



### Для всех задач:

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по памяти: `512 Мб`

## Задача 1 . Заполнение листа

Ограничение по времени:

1 секунда на тест

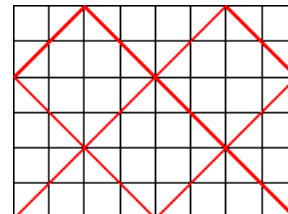
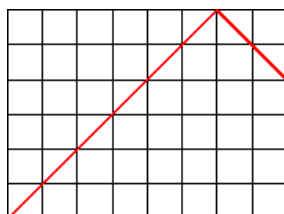
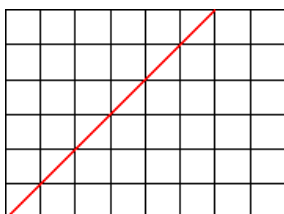
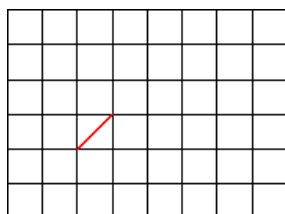
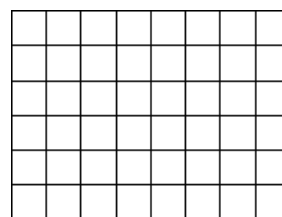
У Пети есть очень много листов бумаги. Все листы разлинованы в клеточку, ровно по  $N \times M$  клеток каждый.

Заполнением листа Петя называет следующую последовательность действий: сначала он выбирает одну из клеток и проводит в ней отрезок, совпадающий с одной из её диагоналей.

Затем он продолжает проводить этот отрезок в обе стороны до краёв листа. Если проводимый отрезок попадает в угол, далее в эту сторону он не проводится.

Если же отрезок попадает на сторону листа, то Петя начинает проводить его «отражение», то есть перпендикулярный попавшему на сторону листа отрезку, пересекающийся с ним в той точке, где он достиг края листа.

Подобным образом процесс заполнения продолжается до тех пор, пока имеется возможность его продолжить. Если изображение в какой-то момент перестает изменяться, то процесс останавливается



Петя заметил, что если выбирать разные начальные позиции (то есть клетку и диагональ), то в результате заполнения листа могут получаться как одинаковые изображения, так и различные. При этом он считает изображения, которые могут быть получены друг из друга поворотом или отражением, одинаковыми. Теперь Петя хочет узнать, сколько различных изображений может получиться в результате заполнения листа.

Вам предлагается написать программу, которая посчитает для Пети количество таких различных изображений.

### Входные данные

В первой и единственной строке входного файла содержатся два целых числа  $N$  и  $M$  — ширина и высота листа соответственно ( $1 \leq N, M \leq 10^9$ ).

### Выходные данные

В выходной файл необходимо вывести одно целое число — ответ на задачу.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
8 6	2
3 5	1



## Задача 2. Сейфы

Ограничение по времени:

1 секунда на тест

Грабитель проник в крупный банк, имея связку ключей.

В банке имеется  $N$  сейфов, некоторые сейфы находятся внутри других сейфов. В каждом сейфе может храниться некоторая сумма денег, а также, возможно, другие сейфы. Каждым ключом грабитель может открыть ровно один из сейфов, при этом каждый сейф открывается не более чем одним ключом.

Про связку ключей известно, что:

- 1) в ней ровно  $M$  ключей;
- 2) если в связке есть ключ от какого-то сейфа, то есть и ключи от всех сейфов, внутри которых он находится.

Вам требуется написать программу, которая определит для каждой связки ключей, удовлетворяющей этим условиям, количество денег, которые может добыть грабитель, и посчитать сумму по всем таким связкам.

### Входные данные

В первой строке входного файла заданы через пробел два целых числа  $N$  и  $M$  — количество сейфов и количество ключей в связке соответственно ( $1 \leq M \leq N \leq 150$ ).

В следующих  $N$  строках содержится описание сейфов, по строке на каждый сейф. Сейфы пронумерованы числами от 1 до  $N$ . Описание сейфов дается в порядке их нумерации.

Первое число в строке описания сейфа — это количество непосредственно вложенных в него сейфов, далее через пробел перечисляются номера вложенных сейфов. Последнее целое число в строке  $S$  — сумма денег, которые хранятся в данном сейфе ( $0 \leq S < 10^9$ ).

### Выходные данные

В выходной файл вывести целое число — остаток от деления суммы всех денег, которые может получить грабитель для всех возможных наборов ключей, на число  $(10^9 + 7)$ .

### Примеры

<i>input.txt</i>	<i>output.txt</i>
4 1 1 2 0 0 40 0 50 0 100	150
4 2 0 7 2 1 3 5 0 9 0 13	44

### Комментарий

В первом примере у грабителя только один ключ, а сейфов в банке — 4. Если у него будет ключ от сейфа с номером 1, то он получит 0 рублей, если от сейфа 3, то ему достанется 50 рублей, а если ключ от сейфа с номером 4, то 100 рублей. Так как у него только один ключ, то сейф с номером 2 он открыть не сможет, т.к. он находится внутри первого сейфа, и для его открытия грабитель должен иметь, как минимум, два ключа. Таким образом, сумма всех возможных вознаграждений грабителя составляет 150 рублей.

Во втором примере грабитель может двумя ключами открыть сейфы 2 и 1, либо 2 и 3, либо 2 и 4, добыв, соответственно, 12, 14 или 18 рублей. Итого, сумма всех возможных вознаграждений составит 44 рубля.



### Задача 3. Клад

Ограничение по времени:

1 секунда на тест

Как-то по утру Петр выудил из моря бутылку, в которой обнаружилась карта соседнего острова. К карте прилагалась зашифрованная записка. Обратившись к знакомому криптографу, Петр выяснил, что отмеченные на карте три точки — это вершины треугольника, в точке пересечения высот которого спрятан пиратский клад.

Помогите Петру по координатам этих точек вычислить место, где зарыто сокровище. Напишите программу, которая определит координаты спрятанного клада.

#### Входные данные

Во входном файле через пробел записаны три пары целых чисел, по модулю не превосходящих 1000. Это координаты трех точек на карте.

Гарантируется, что точки не лежат на одной прямой.

#### Выходные данные

В выходной файл необходимо вывести через пробел два числа —  $X$ - и  $Y$ -координаты точки пересечения высот заданного треугольника. Ответ будет считаться правильным, если его относительная или абсолютная погрешность не будет превышать  $10^{-4}$ . Гарантируется, что координаты клада по модулю не превосходят 10000.

#### Пример

<i>input.txt</i>	<i>output.txt</i>
-7 -4 -3 -3 -7 -3	-7 -3

### Задача 4. Белка

Ограничение по времени:

1 секунда на тест

В очень странном лесу живет белка. В этом лесу растет  $F$  деревьев. Деревья состоят из веток и вершин. Любая ветка соединяет две различные вершины. В каждом дереве есть одна вершина, которая называется корнем, она расположена на земле. До любой вершины в дереве можно добраться от корня по веткам, причем единственным способом. Назовем высотой вершины количество веток, лежащих на пути от данной вершины до корня. Корень имеет высоту, равную 0. В некоторых вершинах деревьев могут находиться запасы орехов.

Белка любит прыгать по деревьям, причем прыгает только по вершинам. При себе она носит мешок с орехами, которыми иногда подкрепляется. Белка может свободно прыгать с одной вершины на другую вершину, если эти вершины находятся на одинаковой высоте, независимо, на каком дереве они находятся. Чтобы перепрыгнуть на вершину, имеющую другую высоту, большую или меньшую высоты той вершины, в которой белка находится, белка должна подкрепиться: съесть из своего мешка количество орехов, равное разнице между высотами начальной и конечной вершин. Перепад высот, на который может прыгнуть белка, ограничен числом  $D$ . Если в мешке белки нет нужного количества орехов, либо перепад высот превосходит число  $D$ , такой прыжок совершить не удастся. Если белка попадает в вершину, в которой хранятся орехи, она забирает их все в свой мешок.

В начальный момент времени белка находится в вершине  $v_1$  дерева  $t_1$ , в её мешке лежит  $N$  орехов. Ей требуется добраться до вершины  $v_2$  дерева  $t_2$ . Определите, какое максимальное количество орехов белка может в эту вершину принести.



## Входные данные

В первой строке входного файла записаны три целых числа  $F$ ,  $N$  и  $D$ , где  $F$  — количество деревьев в лесу,  $N$  — количество орехов у белки в начальный момент, и  $D$  — ограничение на высоту прыжка ( $1 \leq F \leq 100$ ,  $0 \leq N \leq 10^5$ ,  $1 \leq D \leq 100$ ).

Назовем координатами вершины пару чисел, первое из которых — номер дерева, в котором эта вершина находится, а второе — ее номер в этом дереве. Деревья пронумерованы числами от 1 до  $F$ . Если в  $i$ -ом дереве  $t_i$  вершин, то вершины этого дерева пронумерованы числами от 1 до  $t_i$ . Количество вершин в каждом дереве — это положительное число, не превосходящее 1500.

Во второй строке входного файла записаны через пробел четыре числа  $t_1$ ,  $v_1$ ,  $t_2$ ,  $v_2$  — координаты начальной и конечной вершин.

Далее по порядку номеров описаны  $F$  деревьев. Дерево с номером  $i$  описывается в следующем формате. В первой строке описания заданы числа  $t_i$  и  $r_i$  — количество вершин в дереве и номер вершины, которая является корнем. В следующей строке записаны через пробел  $t_i$  неотрицательных целых чисел, задающих количество орехов в каждой из вершин этого дерева, перечисленных по порядку номеров. Количество орехов в любой вершине не превосходит  $10^5$ .

В следующих  $(t_i - 1)$  строках перечислены ветки этого дерева. Каждая ветка задается двумя положительными целыми числами — номерами вершин, которые она соединяет.

## Выходные данные

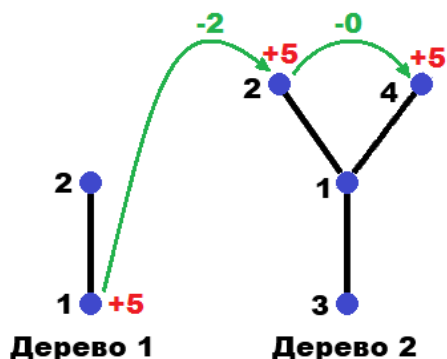
Если белка не сможет добраться до вершины  $v_2$  дерева  $t_2$ , в выходной файл нужно вывести число  $-1$ . Иначе в первую строку выходного файла нужно записать целое число, равное максимальному количеству орехов, которые белка сможет принести с собой в мешке в эту вершину, перемещаясь по вершинам деревьев в лесу. В следующих строках необходимо вывести любой маршрут, пройдя по которому белка соберёт указанное количество орехов. Маршрут белки — это последовательность координат вершин, по которым будет белка прыгать. Координаты каждой вершины следует записывать в отдельную строку через пробел. Маршрут начинается с координат начальной вершины и заканчивается координатами конечной вершины.

## Пример

<i>input.txt</i>	<i>output.txt</i>
2 0 2 1 1 2 4 2 1 5 0 1 2 4 3 0 5 0 5 1 2 1 4 3 1	13 1 1 2 2 2 4

## Комментарий

На рисунке показан один из возможных оптимальных маршрутов для приведенного в условии примера. Стрелки обозначают направление движения белки по вершинам деревьев.





## Задача 5. Инди

Ограничение по времени:

2 секунды на тест

Индиана Джонс в опасности.

Очередные археологические раскопки в Египте не предвещали ничего серьёзного до тех пор, пока исследователи не наткнулись на затерянную гробницу. И вот теперь Инди находится в самой её глубине, в зале с таинственным артефактом. Он очень хочет заполучить ценный трофей, но между ним и артефактом пролегает ров со змеями. «Змеи — замечательные животные, такие маленькие и гладкие», — думает он, отчаянно пытаясь побороть себя.

Над ровом выступает цепочка из  $N$  камней. Все камни находятся на расстоянии одного фута друг от друга. На таком же расстоянии находятся первый и последний камни от соответствующих берегов рва. Инди хочет пропрыгать по камням на противоположный берег, забрать артефакт, а затем вернуться по камням обратно. Он умеет прыгать на любое расстояние до  $L$  футов включительно, прыжок занимает одну секунду времени.

По неизвестной причине каждый камень опускается в ров сразу же после того, как на него наступили. Поэтому на каждый камень можно прыгать не более одного раза. Более того, Инди не может ждать, стоя на камне, потому что в этом случае встреча со змеями ему обеспечена. Так что, приземлившись на камень, он должен сразу же прыгнуть дальше. Профессор от природы нетерпелив, поэтому ждать на берегу рва он тоже не может.

Дополнительная сложность заключается в том, что гробница медленно, но верно рушится. Стоит ли упоминать, в чём причина такого невезения? ... Папаша всегда говорил Инди: надо сначала думать, а потом жать на кнопки. Но, увы, теперь уже ничего не поделаешь. С потолка регулярно падают осколки внушительного размера. Если в какой-то момент осколок упадёт на камень, то воспользоваться им в дальнейшем будет уже нельзя. Для каждого камня известно, в течение какого времени он останется безопасным для использования (у Индианы Джонс типичный дар предвидения).

Профессор мог бы остановиться и подумать... если бы не та злополучная кнопка! Действовать нужно быстро. Помогите ему, напишите программу, которая посчитает минимальное время, за которое он может добраться до другого берега и вернуться назад.

### Входные данные

В первой строке входного файла задано два целых числа:  $N$  — количество камней надо рвом и  $L$  — дальность прыжка Инди в футах ( $1 \leq N \leq 1000$ ,  $1 \leq L \leq \min(N, 20)$ ).

Во второй строке записано  $N$  целых чисел — информация о камнях,  $k$ -ое из них равно  $T_k$  — количество секунд, в течение которых  $k$ -й камень будет безопасным ( $1 \leq T_k \leq 1000$ ). Камни пронумерованы числами от 1 до  $N$  в порядке их удаления от того берега, на котором стоит Инди.

*Пояснение:* фактически, Инди может прыгнуть на  $k$ -ый камень, если он совершил менее  $T_k$  прыжков до этого.

### Выходные данные

Если забрать артефакт и вернуться живым невозможно, в выходном файле должно быть записано число  $-1$ .

В противном случае в первую строку выходного файла необходимо вывести целое число  $T$  — минимальное время в секундах. Далее через пробел должны быть записаны  $(T+1)$  целых чисел — оптимальный маршрут Инди — номера камней в порядке их использования. Будем считать, что берег, на котором стоит Инди, имеет номер 0, а берег с артефактом — номер  $N+1$ . Если существует несколько оптимальных маршрутов, то можно вывести любой.

## Пример

<i>input.txt</i>	<i>output.txt</i>
3 3 1 1 1	-1
5 3 3 1 4 3 1	5 0 2 4 6 3 0

## Система оценки

Некоторые тесты объединены в группы. За каждую группу тестов баллы даются только в том случае, если решение корректно работает на всех тестах из неё.

Решение, корректно работающее при дополнительном условии  $N \leq 50$ , получает 40 баллов.

Решение, корректно работающее при дополнительном условии  $N \leq 200$ , получает 60 баллов.

Решение, корректно работающее при дополнительном условии  $N \leq 333$ , получает 80 баллов.

## Комментарий

В первом примере Индиана Джонс может попасть первым прыжком на любой камень, но не на противоположный берег. После второго прыжка все камни станут недоступными, значит, он не успеет вернуться назад.

Во втором примере оптимальное время нельзя улучшить. Действительно, чтобы за два прыжка оказаться на противоположном берегу, необходимо использовать третий камень. Использовать его два раза нельзя, значит, за четыре прыжка сделать всё необходимое не получается. В оптимальном решении нельзя прыгать сразу на третий камень, потому что в таком случае не получится вернуться назад.

