



## II (муниципальный) этап Всероссийской олимпиады школьников 2013/2014 уч. год. Новосибирская область

Предмет	Класс	Дата	Время начала	Время окончания
<b>Информатика</b>	<b>9-11</b>	<b>29.11.2013</b>	<b>10:00</b>	<b>15:00</b>

### Для всех задач:

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда на тест
Ограничение по памяти:	256 Мб

### Задача 1. Треугольник силы

На территории древней культуры Алтая проводилась значительная археологическая работа. Было найдено большое количество артефактов в тех местах, которые были названы полями духов: это небольшая область, меньше чем 100 квадратных километров, где имеются несколько высоких гор с кристаллами на вершинах. Большинство артефактов было обнаружено внутри треугольной области между тремя из таких гор. Эта область была названа треугольником силы. После совместного анализа археологи поняли, как выбрать этот треугольник из всех треугольников с тремя горами в качестве вершин: это треугольник с наибольшей площадью, не содержащий других гор внутри себя или на своей границе. Каждая область содержит только один такой треугольник.

Археологические бригады продолжают поиски новых полей духов. Они бы хотели автоматизировать задачу поиска треугольников силы в полях духов.

Напишите программу, которая по положению гор в поле духов определяет треугольник силы для каждого такого поля.

#### Входные данные

Первая строка входного файла содержит целое число  $N$  — количество гор с кристаллами на вершинах ( $4 \leq N \leq 15$ ).

В следующих  $N$  строках описываются горы с кристаллами, по одной на строке. Строка, содержащая описание одной горы, начинается с символьной метки (один символ), затем даны целочисленные неотрицательные координаты горы, каждая из которых меньше 100. В первой строке записана метка **A**, во второй — **B**, и т.д.

#### Выходные данные

Выходной файл должен состоять из одной строки. В этой строке должны быть записаны три символьных метки вершин треугольника силы, перечисленные в алфавитном порядке, без пробелов.

#### Пример

<i>input.txt</i>	<i>output.txt</i>
6 A 1 0 B 4 0 C 0 3 D 1 3 E 4 4 F 0 6	BEF
4 A 0 0 B 1 0 C 99 0 D 99 99	BCD



## II (муниципальный) этап Всероссийской олимпиады школьников 2013/2014 уч. год. Новосибирская область

Предмет	Класс	Дата	Время начала	Время окончания
<b>Информатика</b>	<b>9-11</b>	<b>29.11.2013</b>	<b>10:00</b>	<b>15:00</b>

### Задача 2. Ремонт на железной дороге

Как известно, железнодорожные рельсы лежат на шпалах. Со временем шпалы приходят в негодность, поэтому для обеспечения безопасности перевозок старые шпалы необходимо заменять. Производить замену железнодорожного пути целиком очень дорого, тем более что шпалы выходят из строя не все сразу. Поэтому имеется специальная контролирующая машина, которая идет по рельсам и проверяет качество каждой шпалы. Эта машина отмечает каждую плохую шпалу специальным маркером и заносит ее номер в реестр. Все шпалы пронумерованы, начиная с 1. Маленький паровозик развозит новые шпалы к месту ремонта. После этого дорожные рабочие заменяют отмеченные шпалы. И здесь возникает проблема организации максимально эффективной доставки шпал.

Новые шпалы упакованы в пакеты по  $M$  штук в каждом. При доставке на паровозике весь пакет сгружается целиком, а затем шпалы от места разгрузки развозятся поштучно к месту установки. Стоимость доставки одной новой шпалы от позиции с номером  $i$  к позиции с номером  $k$  прямо пропорциональна расстоянию между позициями. Шпалы лежат на равном расстоянии одна от другой.

Ваша задача — написать программу, которая вычислит наиболее дешевый вариант доставки шпал к месту замены. Для этого требуется определить необходимое количество пакетов шпал, место выгрузки каждого пакета и участок, на котором этот пакет будет использоваться. При этом следует учитывать, что суммарное количество оставшихся после ремонта неиспользованных шпал не должно превышать  $M - 1$ . Стоимость доставки пакета шпал для любой точки пути не учитывается.

#### Входные данные

В первой строке входного файла указаны два целых числа  $N$  и  $M$  ( $1 \leq N \leq 3000$ ,  $1 \leq M \leq 200$ ) — количество шпал, которые подлежат замене, и количество шпал в пакете, соответственно. Следующая строка содержит  $N$  различных целых чисел, записанных через пробел. Это номера шпал, подлежащих замене. Значения номеров не превышают 30000.

#### Выходные данные

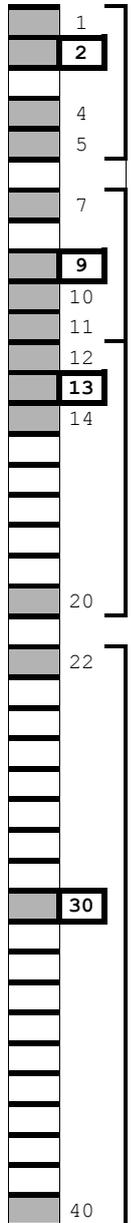
В выходной файл вы должны вывести столько строк, сколько пакетов шпал предусматривается использовать для ремонта железнодорожного пути, по одной строке на пакет. В начале строки указывается номер шпалы, у которой нужно сгрузить очередной пакет. А затем после двоеточия через пробел указывается участок для ремонта, соответствующий этому пакету, в форме "from  $i$  to  $j$ ", где  $i$  и  $j$  — начальный и конечный номера шпал на участке. Если существует несколько решений, вы должны вывести любое из них.

#### Пример

<i>input.txt</i>	<i>output.txt</i>
15 4 1 10 11 12 13 14 2 20 22 30 4 40 5 7 9	2: from 1 to 5 9: from 7 to 11 13: from 12 to 20 30: from 22 to 40

#### Комментарий

На рисунке серым цветом выделены неисправные шпалы и рядом указаны их номера. Квадратиками обозначены номера шпал, около которых нужно выгружать пакеты. Квадратные скобки обозначают участки ремонта для каждого пакета.





## II (муниципальный) этап Всероссийской олимпиады школьников 2013/2014 уч. год. Новосибирская область

Предмет	Класс	Дата	Время начала	Время окончания
Информатика	9-11	29.11.2013	10:00	15:00

### Задача 3. Перекрёсток

В недалеком будущем проблема пробок и аварий на дорогах полностью решена компанией Кугл с помощью беспилотных автомобилей. Такие автомобили едут, соблюдая все правила и с постоянной скоростью.

К сожалению, в одной из стран в правилах дорожного движения обнаружилась одна неразрешимая ситуация. Если четыре машины подъезжают к перекрестку в один и тот же момент с разных сторон, и всем необходимо ехать вперед, то каждая машина обязана пропустить машину справа, в итоге все могут ожидать вечно.

Временно эта проблема решена с помощью вмешательства человека.

Имеется карта города, на которой изображены все перекрестки. Для каждого перекрестка отмечено, с какого перекрестка на него можно въехать. Следует отметить, что на любой перекресток можно приехать не более чем с четырех других различных перекрестков, которые будем называть соседями данного перекрестка слева, справа, снизу и сверху. В городе невозможно вернуться на перекресток, выехав с него, не проезжая через другие перекрестки. Если один перекресток является для другого соседом снизу, то это не гарантирует, что тот, в свою очередь, будет для данного перекрестка соседом сверху.

По заданной карте и маршрутам необходимо определить, возникнет ли неразрешимая ситуация на дороге, и если да, то в каком месте.

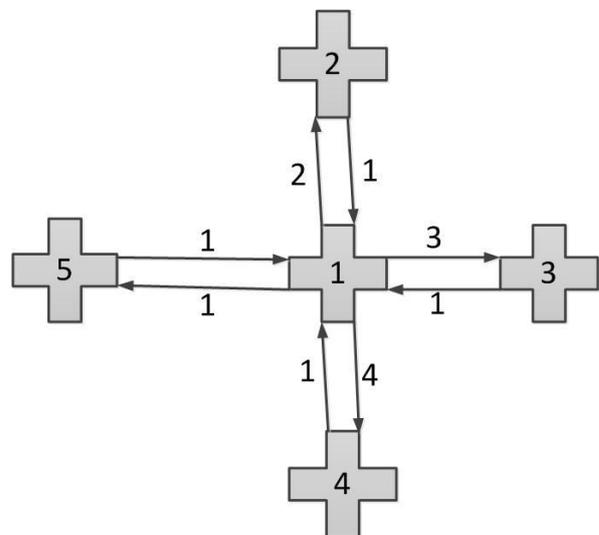
#### Входные данные

В первой строке входного файла задано три целых числа  $N$ ,  $M$  и  $V$  соответственно, число перекрестков, машин, скорость для всех машин ( $1 \leq N < 500$ ,  $4 \leq M < 5000$ ,  $1 \leq V < 100$ ).

Далее в  $N$  строках дано описание перекрестков. Все перекрестки пронумерованы числами от 1 до  $N$ . Описание перекрестков дано в порядке возрастания их номеров. Каждый перекресток описан на отдельной строке, содержащей четыре пары целых чисел, соответствующие описанию его соседей. Соседние перекрестки заданы в следующем порядке: сверху, справа, снизу и слева. Первое число в паре задает номер перекрестка, а второе — расстояние от него до описываемого (натуральное число от 1 до 100). Если у перекрестка нет соответствующего соседа, то записана пара  $-1$   $-1$ .

Далее в  $M$  строках дано описание маршрута каждой машины. Машины пронумерованы числами от 1 до  $M$ . Описание маршрутов машин дано в порядке возрастания их номеров, по одному на строке. Первым в строке записано целое число — время старта машины  $T$  ( $0 \leq T \leq 10^6$ ), вторым записано натуральное число — количество перекрестков на маршруте, а далее через пробел перечисляются номера перекрестков данного маршрута. Количество перекрестков на маршруте не превосходит 100. Машины следуют строго по заданному маршруту.

Все машины начинают путь одновременно, перекрестки, для которых проезд определен, они проезжают мгновенно — умная система позволяет избежать пробок. Первый перекресток на маршруте считается уже пересеченным, а последний пересекать не надо.



#### Выходные данные



## II (муниципальный) этап Всероссийской олимпиады школьников 2013/2014 уч. год. Новосибирская область

Предмет	Класс	Дата	Время начала	Время окончания
<b>Информатика</b>	<b>9-11</b>	<b>29.11.2013</b>	<b>10:00</b>	<b>15:00</b>

В выходной файл необходимо вывести целое число — номер перекрестка, на котором образуется неразрешимая ситуация. При нескольких вариантах нужно выбрать вариант с наименьшим временем, при нескольких совпадающих — с наименьшим номером перекрестка.

Если неразрешимая ситуация не встречается, то необходимо вывести число **-1**.

### Примеры

<i>input.txt</i>	<i>output.txt</i>
5 4 1 2 3 3 2 4 2 5 3 -1 -1 -1 -1 1 3 -1 -1 -1 -1 -1 -1 -1 -1 1 3 1 2 -1 -1 -1 -1 5 1 -1 -1 1 3 4 1 -1 -1 0 4 5 4 1 2 1 3 3 1 5 0 3 5 1 4 0 3 5 1 3	-1
5 4 100 2 1 3 1 4 1 5 1 -1 -1 -1 -1 1 2 -1 -1 -1 -1 -1 -1 -1 -1 1 3 1 4 -1 -1 -1 -1 -1 -1 -1 -1 1 1 -1 -1 -1 -1 0 3 2 1 4 0 3 4 1 2 0 3 3 1 5 0 3 5 1 3	1

### Задача 4. Числа

Дано  $N$  цифр восьмеричной системы счисления.

Требуется посчитать, сколько различных  $N$ -значных чисел (в восьмеричной системе) можно составить из этих цифр, таких, чтобы значение каждого полученного числа не превышало  $2^{32}-1$ . В числах допускается наличие ведущих нулей.

#### Входные данные

Во входном файле записана непустая строка, содержащая цифры, каждая из которых может принимать значения от 0 до 7 включительно. Длина строки не превосходит 11.

#### Выходные данные

В выходной файл необходимо вывести одно целое число — количество возможных восьмеричных чисел, удовлетворяющих условию задачи.

#### Пример

<i>input.txt</i>	<i>output.txt</i>
61	2



## II (муниципальный) этап Всероссийской олимпиады школьников 2013/2014 уч. год. Новосибирская область

Предмет	Класс	Дата	Время начала	Время окончания
Информатика	9-11	29.11.2013	10:00	15:00

### Задача 5. Остановки

В некотором городе вдоль длинной улицы запустили новый маршрут автобуса. По одной стороне улицы от начальной точки маршрута до конечной автобус делает  $K$  остановок, в обратном направлении для него предусмотрено  $M$  остановок. Автобус начинает маршрут от начала левой стороны улицы, в конце улицы разворачивается и следует в обратном направлении по правой стороне улицы.

При выборе названий для остановок было решено парные остановки на разных сторонах улицы назвать одинаково. Две остановки, находящиеся на разных сторонах улицы, считаются парными, если они находятся друг от друга не более чем на расстоянии  $D$ . Известно, что на одной и той же стороне улицы остановки располагаются на расстоянии более чем  $2 \cdot D$ .

Необходимо написать программу, которая определяет парные остановки.

#### Входные данные

В первой строке входного файла записано четыре целых числа  $K$ ,  $M$ ,  $L$  и  $D$ , количество остановок на каждой стороне улицы, длина улицы и разрешенная удаленность остановок друг от друга ( $0 < K, M \leq 10^6$ ,  $1 \leq L \leq 10^9$ ,  $0 \leq D \leq 10$ ).

Во второй строке записано через пробел  $K$  целых чисел — расстояния от начала маршрута до остановок по левой стороне улицы, расстояния перечисляются в порядке возрастания. В третьей строке записано  $M$  целых чисел через пробел — расстояния от места разворота автобуса до остановок по правой стороне улицы, расстояния перечисляются в порядке возрастания. Остановки нумеруются в порядке перечисления в каждой строке, начиная с 1.

#### Выходные данные

В выходной файл необходимо вывести номера парных остановок, по два числа на строке, сначала номер остановки на левой стороне, потом через пробел номер остановки по правой стороне.

Если парных остановок не существует, то необходимо вывести 0.

#### Примеры

<i>input.txt</i>	<i>output.txt</i>
3 3 8 1 0 5 8 1 5 8	1 3 3 1
2 1 5 0 1 5 3	0