

Контест 9

11 янв 2022, 18:20:27

старт: 23 фев 2021, 12:30:00

финиш: 1 мар 2021, 23:59:59

длительность: 6д. 11ч.

начало: 23 фев 2021, 12:30:00

конец: 1 мар 2021, 23:59:59

Е. Зарплата в Зурумбии

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод
Вывод	стандартный вывод

В Зурумбии недавно встретили новый год и теперь настало время строить планы на новый только что начавшийся, каждый житель этой маленькой стороны зарабатывает сколько-то денег, скорее всего(в реалистичном случае), через год житель будет зарабатывать на 10% больше (это средний уровень ежегодной инфляции), в худшем случае, зарплата будет составлять только 60% от исходной, ну а в лучшем случае станет в два раза больше, чем была в начале года. Так как в рамках оптимизации образования жителей Зурумбии перестали учить математике, помогите каждому жителю узнать прогнозируемые значения зарплат в худшем, реалистичном и лучшем случае.

Формат ввода

Вводится число N – количество жителей, а затем реальные зарплаты каждого из жителей (целые положительные числа) по одному в новой строке. В случае некорректного ввода необходимо выбрасывать исключение типа `ArgumentException` с сообщением "Incorrect citizen".

Формат вывода

Выведите по 3 числа в каждой строке: возможные зарплаты для каждого жителя(в худшем, реалистичном и лучшем случае).

Пример 1

Ввод	<input type="text"/>	Вывод	<input type="text"/>
2		60.000 110.000 200.000	
100		120.000 220.000 400.000	
200			

Пример 2

Ввод	<input type="text"/>	Вывод	<input type="text"/>
1		600.000 1100.000 2000.000	
1000			

Примечания

В систему необходимо сдавать только файл `Citizen.cs`.

```
1 using System;
2
3 public class Citizen : IOptimist, IPessimist
4 {
5     int predictedValue;
6
7     public Citizen(int predictedValue)
8     {
9         this.predictedValue = predictedValue;
10    }
11
12    public double GetForecast()
13    {
14        return predictedValue * 1.1;
15    }
16
17    double IPessimist.GetForecast()
18    {
19        return predictedValue * 0.6;
20    }
21
22    double IOptimist.GetForecast()
23    {
24        return predictedValue * 2;
25    }
26
27    internal static Citizen Parse(string input)
28    {
29        if (!int.TryParse(input, out int value) || value <= 0)
30        {
31            throw new ArgumentException("Incorrect citizen");
32        }
33
34        return new Citizen(value);
35    }
36 }
```