

**XXV ОЛИМПИАДА ШКОЛЬНИКОВ КРАСНОЯРСКОГО КРАЯ ПО ИНФОРМАТИКЕ
2011/2012 УЧЕБНЫЙ ГОД**

II этап – муниципальный
(основной тур, 29.11.2011 г.)

ЗАДАЧИ

Максимальное время выполнения заданий: 240 мин.

Максимальное количество набранных баллов: 500

Задача А. Колокол

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 Мб

Максимальный балл: 100

Требуется написать программу, которая в массиве из n целых чисел наименьший элемент поместит на первое место, наименьший из оставшихся – на последнее, следующий по величине – на второе место, следующий – на предпоследнее и так далее – до середины массива.

Входные данные

Во входном файле INPUT.TXT записано в первой строке число n ($1 \leq n \leq 30000$). Во второй строке записаны через пробел элементы массива, числа по абсолютной величине не больше 32767.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести элементы полученного массива, разделенные одним пробелом.

Пример

INPUT.TXT	OUTPUT.TXT
5 1 2 3 4 5	1 3 5 4 2

Задача В. Пятнашки

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 Мб

Максимальный балл: 100

Пятнашки – популярная головоломка, представляющая собой набор одинаковых квадратных костяшек с нанесёнными числами, заключённых в квадратную коробку, имеющей размер 4×4 . Цель игры — перемещая костяшки по коробке добиться упорядочивания их по номерам (как показано на рисунке), желательно сделав как можно меньше перемещений. Известно, что не любое размещение костяшек на доске позволяет получить решаемую задачу.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Рассмотрим более общую игру для доски $N \times N$, где будет использоваться $N^2 - 1$ костяшек с числами. Самый надёжный способ получить решаемую головоломку – это провести последовательность произвольных ходов из конечного решенного состояния. Такой набор действий удобно представить в виде последовательности символов, обозначающих направления движения пустого места на доске. Пусть «U», «D», «L» и «R» – возможные направления движения, обозначающие «вверх», «вниз», «влево» и «вправо» соответственно. Игровую коробку удобно представить матрицей, а костяшки – числами. Пустое место будем обозначать цифрой «0».

Например, для $N=3$ первоначально мы будем иметь следующую доску:

```
1 2 3
4 5 6
7 8 0
```

После команды «ULD» мы получим следующее состояние:

```
1 2 3
4 8 5
7 0 6
```

Заметим, что команда «URLD» невыполнима в связи с невозможностью на втором ходе передвинуть пустое поле вправо.

По заданному размеру поля и последовательности команд требуется определить конечное состояние игрового поля.

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число N – размерность игрового поля ($N \leq 20$). Во второй строке располагается последовательность команд (не более 10^4 действий), содержащая только символы «U», «D», «L» и «R», записанные слитно.

Выходные данные

В выходной файл OUTPUT.TXT выведите таблицу конечного состояния игрового поля. В том случае, когда команда не выполнима, в выходной файл следует вывести только текст «ERROR K», где K – номер хода, на котором

произошла ошибка. При выводе допускается использование избыточных пробелов и переносов строк.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	3 ULDLURULD	8 1 3 0 2 5 4 7 6
2	2 URL	ERROR 2

Задача С. Манхэттенские улицы

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 Мб

Максимальный балл: 100

Система улиц Нью-Йоркского района Манхеттен весьма интересна. В Манхеттене есть n улиц, идущие с запада на восток (авеню), и m улиц, идущие с севера на юг (просто улицы). Ширина каждого авеню и каждой улицы равна d метров, а длина – 1 метр. При этом каждая улица пересекает каждый авеню и не имеет общих точек с другими улицами, а каждый авеню пересекает каждую улицу и не имеет общих точек с другими авеню.

Разумеется, все авеню и улицы имеют асфальтовое покрытие. Дорожно-ремонтные службы интересуются: сколько квадратных метров асфальта уложено на все авеню и улицы. На перекрестках, без сомнения, асфальт уложен в один слой.

Напишите программу, вычисляющую ответ на их вопрос.

Входные данные

Входной файл INPUT.TXT содержит четыре натуральных числа n, m, d, l ($1 \leq n, m, d, l \leq 10^9, l > m \cdot d, l > n \cdot d$).

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	1 1 5 10	75
2	100 10 23 4560	11007800

Задача D. ДНК

Ограничение по времени: 2 секунды

Ограничение по памяти: 32 Мб

Максимальный балл: 100

Вася никогда не любил биологию. Но когда он узнал про ДНК, у него появился живой интерес. Он решил, что если все существа произошли друг от друга, то и ДНК у них должны быть похожими. У некоторых более похожие, у некоторых – менее, но у всех ДНК можно записать в виде строки, состоящей из символов *A*, *C*, *G* и *T*. Поэтому он решил найти какой-нибудь показатель родства. И придумал следующее. Он берет из двух ДНК по подстроке. Если одна из них является анаграммой другой (т. е. получается перестановкой букв), то это хорошая пара подстрок. Естественно, в любой хорошей паре обе подстроки имеют одинаковую длину. Тогда степень родства двух ДНК – это максимально возможная длина подстрок в хорошей паре.

Входные данные

В первой строке входного файла INPUT.TXT находится ДНК Васи. А во второй строке – ДНК первого попавшегося Васе живого существа. Обе строки не пусты и состоят не более, чем из 1 300 символов *A*, *C*, *G* и *T*.

Выходные данные

В первую строку выходного файла OUTPUT.TXT выведите степень родства Васи с подопытным существом. Если степень родства отлична от нуля, то во вторую следует вывести две начальные позиции подстрок из соответствующей хорошей пары в первой и второй ДНК соответственно. В случае неоднозначности последних двух чисел, выведите любые подходящие.

Примеры

№	INPUT.TXT	OUTPUT.TXT
1	ACGT GTC	3 2 1
2	ACGT CAT	2 1 1
3	ACA AC	2 2 1

Задача Е. Мусорщик

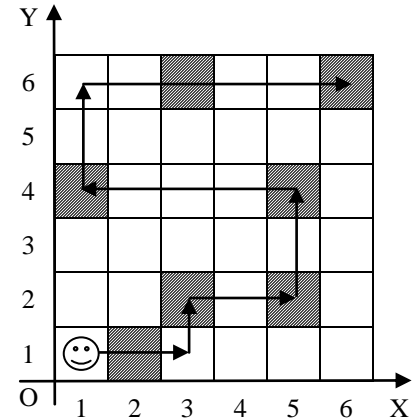
Ограничение по времени: 1 секунда

Ограничение по памяти: 16 Мб

Максимальный балл: 100

Всем известно, что работа уборщицы становится все менее престижной, чем работа программиста. Поэтому, все больше становится программистов и все меньше уборщиц. И скоро, возможно, совсем некому будет делать уборку помещений, а чистота – она всегда актуальна и важна для работы, например, для тех же программистов.

Сотрудники одной из фирм разработали специальную машину «Мусорщик-001», которая предназначена для уборки прямоугольных пустых помещений. Машина не совершенна и может пока двигаться на 1 метр только влево, вправо и вперед (вдоль оси OY). Каждое помещение можно разбить на квадратные сектора со стороной в 1 метр и обозначить те, которые загрязнены. Для уборки помещения достаточно, чтобы машина-уборщик побывала в каждом из загрязненных секторов. Известно, что перед уборкой машина всегда находится в клетке (1,1).



Одна из компаний, где в штате нет уборщицы, но имеется полный штат программистов, приобрела «Мусорщик-001». Пока программистам никак не удастся написать программу, определяющую по заданному плану загрязнения помещения минимально возможную длину маршрута машины-уборщика, необходимого для уборки данной территории. Возможно, Вам удастся им помочь!

Входные данные

Первая строка входного файла INPUT.TXT содержит натуральное число n ($n \leq 1000$). Следующие n строк содержат по два натуральных числа: x_i и y_i – координаты загрязненных секторов в заданном помещении ($x_i, y_i \leq 50$).

Выходные данные

В выходной файл OUTPUT.TXT выведите целое число, соответствующее минимальной длине маршрута в метрах, необходимого для уборки.

Пример

INPUT.TXT	OUTPUT.TXT
7 2 1 3 2 5 2 5 4 1 4 3 6 6 6	18