OLIMPIADA LA INFORMATICĂ

etapa raională/municipală, 13.02.2022, Clasa a 10-a

Timp de lucru: 240 de minute *Mult succes!*

Общее описание задач

№	Название задачи	Имя исходного файла	Ограничение на объем используемой памяти	Ограничение на время выполнения программы	Число тестов	Количество баллов, присвоенных одному тесту	Количество баллов, присвоенных задаче
1.	Domino Домино	`domino.pas`, `domino.c` или `domino.cpp`	≤ 8 Mb	≤ 1,0 s	20	10	200
2.	Energie electrică Электрическая энергия	`energie.pas`, `energie.c` или `energie.cpp`	≤ 8 Mb	≤ 0,1 s	10	10	100
3.	Numere romane Римские числа	`numere.pas`, `numere.c` или `numere.cpp`	≤ 8 Mb	≤ 0,1 s	5	100	100

Всего: 3 задачи, 400 баллов

Примечание. При равенстве общего числа баллов, лучшей будет считаться решение с меньшим временем выполнения.
В случае равенства и времени выполнения, лучшей будет считаться решение, использующее меньший объем памяти.

Domino / Домино

В игре Домино используют 28 различающихся костяшек, представляющих собой прямоугольные пластины. Каждая костяшка идентифицируется двумя числами, нанесенные на пластины с помощью точек (*рис. 1*), причем каждое число принадлежит множеству {0, 1, 2, 3, 4, 5, 6}. Маркировка указанных чисел осуществляется путем разделения поверхности каждой костяшки на две одинаковые области.

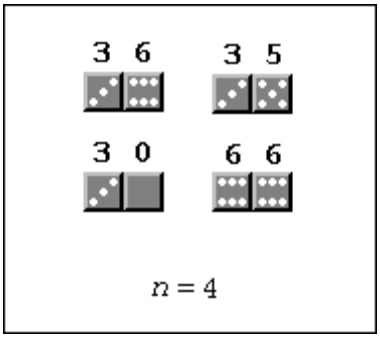

Рис. 1

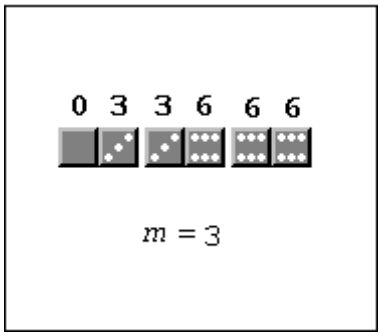

Рис. 2

Рассматривая костяшки в качестве воображаемых вагонов, из *n* различных костяшек можно построить „железнодорожный поезд” со следующим свойством: соседние числа любых соприкасающихся костяшек равны между собой (*рис. 2*). Длина поезда определяется количеством костяшек *m* из которого он состоит. Очевидно, $m \le n$.

Напишите программу которая определяет количество костяшек в „поезде” максимальной длины, который можно построить из *n* заданных различающихся костяшек.

Входные данные. Текстовый файл `domino.in` содержит в первой строке количество костяшек *n*. Следующие *n* строк входного файла содержат по два целых числа, разделенных пробелом, причем каждая из этих строк описывает отдельную костяшку.

Выходные данные. Текстовый файл `domino.out` должен содержать в единственной строке количество костяшек *m* „поезда” максимальной длины.

Пример.

`domino.in`

```
4
3 6
3 5
3 0
6 6
```

`domino.out`

```
3
```

Ограничения. Для первых 10 тестов $1 \le n \le 10$. Для последующих 10 тестов $11 \le n \le 28$. Ограничения на время выполнения и на объем используемой памяти приведены на странице «Общее описание задач». Исходный файл должен иметь имя `domino.pas`, `domino.c` или `domino.cpp`.

Energie electrică / Электроэнергия

Все мы хотим жить на зеленной планете. Для этого очень важно экономить электроэнергию.

Рассматриваются абоненты, подключенные к электрической сети определенного населенного пункта. Для каждого из этих абонентов известны фамилия, имя и годовое потребление электроэнергии, выраженное в киловатт-часах.

Задание. Разработайте программу, которая:

1) вычисляет среднегодовое потребление электроэнергии на одного абонента;

2) определяет абонента с максимальным годовым потреблением электроэнергии;

3) определяет абонента с минимальным годовым потреблением электроэнергии.

В качестве учебного примера, ниже представлена таблица с информацией об абонентах некоторого населенного пункта.

Фамилия, имя абонента	**Годовое потребление электроэнергии, киловатт-часы**
Munteanu Ion	25
Prisăcaru Viorica	59
Poștașul Valentina	38

Из вышеприведенной таблицы следует что абонентом с максимальным потреблением электроэнергии является *Prisăcaru Viorica*, а абонентом с минимальным потреблением – *Munteanu Ion*. Среднегодовое потребление электроэнергии на одного абонента вычисляется следующим образом:

$$(26 + 59 + 38) : 3 = 40.6666666666667$$ киловатт-часов.

Входные данные. Текстовый файл `energie.in` содержит в первой строке целое число – число абонентов. Каждая из следующих строк входного файла содержит строку символов: фамилию, имя и годовое потребление абонента, разделенные пробелами. Другими словами, информация об каждом абоненте структурирована следующим образом:

<Фамилия> <Пробел> <Имя> <Пробел> <Целое число>

В этом описании *<Фамилия>* и *<Имя>* представляют собой символьные строки без пробелов, а *<Целое число>* – годовое потребление электроэнергии соответствующего абонента.

Выходные данные. Текстовый файл `energie.out` должен содержать в первой строке вещественное число – среднегодовое потребление электроэнергии на одного абонента. Вещественное число должно быть записано на 10 позициях, с выравниванием по правому краю, без масштабного множителя, с двумя знаками после десятичной точки.

Вторая строка выходного файла должна содержать информацию об абоненте с максимальным потреблением электроэнергии, а третья строка - информацию об абоненте с

минимальным потреблением электроэнергии. Информация во второй и третьей строках выходного файла будет структурирована точно так же, как и во входном файле.

Ограничения. Входной файл содержит не более 10 абонентов. Среднегодовое потребление каждого из абонентов не превышает 4000 киловатт-часов, во входном файле нет абонентов с одинаковым годовым потреблением. Ограничения на время выполнения и на объем используемой памяти приведены на странице «Общее описание задач». Исходный файл должен иметь имя `energie.pas`, `energie.c` или `energie.cpp`.

Пример.

`energie.in`

```
3
Munteanu Ion 25
Prisacaru Viorica 59
Postasul Valentina 38
```

`energie.out`

```
      46.7
Prisacaru Viorica 59
Munteanu Ion 25
```

Numere romane / Римские числа

В римской системе счисления используется 7 цифр, представленные следующими буквами латинского алфавита: `I`, `V`, `X`, `L`, `C`, `D` и `M`. В десятичной системе рассматриваемые цифры имеют следующее значения:

Цифра в римской системе счисления	**Значение римской цифры в десятичной системе счисления**
`I`	1
`V`	5
`X`	10
`L`	50
`C`	100
`D`	500
`M`	1000

Напоминаем, что римская система счисления является непозиционной. Число, записанное в римской системе, читается слева направо. Десятичное значение числа определяется путем суммирования отдельных десятичных значений каждой из цифр, если только текущая цифра не предшествует цифре с более высоким значением. В этом случае десятичное значение текущей цифры вычитается из десятичного значения следующей цифры.

Примеры:

Число в римской системе счисления	**Число в десятичной системе счисления**
`X`	10
`IX`	9
`XIII`	13
`XIV`	14
`MMCIX`	2109
`MMXXII`	2022

Задание. Напишите программу для перевода римских чисел в их значения в десятичной системе счисления.

Входные данные. Текстовый файл `numere.in` содержит в первой строке число *n* римских чисел из входного файла. Каждая из следующих *n* строк входного файла содержит по одному римскому числу.

Выходные данные. Текстовый файл `numere.out` должен содержать в первой строке целое число *n*. Каждая из следующих *n* строк выходного файла должно содержать по одному целому числу – десятичное значение римского числа из соответствующей строки входного файла.

Ограничения. $1 \leq n \leq 10$. Каждое из римских чисел из входного файла содержит не более 255 римских цифр. Ограничения на время выполнения и на объем используемой памяти приведены на странице «Общее описание задач». Исходный файл должен иметь имя `numere.pas`, `numere.c` или `numere.cpp`.

Пример.

`numere.in`

```
6
X
IX
XIII
XIV
MMCIX
MMXXII
```

`numere.out`

```
6
10
9
13
14
2109
2022
```