PRN:-2019033800120821

NAME:-PATEL PRIT SANJAYKUMAR

BATCH: - A

Roll no.: - 412052

Git repo: https://github.com/pritpatel179/assignment-3

**Assignment 3 .NET TECH ASSIGNMENT**

**QUESTION-1**

**CODE: Fraction.cs**

using System;

namespace ne

{

class Fraction

{

private int numerator,denominator;

public Fraction(int a,int b){ this.denominator = b; this.numerator = a; if(b==0){

throw new DivideByZeroException();

}

}

public double FloatingDivision(){

return ((float)numerator/denominator);

}

public void SimplifyFraction(){

int gcd = 0;

for(int i=1;i<denominator;i++){

if(numerator%i==0 && numerator%i == 0){

gcd = i;

}

}

Console.WriteLine($"Integer division result

= {numerator/gcd}/{denominator/gcd}");

}

public void IntegerDivison(){ Console.WriteLine($"Integer division result =

{(int)numerator/denominator} with a remainder

{numerator%denominator}");

}

public void ReduceFraction(){

int wholeNum = (int)numerator/denominator;

Console.WriteLine($"The result as a mixed fraction

is = {wholeNum} {numerator%denominator}/{denominator}");

}

}

}

**Program.cs:**

using System;

using ne;

namespace ne3

{

class Main\_Program

{

public static void Main(){

Console.WriteLine("Please enter the numerator?");

int numerator = Convert.ToInt32(Console.ReadLine()); Console.WriteLine("Please enter the denominator?");

int denominator = Convert.ToInt32(Console.ReadLine()); Fraction F = new Fraction(numerator,denominator);

Console.WriteLine($"Floating point division result =

{F.FloatingDivision()}");

F.IntegerDivison();

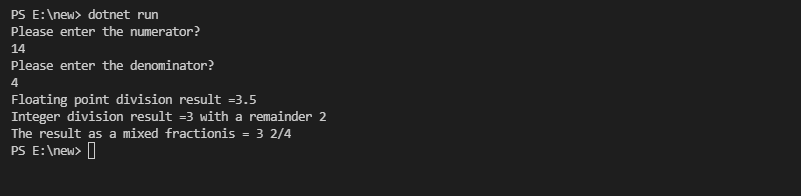
F.ReduceFraction();

}

}

}

**Output:**



**QUESTION-2**

**CODE:**

**string.cs**

using System;

namespace ne4{

class StringPro

{

public static int Length(string s){

return s.Length;

}

public static bool isDeclarative(string s){

return s[s.Length-1] == '.';

}

public static bool isInterrogative(string s){

return s[s.Length-1] == '?';

}

public static bool isExclamation(string s){

return s[s.Length-1] == '!';

}

public static void PrintName(string s){

string[] arr = s.Split(" "); Console.WriteLine($"You have entered name

\'{arr[1]},{arr[0]}\'");

}

public static void EnhancedPrintName(string s){

string[] arr = s.Split(" ");

if(arr.Length == 1){

Console.WriteLine($"You have entered name

\'{arr[0]}\'");

}else{

Console.WriteLine($"You have entered name

\'{arr[1]},{arr[0]}\'");

}

}

}

}

**Program1.cs:**

using System;

using ne4;

namespace ne5{

class MainProgram

{

public static void Main(string[] args){

//1

Console.WriteLine("\nEnter a string : "); String str = Console.ReadLine()??throw new

Exception();

Console.WriteLine($"\'{str}\' has length :

{StringPro.Length(str)}");

//2

Console.WriteLine("\nEnter a sentence : ");

String sentence = Console.ReadLine()??throw new

Exception();

if(StringPro.isDeclarative(sentence)){ Console.WriteLine("It is a declarative sentece");

}else if(StringPro.isExclamation(sentence)){

Console.WriteLine("It is a Exclamatory sentece");

}else if(StringPro.isInterrogative(sentence)){ Console.WriteLine("It is a interrogative sentece");

}

//3

Console.WriteLine("\nEnter a name, having first name and last name: ");

String name = Console.ReadLine()??throw new

Exception();

StringPro.PrintName(name);

//4

Console.WriteLine("\nEnter a name : ");

String sname = Console.ReadLine()??throw new

Exception();

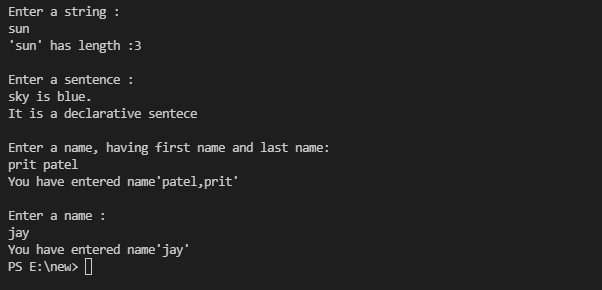
StringPro.EnhancedPrintName(sname);

}

}

}

**Output:**



**QUESTION-3**

**CODE: Program.cs:**

[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;

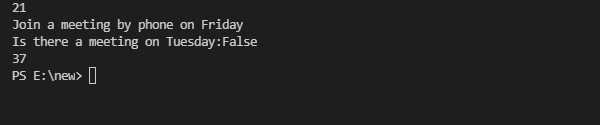
Console.WriteLine(a);

// Output:

// Monday, Wednesday, Saturday

}

}

**Output:** ****