

Контест 2

11 янв 2022, 16:49:39

старт: 22 сен 2020, 16:00:00

финиш: 28 сен 2020, 23:59:59

длительность: 6д. 7ч.

начало: 22 сен 2020, 16:00:00

конец: 28 сен 2020, 23:59:59

J. Робинзон

Ограничение времени	10 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В 18 веке одинокий Робинзон вырисовывал на камнях чёрточки, чтобы понимать, сколько дней он провёл на острове. Но невнимательность подвела его, и он забыл указать день недели, когда начинал отсчёт. Это его жутко огорчило, ведь теперь он не знает, когда наступит его любимый день недели – пятница. К счастью, Робинзон помнит дату первой чёрточки, и теперь ему необходимо посчитать дату следующей пятницы после той даты, которую он помнит. Помогите Робинзону это сделать.

1. Каледарь 18 века - <http://kalendata.ru/>;
2. Как правильно вычислить день недели по дате - <https://habr.com/ru/post/217389/>.

Формат ввода

Вводятся три целых числа – день, месяц и год, каждое с новой строки.

Формат вывода

Программа должна вывести одну строку – дату следующей после введённой даты пятницы в формате "день.месяц.год". Если введённая дата уже является пятницей, требуется вывести её же. Если входные данные некорректны, программа должна вывести сообщение "Incorrect input" и ничего более.

Пример 1

Ввод

```
15
01
1711
```

Вывод

```
16.01.1711
```

Пример 2

Ввод

```
20
09
2020
```

Вывод

```
Incorrect input
```

Примечания

Входные данные считаются некорректными, если:

1. Введённая дата не относится к 18 веку, то есть не лежит в диапазоне $[01.01.1701; 31.12.1800]$;
2. Введённая дата некорректна или не существует. Например, 32 июля 2020 года.

В этой задаче в систему надо сдавать только файл `Program.Friday.cs` или его содержимое.

[Набрать здесь](#)[Отправить файл](#)

```
1 using System;
2
3 partial class Program
4 {
5     private static bool ValidateData(int day, int month, int year)
6     {
7         if (!(year >= MinYear && year <= MaxYear))
8         {
9             return false;
10        }
11
12        if ((year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0))
13        {
14            if ((month == 01 || month == 03 || month == 05 || month == 07 || month == 08 || month == 10 || month == 12) && (day
15            {
16                return true;
17            }
18
19            if ((month == 04 || month == 06 || month == 09 || month == 11) && (day >= 1 && day <= 30))
20            {
21                return true;
22            }
23
24            if (month == 02 && (day >= 1 && day <= 29))
25            {
26                return true;
27            }
28        }
29        else
30        {
31            if ((month == 01 || month == 03 || month == 05 || month == 07 || month == 08 || month == 10 || month == 12) && (day
32            {
33                return true;
34            }
35
36            if ((month == 04 || month == 06 || month == 09 || month == 11) && (day >= 1 && day <= 30))
37            {
38                return true;
39            }
40        }
41    }
42 }
```

[Отправить](#)[Предыдущая](#)