Name: Session:

Programming II

Lab Exercise 4.11.2024

In this lab you will create applications to solve the following problems. For each application, turn in a screen shot of your running application.

1. How tall is the rainbow? Because of the way in which light is refracted by the water droplets, the angle between the level of your eye and the top of the rainbow is always the same (the magic angle). If you know the distance to the rainbow, you can multiply it by the tangent of the angle to find the height of the rainbow. The magic angle is 42.333333 degrees. The Visual Studio library tangent function works with radians and not degrees. You therefore have to convert degrees to radians with the following formula:

$$radians = degrees \times \frac{\pi}{180}$$

Where π = 3.14159265

Math.Tan is a value returning function that takes a floating point parameter and returns a floating point result:

$$x = Math.tan(someAngle)$$

If you multiply the tangent by the distance to the rainbow, you get the height of the rainbow.

Design and code an application to read a floating point value (the distance to the rainbow) and computes the height of the rainbow. The program should display the distance to the rainbow and its height.

We will write this program as a Console application.

```
double distance;
double height;
const double degrees = 42.333333;
double radians;
radians = degrees * Math.PI / 180;
Console.Write("Enter the distance in feet to the rainbow: ");
distance = Convert.ToDouble(Console.ReadLine());
height = distance * Math.Tan(radians);
Console.WriteLine("Height of rainbow = " + height.ToString("f2") + " feet");
```

2. The factorial of a number n is:

   n * (n-1) * (n-2) * … * 2 * 1

   only applies to integers and can get very large. Remember, integers are limited to roughly 4 billion. Stirling's Formula is a floating point approximation for large values of n.

   Factorial of $n = \frac{n^n\sqrt{2\pi n}}{e^n}$

   Where π = 3.14159265 and e is 2.718282 (or you could use Math library)

   Write a program that allows the user to enter an integer value and stores it in a double variable n and calculates the value of the factorial of n using Sterling's Formula. It should then display the result. What is the largest factorial you can calculate?

We will write this program as a Console application.

```
double n, f;
Console.Write("Enter the number you wish to factorialize: ");
n = Convert.ToDouble(Console.ReadLine());
f = (Math.Pow(n,n) * Math.Sqrt(2 * Math.PI * n)) / Math.Exp(n);
Console.WriteLine(n +"! = " + f);
```

Now re-write the program to find the largest number that can be calculated using Sterling's Equation.

```
double n, f;
for (n = 1; n < 200; n++)
{
        f = (Math.Pow(n, n) * Math.Sqrt(2 * Math.PI * n)) / Math.Exp(n);
        Console.WriteLine(n + "! = " + f);
}
```

3. Here are 32 football teams in the NFL (National Football League.) Some stores have vending machines that dispense miniature team helmets for one quarter (25 cents) each. When you put in your quarter, you never know which helmet you will get -- any one of the 32 team helmets is as likely as any other to be dispensed by the machines. They are given out at random.

   Write a program to simulate putting quarters in one of these machines until all 32 NFL HELMETS have been obtained. Your programs should print the following information:

   ```
   AMOUNT SPENT TO GET ALL 32 NFL HELMETS =
   ```

   Run your simulation program three times.

   Sample Run

   ```
   AMOUNT SPENT TO GET ALL 32 NFL HELMETS = $22.25
   AMOUNT SPENT TO GET ALL 32 NFL HELMETS = $32.00
   AMOUNT SPENT TO GET ALL 32 NFL HELMETS = $61.00
   ```

We will write this program as a Windows application

a. Create a global array to hold 32 integers

```
int[] helmets = new int[32];
```

b. Add the following code to the btnSim_Click event handler.

```
Random r = new Random();
int total = 0;
int rNumber = 0;
double amount = 0;

while (total != 32)
{
        amount += 0.25;
        rNumber = r.Next(0, 32);
        if (helmets[rNumber] == 0)
                helmets[rNumber]++;
        total = calcTotal();
}

lblOutput.Text = "AMOUNT SPENT TO GET ALL 32 HELMETS = " + amount.ToString("c");

for (int i = 0; i < helmets.Length; i++)
{
        helmets[i] = 0;
}
```

c. Add the following code to the calcTotal function.

```
int sum = 0;
for (int i = 0; i < helmets.Length; i++)
{
        sum += helmets[i];
}
return sum;
```

d. Now test your program to ensure it works correctly.

4. Write a Console Application that allows a user to enter a string and converts the string to backwards talk. The program should use a function to reverse each word. For example: if you enter the phrase “do you like snow’ it should convert it to “od uoy ekil wons” .