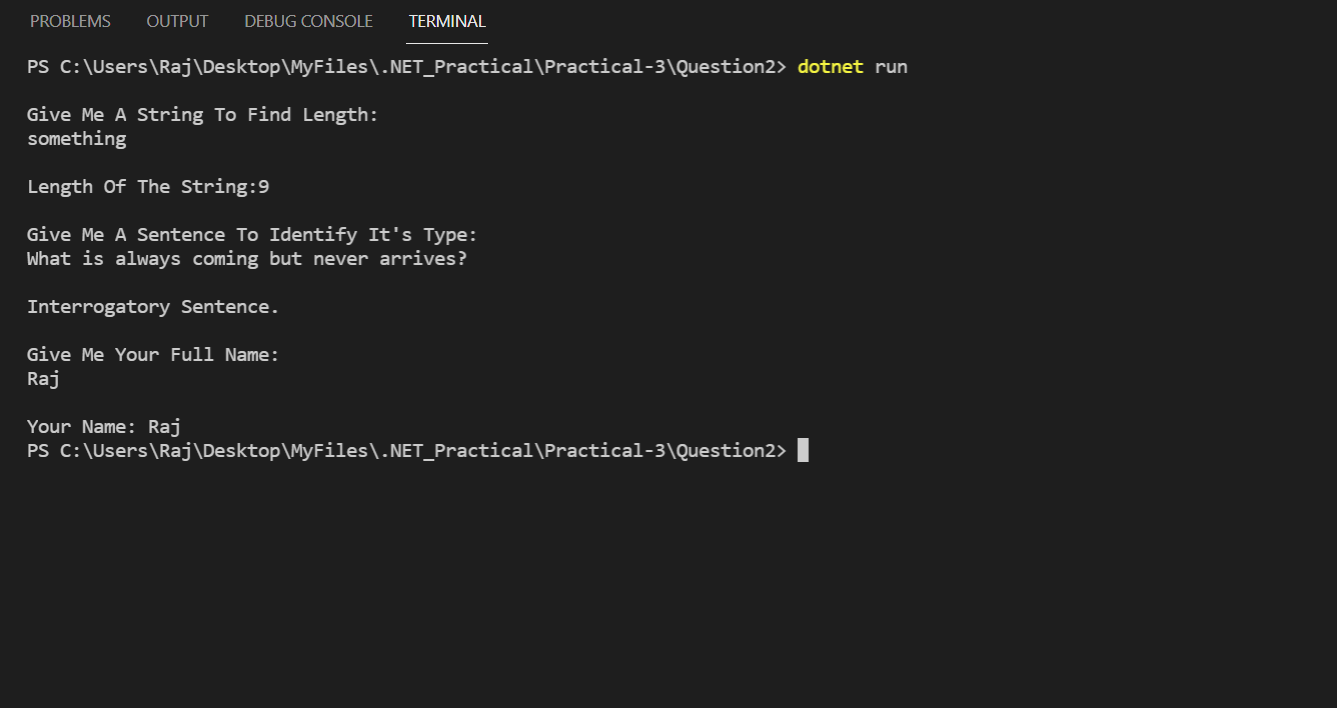
**Console.WriteLine("\nYour Name: "+splittedName[1]+", "+splittedName[0]);**

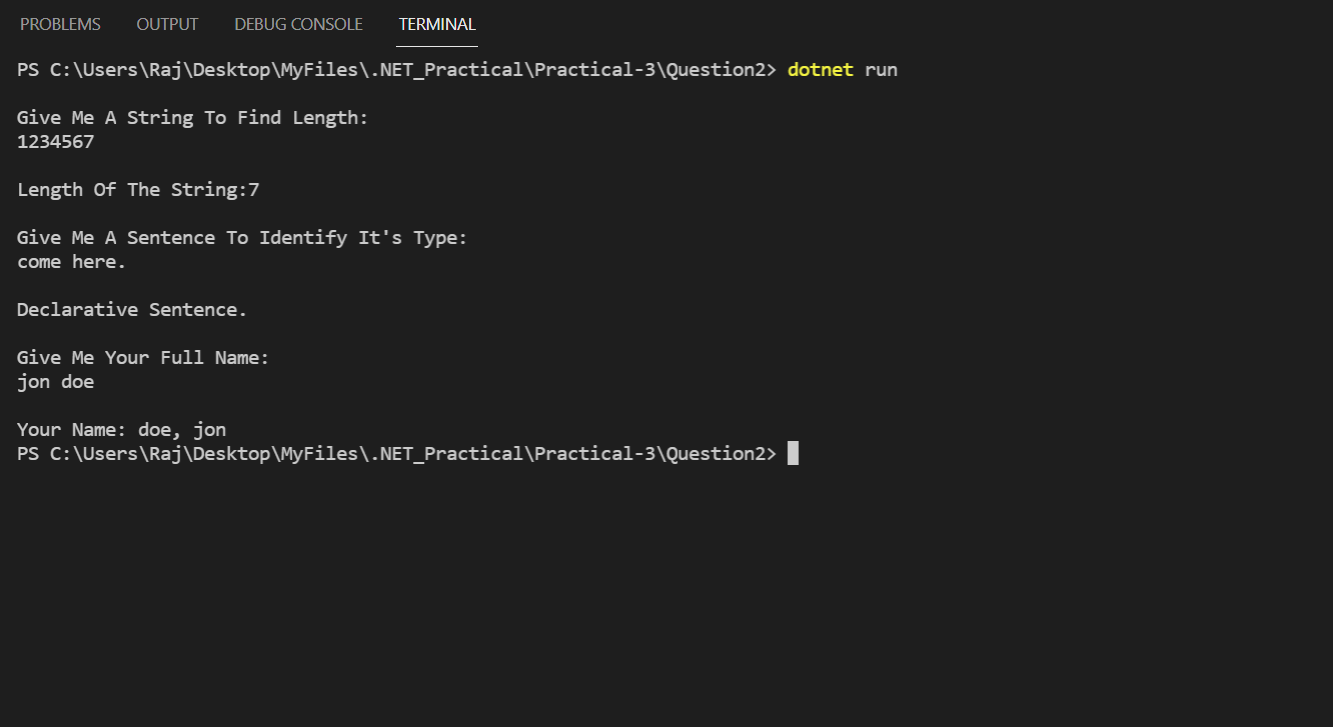
**}**

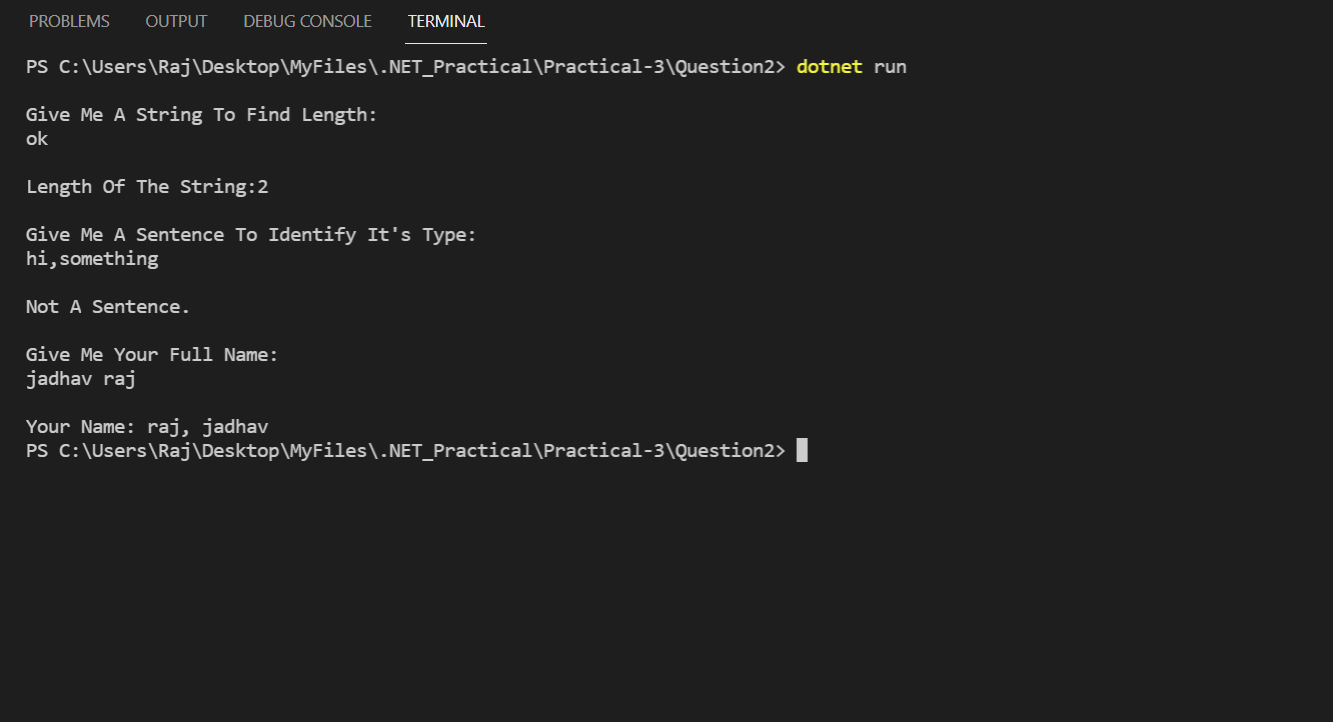
**}**

**OUTPUT:**

****

****

****

****

**QUESTION-3**

Enumeration Sample with bit flags

**CODE:**

**[Flags]**

**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**

**}**

**public class FlagsEnumExample**

**{**

**public static void Main()**

**{**

**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:**

**// Monday, Wednesday, Saturday**

**}**

**}**

**OUTPUT:**

