

## Problem A. Политическая задача (только Div. 1)

Input file: `parties.in`  
Output file: `paeries.out`  
Time limit: 5 seconds  
Memory limit: 256 megabytes

Государственное устройство Берляндии - федерация. Она делится на провинции одинаковых размеров. Провинции формируют страну прямоугольной формы размера  $n \times m$  провинций.

В каждой провинции есть единственная партия власти. В канун президентских выборов один из кандидатов решает, какие партии выгодно поддержать. Уже определившись, что это будут две партии, он понимает, что если поддержать какую-нибудь партию, то жители провинций, в которых эта партия является партией власти, скорее всего за него проголосуют. На прямоугольной карте Берляндии провинции, голосующие за кандидата, формируют связные компоненты. Здесь под *связной компонентой* понимается такое множество провинций, что между любыми двумя провинциями существует путь по провинциям, смежным по стороне, из этого множества.

Кандидат на пост президента страны, выбирая две партии, видит своей целью максимизировать площадь (количество провинций) в наибольшей компоненте. В случае неоднозначности выбора таких партий он хочет максимизировать количество компонент связности. Если по-прежнему выбор неоднозначен, то кандидат стремится выбрать лексикографически наименьшую из упорядоченных пар  $a, b$  (то есть ту, у которой  $a$  наименьшее или  $b$  наименьшее, если  $a$  равны).

### Input

Входной файл содержит один или несколько тестов.

Каждое описание теста начинается со строки, содержащей два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 500$ ), где  $n$  задает количество провинций в столбце, а  $m$  — количество провинций в строке. Следующие  $n$  строк содержат по  $m$  целых чисел, задающих идентификаторы политических партий. Идентификаторы лежат между 1 и  $10^9$  включительно. Гарантируется, что в данной матрице есть хотя бы два различных числа.

Количество тестов во входном файле не превосходит 100, суммарное количество провинций по всем тестам не превосходит  $2.5 \cdot 10^5$ .

### Output

Для каждого теста выведите строку, содержащую пару искомых чисел  $a, b$  — идентификаторы партий ( $a < b$ ).

## Examples

parties.in	paeries.out
3 3	1 2
1 1 1	1 5
2 2 3	22 26
2 1 1	
5 5	
1 5 3 1 5	
5 3 1 1 5	
3 4 1 1 5	
3 4 3 1 5	
5 3 4 4 1	
3 3	
26 26 26	
26 6 22	
22 17 17	

## Problem B. Упрощение формулы

Input file:            formula.in  
Output file:           formula.out  
Time limit:            5 seconds  
Memory limit:         256 megabytes

Ваша задача - упростить заданную математическую формулу. Формула удовлетворяет следующей грамматике:

- $\langle formula \rangle \rightarrow (\langle formula \rangle)$
- $\langle formula \rangle \rightarrow -\langle formula \rangle$
- $\langle formula \rangle \rightarrow \cos(\langle formula \rangle)$
- $\langle formula \rangle \rightarrow \sin(\langle formula \rangle)$
- $\langle formula \rangle \rightarrow \langle formula \rangle + \langle formula \rangle$
- $\langle formula \rangle \rightarrow \langle formula \rangle * \langle formula \rangle$
- $\langle formula \rangle \rightarrow \langle formula \rangle - \langle formula \rangle$
- $\langle formula \rangle \rightarrow x|0|1|2|3|4|5|6|7|8|9$

где  $x$  - переменная на отрезке  $[a, b]$ . Ваша задача - найти формулу, которая:

- удовлетворяет описанной выше грамматике;
- определяет ту же функцию, что и заданная формула, на отрезке  $[a, b]$ ;
- имеет строковое представление минимально возможной длины;
- имеет лексикографически наименьшее строковое представление среди всех таких формул.

Гарантируется, что ответ содержит не более 8 символов.

### Input

Первая строка входного файла содержит формулу. Вторая строка содержит отрезок « $[a, b]$ », где  $a$  и  $b$  - целые числа ( $-10 \leq a \leq b \leq 10$ ). Формула состоит из не более чем 28 символов и не содержит пробелов.

### Output

Выведите искомую формулу в единственной строке выходного файла. Она не должна содержать пробелов. Гарантируется, что ответ содержит не более 8 символов.

### Examples

formula.in	formula.out
cos(x)*sin(x)*2 [0, 1]	sin(2*x)

## Problem C. Лягушка (только Div. 1)

Input file: frog.in  
Output file: frog.out  
Time limit: 10 seconds  
Memory limit: 256 megabytes

В одном из концов плоской пещеры сидит лягушка. Пол пещеры можно представить в виде отрезка  $[0, n]$  на оси  $OX$ . Изначально лягушка сидит в точке 0. Потолок пещеры ограничен ломаной, состоящей из  $k$  узлов  $px_i, py_i$  ( $px_1 = 0, px_k = n, px_i < px_{i+1}$  for  $i = 1 \dots k - 1$ ). Внутренностью потолка является множество точек, лежащих строго выше ломаной.

Под потолком парят, не меняя своего местоположения,  $m$  комаров. Все комары находятся строго ниже потолка. Также у лягушки имеется заданная длина языка  $s$ . После того, как она съест одного комара, ее язык вырастает на 2 в длину. Лягушка способна съесть комара только в том случае, если длина ее языка не меньше расстояния от лягушки до комара и отрезок, соединяющий точки их местоположений, не имеет общих точек с внутренностью потолка.

Лягушка может прыгать между целочисленными точками на отрезке, то есть она способна прыгнуть, находясь в точке  $i$ , в точку  $j$ , где  $0 \leq i, j \leq n$  и  $i \neq j$ . В этом случае траектория полета лягушки это полудуга окружности, построенной на отрезке, соединяющем точки  $i$  и  $j$ , как на диаметре. Траектория полета не может иметь общих точек с внутренностью потолка. Лягушка не способна есть комаров во время полета.

Так как лягушка очень жадная, то она хочет максимизировать количество съеденных ею комаров. И конечно, это не единственный ее порок, что мешает жить не только ей: она очень ленивая лягушка, поэтому хочет совершить как можно меньше прыжков, съев максимальное количество комаров и оказавшись в другом конце пещеры.

### Input

Первая строка входного файла содержит 4 целых числа  $n, k, m, s$  ( $1 \leq n \leq 200, 2 \leq k \leq 10\,000, 1 \leq m \leq 10, 1 \leq s \leq 200$ ), задающих длину отрезка, по которому лягушка перемещается, количество узлов ломанной, количество комаров и начальную длину лягушачьего языка соответственно.

Каждая из следующих  $k$  строк содержит по два вещественных числа  $px_i, py_i$ , задающих координаты узлов ломанной ( $1 \leq py_i \leq 200$ ). Координаты даны с точностью не более двух знаков после десятичной точки.

Следующие  $m$  строк содержат по два вещественных числа  $qx_i, qy_i$  ( $1 \leq qy_i \leq 200, 0 \leq qx_i \leq n$ ), которые задают координаты комаров. Координаты даны с точностью не более двух знаков после десятичной точки. Никакие два комара не находятся в одной точке пространства.

### Output

Выведите в единственную строку выходного файла два числа, разделенных пробелом: максимальное количество комаров и наименьшее количество прыжков, требуемых лягушке, чтобы удовлетворить своей ненасытный аппетит и оказаться в точке  $n$ .

## Examples

frog.in	frog.out
4 5 1 1 0.00 2.00 1.00 1.00 2.00 2.00 3.00 1.00 4.00 2.00 2.00 1.00	1 2
5 6 2 1 0.00 3.00 1.00 4.00 2.00 3.00 3.00 1.00 4.00 2.00 5.00 1.00 1.00 3.00 4.00 1.00	2 6

## Problem D. Лучшие подарки (только Div. 1)

Input file:            `gifts.in`  
Output file:          `gifts.out`  
Time limit:            2 seconds  
Memory limit:        256 megabytes

Елена любит, когда ей пишут письма, а также дарить своим друзьям подарки. И она считает, что конверт — это лучший подарок. Очевидно, такой подарок резко увеличивает вероятность получения писем от друзей!

У Елены  $n$  друзей, каждый из которых характеризуется числом  $a_i$  — количеством отправленных ей писем. Справедливо думать, что друзья с большей величиной этой характеристики заслуживают лучших подарков!

Елена уже решила, что  $i$ -му другу она подарит хотя бы  $a_i$  конвертов. Войдя в магазин, она обнаружила, что на прилавках имеются только упаковки по  $m$  конвертов в каждой. Купив упаковку, Елена хочет раздарить все  $m$  конвертов (она не хочет оставлять конверты у себя, так как сочинение писем навевает ей скуку).

Впереди череда праздников, и Елена планирует подготовить несколько подарочных наборов. Каждый набор будет состоять из  $m$  конвертов (из одной упаковки). В каждом наборе должно быть  $n$  подарков, по одному на каждого друга, с одним важным ограничением:  $i$ -ый друг должен получить  $a_i$  или более конвертов в подарке.

Согласно воспоминаниям Елены, далеко не всем друзьям нравился ее подход к разнообразию подарков. Они думали, что ее подарки похожи друг на друга. И теперь она решила не повторять свои прежние ошибки. Поэтому она впредь не будет делать два таких подарочных набора, в которых хотя бы для одного друга количество конвертов в его подарке в этих наборах одинаково.

Елена не знает, сколько упаковок конвертов ей приобрести. Помогите ей, а для этого определите, какое максимальное число подарочных наборов она может приготовить.

### Input

Первая строка входного файла содержит количество друзей Елены  $n$  и количество конвертов в упаковке  $m$  ( $1 \leq n \leq 1000$ ;  $1 \leq m \leq 10\,000$ ). Во второй строке содержится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 1000$ ), где  $a_i$  — это минимальное число конвертов, которые должен получить в подарке  $i$ -ый друг Елены.

### Output

В первую строку выведите максимальное число подарочных наборов  $t$ . Следующие  $t$  строк должны содержать описание подарочных наборов. Выведите в каждую строку по  $n$  целых чисел, где  $i$ -ое число равно количеству конвертов в подарке для  $i$ -го друга в соответствующем подарочном наборе.

### Examples

<code>gifts.in</code>	<code>gifts.out</code>
2 3	2
1 1	1 2
	2 1

## Problem E. Лёнины четверки

Input file:           fours.in  
Output file:          fours.out  
Time limit:           5 seconds  
Memory limit:        256 megabytes

Однажды давным-давно жил-был Лёня. Он был известен среди математиков всего мира из-за своей любимой игры. Правила игры были предельно просты: Лёня давал игроку  $n$  целых чисел  $a_1, a_2, \dots, a_n$  и просил его выписать все четверки различных индексов  $i, j, k, p$  таких, что  $a_i + a_j = a_k + a_p$ . Многие сошли с ума в поисках этих четверок. Ассоциация Снарковских Математиков (АСМ) решила, что Лёня всех дурит. В процессе начавшегося расследования вам поручили сосчитать требуемое количество четверок.

### Input

В первой строке входных данных содержится целое число  $n$  ( $4 \leq n \leq 3000$ ). Во второй строке входного файла содержится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ).

### Output

Выведите одно число - количество способов выбрать четыре различных индекса  $i, j, k, p$  таких, что  $a_i + a_j = a_k + a_p$ .

### Examples

fours.in	fours.out
4 1 2 3 4	8
4 1 1 1 1	24

### Note

Четверки  $(i_1, j_1, k_1, p_1)$  и  $(i_2, j_2, k_2, p_2)$  считаются одинаковыми тогда и только тогда, когда  $i_1 = i_2$ ,  $j_1 = j_2$ ,  $k_1 = k_2$  и  $p_1 = p_2$ .

## Problem F. Задача о доставке (только Div. 1)

Input file:           ndice.in  
Output file:         ndice.out  
Time limit:          2 seconds  
Memory limit:       256 megabytes

Мистер Битл работает почтальоном. Его рутинная работа заключается в доставке новых выпусков местного журнала “N-dice” его подписчикам. Каждый понедельник ровно в 8 утра он садится на свой велосипед и отправляется работать. Он должен доставить один экземпляр журнала каждому подписчику.

Экземпляры нового журнала могут иметь обложки разного цвета, к примеру желтого, зеленого, синего и т.п. Предположим, что цвета пронумерованы от 1 до  $N$ , и для любого цвета  $i$  в сумке мистера Битла содержится бесконечное число экземпляров с таким цветом обложки. Он знает, что подписчикам не важно, журнал с каким цветом обложки им достанется. Поэтому, подъезжая к дому очередного подписчика, он действует следующим образом: засовывает руку в сумку, наугад достает экземпляр журнала и опускает его в почтовый ящик.

Мистер Битл любит такие дни, когда ему удастся доставить хотя бы по одному экземпляру каждого цвета. Если вероятность этого не менее  $x$ , он называет такой день “счастливым”.

Мистер Битл не любит утруждать себя вычислениями. Но он хочет знать, какое минимальное число экземпляров он должен разнести подписчикам, чтобы быть уверенным, что день будет счастливым. Ваша задача — помочь мистру Битлу найти это число.

### Input

Входной файл содержит целое число  $N$  ( $1 \leq N \leq 7000$ ) и вещественное число  $x$ , заданное ровно с 6 знаками после десятичной точки ( $0.1 \leq x < 1$ ).

### Output

Выведите минимальное число экземпляров журнала, которые мистер Битл должен разнести подписчикам, чтобы быть уверенным, что день будет счастливым.

### Examples

ndice.in	ndice.out
1 0.999999	1
2 0.777770	4



## Problem G. Задача психолога

Input file: `psychologist.in`  
Output file: `psychologist.out`  
Time limit: 5 seconds  
Memory limit: 256 megabytes

Профессор Неврозис — практикующий психолог. Он очень популярен в Берляндии. По прибытии в клинику, каждый клиент говорит секретарю длительность  $t_i$  предстоящего сеанса с психологом.  $i$ -й клиент приходит в клинику в момент времени  $s_i$ , и секретарь незамедлительно сообщает профессору о новом пациенте и необходимом времени сеанса  $t_i$ .

Профессор знает, что многие его пациенты - деловые люди, ценящие свое время. Поэтому он хочет минимизировать суммарное время, проведенное пациентами в стенах его клиники. Для каждого пациента время в клинике состоит из времени ожидания и длительности сеанса.

Недавно Неврозис пришел к выводу, что перерывы в сеансах не так уж страшны, как кажутся на первый взгляд, потому что может подойти пациент с новой психологической проблемой. Поэтому он делает перерывы в сеансах, переключаясь между пациентами.

Помогите профессору Неврозису распределить работу с пациентами таким образом, чтобы суммарное время, проведенное пациентами в клинике, было минимальным.

### Input

В первой строке входного файла записано целое число  $n$  ( $1 \leq n \leq 10^5$ ) - количество пациентов в течение дня. Следующие  $n$  строк содержат описания пациентов. Каждый пациент описывается двумя целыми числами  $s_i, t_i$  ( $1 \leq s_i, t_i \leq 10^8$ ), где  $s_i$  - это время прихода  $i$ -ого пациента в клинику, и  $t_i$  - продолжительность сеанса с  $i$ -ым пациентом. Строки расположены таким образом, что  $s_i \leq s_{i+1}$  при всех  $1 \leq i < n$ .

### Output

Выведите минимальное суммарное время в первой строке выходного файла. Следующие строки должны содержать расписание работы профессора. Каждая строка должна содержать три целых числа  $j, from, to$  ( $1 \leq j \leq n, from < to$ ), и это означает, что во временном промежутке  $[from, to)$  профессору следует работать с  $j$ -ым пациентом. Очевидно, что нельзя работать с пациентом до его прихода. Сумма длин интервалов работы для каждого пациента  $j$  должна быть  $t_j$ . Упорядочите строки в хронологическом порядке. Если решений несколько, выведите любое из них. Выходной файл не должен содержать более  $10^6$  строк с расписанием профессора.

### Examples

<code>psychologist.in</code>	<code>psychologist.out</code>
2	14
4 4	1 4 8
10 10	2 10 20

## Problem H. Круглый стол

Input file:            table.in  
Output file:           table.out  
Time limit:            10 seconds  
Memory limit:         256 megabytes

Однажды Главы Берляндских провинций принимали участие в саммите в городе Бремлин. Советание было проведено на самом высоком уровне, и проходило два дня. Каждый день  $n$  Глав провинций садились за стол и обсуждали очень важные вопросы. Естественно, стол был круглый. Порядок, в котором сидели Главы вокруг стола, не менялся в течении дня, но мог меняться между днями.

Известно, что Главы провинций очень консервативны и не могут вести конструктивный диалог в меняющихся условиях. Две Главы могли вести конструктивный диалог тогда и только тогда, когда число людей, сидящих между ними, было одинаковым в оба дня. Конечно же, существует два способа посчитать количество людей, разделяющих двух Глав, но в Берляндии по традиции считают от первого Главы ко второму по часовой стрелке (это число может быть равно 0).

На церемонии закрытия саммита правительство Берляндии решило наградить одну из пар Глав, которая вела конструктивную дискуссию. Однако правительство обнаружило, что совсем не просто определить такие пары. Помогите им решить эту задачу и найти нужную пару Глав.

### Input

Первая строка входных данных содержит целое число  $n$  ( $3 \leq n \leq 10^6$ ), число Глав Берляндских провинций. Каждой главе назначен целый номер (от 1 до  $n$ ). Вторая строка входных данных содержит последовательность чисел, которые описывают расположение Глав вокруг стола в течение первого дня. Третья (последняя) строка входных данных содержит последовательность, описывающую порядок, в котором Главы сидели вокруг стола во второй день. Каждая из последних двух строк входных данных есть перестановка чисел от 1 до  $n$ , включительно.

### Output

Выведите пару целых чисел — числа, назначенные любой паре Глав, которые вели конструктивную дискуссию. Если существует несколько пар таких чисел, выведите любую из них. Выведите “-1 -1” (без кавычек) в том случае, если нет такой пары Глав.

### Examples

table.in	table.out
5 3 4 2 1 5 1 2 3 4 5	3 1

## Problem I. К-ая пирамида (только Div. 1)

Input file:            `heaps.in`  
Output file:           `heaps.out`  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Для заданного  $n$  рассмотрим все перестановки чисел от 1 до  $n$ . Каждая перестановка это массив различных чисел от 1 до  $n$ , проиндексированный числами от 1 до  $n$ .

Перестановка называется пирамидой, если для всех его элементов с индексами  $i$  ( $i > 1$ ) верно неравенство  $p[i] < p[i/2]$ , где  $i/2$  это целая часть от половины значения  $i$ .

Например, следующие перестановки являются пирамидами для  $n = 4$ :  $[4, 2, 3, 1]$ ,  $[4, 3, 1, 2]$  и  $[4, 3, 2, 1]$ .

Отсортируем все пирамиды для заданного числа  $n$  в лексикографическом порядке. То есть перестановки следует сравнивать в первую очередь по первому числу, при их равенстве по второму и т.д.

Ваша задача по данным значениям  $n$  и  $k$  найти  $k$ -ую пирамиду в отсортированном списке  $n$ -элементных пирамид.

### Input

Входные данные состоят из пары целых чисел  $n, k$  ( $1 \leq n \leq 1000; 1 \leq k \leq 10^{15}$ ).

### Output

Выведите искомую пирамиду, разделяя ее элементы пробелами. Выведите число -1 в случае, если существует менее  $k$  пирамид размера  $n$ .

### Examples

<code>heaps.in</code>	<code>heaps.out</code>
4 2	4 3 1 2

## Problem J. Биндианские Вигвамы

Input file: wigwams.in  
Output file: wigwams.out  
Time limit: 10 seconds  
Memory limit: 256 megabytes

Много лет тому назад великие Биндианские племена населяли территорию современной Берляндии. Биндианцы по старым традициям выстраивали свои дома, "Вигвамы в форме тетраэдров. Интересно, как же можно построить "Вигвам"? О, это действительно очень просто! Достаточно найти 6 деревянных палок и сконструировать тетраэдр, используя палки в качестве его ребер. Конечно же, палки очень жесткие и вам запрещено ломать их или использовать часть палки как ребро тетраэдра. Для удобства жителей, "Вигвам" обязательно должен иметь положительный объем.

Раз в несколько десятилетий Древние Биндианцы любили организовывать соревнования, где наиболее популярная задача была сформулирована следующим образом: дана куча деревянных палок, найти количество способов выбрать 6 палок из кучи таким образом, чтобы из них можно было построить "Вигвам". Древние Биндианцы с ловкостью умели решать эту задачу. А как же вы? Не забудьте, что 2 палки считаются разными, даже если они имеют одинаковую длину.

### Input

Первая строка входных данных содержит целое число  $n$  ( $1 \leq n \leq 30$ ), где  $n$  это количество палок в куче. Следующая строка содержит список длин палок в куче, разделенных пробелами. Длина палки это целое число от 1 до 100.

### Output

Выведите только одно целое число — число способов выбрать 6 палок из кучи так, чтобы из них можно было построить "Вигвам".

### Examples

wigwams.in	wigwams.out
7 4 3 4 4 4 4 4	7
6 1 1 1 1 1 1	1

## Problem K. Суслики (только Div. 1)

Input file:           gophers.in  
Output file:         gophers.out  
Time limit:          6 seconds  
Memory limit:       256 megabytes

Давным-давно к югу от Берляндии было большое поселение сусликов. В прошлом году бедных сусликов окончательно замучила невыносимая жара.

Тем не менее, новый щедрый правитель Берляндии решил создать поселение сусликов в Берляндском Национальном Парке (BNP). Чтобы предотвратить вымирание сусликов, вы должны сконструировать ровно два водопоя на территории BNP (у правительства не хватит средств, чтобы построить третий).

Введем декартову систему координат на BNP. Во-первых, рассмотрим BNP как квадрат с двумя противоположными углами  $(-20\,000, -20\,000)$  и  $(20\,000, 20\,000)$ . У каждого суслика есть своя норка на территории BNP внутри квадрата с противоположными углами  $(-10\,000, -10\,000)$  и  $(10\,000, 10\,000)$ . Каждый водопой должен иметь форму прямолинейного отрезка, расположенного на территории BNP (он может проходить по его границе). Водопой может проходить через одну или более норок.

В BNP много хищников, поэтому расстояние, проходимое каждым сусликом до ближайшего водопоя и обратно, должно быть минимально возможным. Помните, что суслики не могут выходить за пределы BNP!

### Input

Первая строка содержит целое число  $N$  ( $1 \leq N \leq 120$ ) — количество сусликов в популяции. Следуют  $N$  строк; каждая из них описывает расположение норки  $i$ -ого суслика двумя координатами  $x$  и  $y$ .

### Output

Первая строка выходного файла должна содержать одно вещественное число — минимально возможное максимальное расстояние, проходимое некоторым сусликом от своей норки до ближайшего водопоя и обратно. Выведите это число как минимум с девятью знаками после десятичной точки.

### Examples

gophers.in	gophers.out
5 -10 0 0 0 10 0 0 10 0 -10	0.000000000000
7 -10 0 0 0 10 0 0 10 1 10 0 -10 1 -10	1.000000000000

## Problem L. Политическая задача (только Div. 2)

Input file:           parties.in  
Output file:          parties.out  
Time limit:           5 seconds  
Memory limit:        256 megabytes

Государственное устройство Берляндии - федерация. Она делится на провинции одинаковых размеров. Провинции формируют страну прямоугольной формы размера  $n \times m$  провинций.

В каждой провинции есть единственная партия власти. В канун президентских выборов один из кандидатов решает, какие партии выгодно поддержать. Уже определившись, что это будут две партии, он понимает, что если поддержать какую-нибудь партию, то жители провинций, в которых эта партия является партией власти, скорее всего за него проголосуют. На прямоугольной карте Берляндии провинции, голосующие за кандидата, формируют связные компоненты. Здесь под *связной компонентой* понимается такое множество провинций, что между любыми двумя провинциями существует путь по провинциям, смежным по стороне, из этого множества.

Кандидат на пост президента страны, выбирая две партии, видит своей целью максимизировать площадь (количество провинций) в наибольшей компоненте. В случае неоднозначности выбора таких партий он хочет максимизировать количество компонент связности. Если по-прежнему выбор неоднозначен, то кандидат стремится выбрать лексикографически наименьшую из упорядоченных пар  $a, b$  (то есть ту, у которой  $a$  наименьшее или  $b$  наименьшее, если  $a$  равны).

### Input

Входной файл содержит один или несколько тестов.

Каждое описание теста начинается со строки, содержащей два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ), где  $n$  задает количество провинций в столбце, а  $m$  — количество провинций в строке. Следующие  $n$  строк содержат по  $m$  целых чисел, задающих идентификаторы политических партий. Идентификаторы лежат между 1 и  $10^9$  включительно. Гарантируется, что в данной матрице есть хотя бы два различных числа.

Количество тестов во входном файле не превосходит 100, суммарное количество провинций по всем тестам не превосходит  $10^4$ .

### Output

Для каждого теста выведите строку, содержащую пару искомых чисел  $a, b$  — идентификаторы партий ( $a < b$ ).

## Examples

parties.in	parties.out
3 3	1 2
1 1 1	1 5
2 2 3	22 26
2 1 1	
5 5	
1 5 3 1 5	
5 3 1 1 5	
3 4 1 1 5	
3 4 3 1 5	
5 3 4 4 1	
3 3	
26 26 26	
26 6 22	
22 17 17	

## Problem M. К-ая пирамида (только Div. 2)

Input file:            heaps.in  
Output file:           heaps.out  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Для заданного  $n$  рассмотрим все перестановки чисел от 1 до  $n$ . Каждая перестановка это массив различных чисел от 1 до  $n$ , проиндексированный числами от 1 до  $n$ .

Перестановка называется пирамидой, если для всех его элементов с индексами  $i$  ( $i > 1$ ) верно неравенство  $p[i] < p[i/2]$ , где  $i/2$  это целая часть от половины значения  $i$ .

Например, следующие перестановки являются пирамидами для  $n = 4$ : [4, 2, 3, 1], [4, 3, 1, 2] и [4, 3, 2, 1].

Отсортируем все пирамиды для заданного числа  $n$  в лексикографическом порядке. То есть перестановки следует сравнивать в первую очередь по первому числу, при их равенстве по второму и т.д.

Ваша задача по данным значениям  $n$  и  $k$  найти  $k$ -ую пирамиду в отсортированном списке  $n$ -элементных пирамид.

### Input

Входные данные состоят из пары целых чисел  $n, k$  ( $1 \leq n \leq 15; 1 \leq k \leq 10^7$ ).

### Output

Выведите искомую пирамиду, разделяя ее элементы пробелами. Выведите число -1 в случае, если существует менее  $k$  пирамид размера  $n$ .

### Examples

heaps.in	heaps.out
4 2	4 3 1 2



## Problem N. Суслики (только Div. 2)

Input file:           gophers.in  
Output file:          gophers.out  
Time limit:          2 seconds  
Memory limit:        256 megabytes

Давным-давно к югу от Берляндии было большое поселение сусликов. В прошлом году бедных сусликов окончательно замучила невыносимая жара.

Тем не менее, новый щедрый правитель Берляндии решил создать поселение сусликов в Берляндском Национальном Парке (BNP). Чтобы предотвратить вымирание сусликов, вы должны сконструировать ровно два водопоя на территории BNP (у правительства не хватит средств, чтобы построить третий).

Введем декартову систему координат на BNP. Во-первых, рассмотрим BNP как квадрат с двумя противоположными углами  $(-20\,000, -20\,000)$  и  $(20\,000, 20\,000)$ . У каждого суслика есть своя норка на территории BNP внутри квадрата с противоположными углами  $(-10\,000, -10\,000)$  и  $(10\,000, 10\,000)$ . Каждый водопой должен иметь форму прямолинейного отрезка, расположенного на территории BNP (он может проходить по его границе). Водопой может проходить через одну или более норок.

В BNP много хищников, поэтому расстояние, проходимое каждым сусликом до ближайшего водопоя и обратно, должно быть минимально возможным. Помните, что суслики не могут выходить за пределы BNP!

### Input

Первая строка содержит целое число  $N$  ( $1 \leq N \leq 50$ ) — количество сусликов в популяции. Следуют  $N$  строк; каждая из них описывает расположение норки  $i$ -ого суслика двумя координатами  $x$  и  $y$ .

### Output

Первая строка выходного файла должна содержать одно вещественное число — минимально возможное максимальное расстояние, проходимое некоторым сусликом от своей норки до ближайшего водопоя и обратно. Выведите это число как минимум с девятью знаками после десятичной точки.

### Examples

gophers.in	gophers.out
5 -10 0 0 0 10 0 0 10 0 -10	0.000000000000
7 -10 0 0 0 10 0 0 10 1 10 0 -10 1 -10	1.000000000000

## Problem O. Счастливое число 7 (только Div. 2)

Input file:            seven.in  
Output file:           seven.out  
Time limit:            2 seconds  
Memory limit:         256 megabytes

У Пети есть любимое число — 7. Он пытается везде углядеть признаки наличия этого числа и, если ему это удастся, от чувствует себя по-настоящему счастливым. Вот и теперь, глядя на матрицу из цифр, он думает, как бы найти в ней побольше связи с числом 7. Может, посчитать число семерок в матрице? Ну нет, их будет слишком мало... Тогда Пете в голову пришла блестящая идея: он будет рассматривать пути фиксированной длины по этой матрице, начинающиеся в клетке (1, 1), и будет смотреть на числа, которые будут образованы последовательной записью всех цифр на пути. Если это число делится на 7, то Петя будет считать его счастливым! Таких-то путей уж точно будет много! Осталось только посчитать их количество, и Петя просит вас позаботиться об этом. Соседние клетки в пути должны быть соседними по стороне. Путь может содержать одну и ту же клетку несколько раз.

### Input

В первой строке содержатся 3 целых числа  $n$ ,  $m$  и  $k$  — размеры матрицы и длина пути по ней соответственно ( $1 \leq n, m \leq 50, 1 \leq k \leq 100$ ). В следующих  $n$  строках записано по  $m$  цифр от 1 до 9.

### Output

Выведите число счастливых путей длины  $k$ , начинающихся в клетке (1, 1). Так как это число может быть очень большим, выдайте остаток от деления его на  $10^9$ .

### Examples

seven.in	seven.out
2 2 3 7 1 7 4	2

## Problem P. Номера машин (только Div. 2)

Input file:            numbers.in  
Output file:           numbers.out  
Time limit:            2 seconds  
Memory limit:         256 megabytes

После недавней реформы автомобильные номера в Берляндии состоят из двух записанных подряд блоков символов: сначала идут три латинские буквы ( $A \dots Z$ ), затем три цифры. Первоначально Берляндская Госавтоинспекция планировала выдавать владельцам автомобилей номера в соответствии со следующим алгоритмом: начиная с номера  $AAA000$ , каждому следующему автомобилисту выдается следующий в лексикографическом порядке номер. К примеру, второй пришедший автовладелец получил бы номер  $AAA001$ , а 1001-й —  $AAB000$ . Однако в автоинспекции вовремя вспомнили, что некоторые номера кажутся водителям гораздо более красивыми, чем остальные. Многие, к примеру, хотели бы иметь номер  $XXX007$ . В автоинспекции составили список из  $M$  таких номеров, расположенных в порядке убывания привлекательности. Теперь каждому автоладельцу предоставляется выбор: за отдельную плату он может получить самый красивый из оставшихся в списке номеров, а бесплатно он просто получит лексикографически следующий номер, не входящий в список красивых. Вам в руки попал список красивых номеров и записи о том, кто из первых  $N$  авто владельцев, получивших номера, доплачивал за красивый номер. Вы хотите установить, какой автомобильный номер получил каждый из авто владельцев.

### Input

В первой строке входного файла записано два целых числа  $N$  и  $M$  — число авто владельцев, получивших номера, и число красивых номеров в списке ( $1 \leq N \leq 10000, 0 \leq M \leq 1000$ ). В следующих  $M$  строках записаны номера, входящие в список красивых, по одному в строке, в порядке убывания красоты. В последней строке входного файла записана строка  $s$  из  $N$  символов 0 или 1, причем  $s_i = 1$ , когда  $i$ -й автовладелец доплачивал за красивый номер, и  $s_i = 0$  в противном случае. Гарантируется, что в строке не более  $M$  единиц.

### Output

Выведите  $N$  строк. В  $i$ -й строке выведите номер, полученный  $i$ -м авто владельцем.

### Examples

numbers.in	numbers.out
4 2	AAA001
AAA001	AAA000
XXX007	XXX007
1010	AAA002

## Problem Q. Фонари (только Div. 2)

Input file:            lights.in  
Output file:           lights.out  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Для управления состоянием фонарей на главной улице столицы Берляндии власти хотят приобрести дорогостоящую автоматизированную систему управления. Однако эта система управляется весьма странным образом, поэтому, чтобы устранить все сомнения, вам поручено промоделировать поведение этой системы. Фонари на главной улице расположены в линию и пронумерованы от 1 до  $N$ . Каждый фонарь может находиться в двух состояниях: включенном и выключенном. Программа управления позволяет задать отрезок  $[l_i, r_i]$  из фонарей и изменить состояние всех фонарей на отрезке на противоположное. Кроме того, ваша моделирующая программа должна в любой момент времени отвечать, включен или выключен конкретный фонарь с номером  $q_i$ . Изначально все фонари выключены.

### Input

В первой строке записано два целых числа  $N$  и  $Q$  — число фонарей и число запросов к вашей программе соответственно ( $1 \leq N \leq 100000, 1 \leq Q \leq 200000$ ). В следующих  $Q$  строках перечислены запросы. Первое число в каждой строке задает тип запроса: 1 для переключения состояний на отрезке, 2 для запроса состояния фонаря. В первом случае далее в строке записаны границы отрезка  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq N$ ), во втором случае — номер интересующего фонаря  $q_i$  ( $1 \leq q_i \leq N$ ).

### Output

Для каждого запроса второго типа выведите в отдельной строке 0, если фонарь на момент запроса выключен, и 1, если включен.

### Examples

lights.in	lights.out
4 5	1
1 2 4	1
2 2	0
1 1 3	
2 1	
2 3	