

Problem A. Area of Effect

Input file: *standard input*
Output file: *standard output*
Time limit: 10 seconds
Memory limit: 512 mebibytes

Liam (angryneeson52) играет в очередную версию игры Tower Defence. В этой игре требуется защищать свои поселения, уничтожая атакующие юниты оппонента. В данной версии юниты представлены точками на плоскости.

Наиболее часто используемая Liam-ом атака — заклинание The Area of Effect. Для применения этой атаки нужно выбрать центр и радиус, после чего любой юнит, попавший в круг или на его границу, уничтожается. При этом круг не должен задеть поселение (но может касаться его стен). Поселения также представляют собой окружности. Радиус атаки заклинания ограничен сверху.

Определите, какое количество вражеских юнитов Liam может уничтожить одной атакой The Area of Effect так, чтобы атака не нанесла ущерб своим поселениям.

Input

Первая строка входа содержит три целых числа n , m , r , где:

- n ($1 \leq n \leq 10$) — количество поселений;
- m ($1 \leq m \leq 2000$) — количество юнитов оппонента;
- r ($1 \leq r \leq 2 \cdot 10^4$) — максимальный радиус заклинания.

Каждая из последующих n строк содержит три целых числа vx , vy , vr — координаты и радиус поселения, соответственно ($-2 \cdot 10^4 \leq vx, vy \leq 2 \cdot 10^4$, $1 \leq vr \leq 2 \cdot 10^4$). Никакие два поселения не имеют общей точки (внутренней или внешней).

Каждая из последующих m строк содержит два целых числа mx , my — координаты юнитов оппонента. ($-2 \cdot 10^5 \leq mx, my \leq 2 \cdot 10^4$). Никакие два юнита не находятся в одной точке, никакой юнит не находится внутри поселения.

Output

Выведите одно целое число — максимальное количество юнитов, которые Liam может уничтожить одной атакой.

Examples

standard input	standard output
1 3 3 0 0 1 3 3 -3 3 3 -3	1
1 5 3 0 0 1 3 3 -3 3 3 -3 3 0 0 3	3
4 10 100 0 0 3 10 0 3 10 10 3 0 10 3 0 4 0 5 0 6 5 3 5 -3 5 5 6 7 3 6 10 4 8 4	5

Problem B. Canyon Mapping

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Каньоны существуют не только на Земле. Например, система каньонов Valles Marineris вблизи марсианского экватора имеет площадь, примерно равную площади США.

Крупная компания, занимающаяся изготовлением карт, поручила Вам делать карты каньонов. Каньон представляется в 2D в виде контура простого многоугольника. Карты, которые Вы делаете, должны быть квадратными и ориентированы по осям так, что верхняя часть карты расположена строго на севере. Иногда для большей детализации один каньон отображается на нескольких картах. Подобный набор карт называется системой карт. Система карт каньона должна покрывать всю территорию каньона, все карты в системе должны иметь один и тот же масштаб относительно каньона и один и тот же размер. Благодаря этому карты можно использовать совместно.

Вам требуется создать систему из k карт. Система должна полностью покрывать каньон, при этом каждая карта должна покрывать как можно меньшую область каньона. Все карты должны быть квадратными и должны покрывать куски каньона одинаковой площади; карты могут перекрываться.

Выведите одно число — минимальную длину стороны квадрата, соответствующего картам из Вашей системы на местности.

Input

Первая строка входа содержит два целых числа n ($3 \leq n \leq 2000$) и k ($1 \leq k \leq 3$), где n — количество вершин многоугольника, а k — количество карт в системе.

Каждая из последующих n строк содержит по два целых числа x и y ($-2 \cdot 10^4 \leq x, y \leq 2 \cdot 10^4$) — координаты вершин многоугольника, задающего каньон, в порядке обхода. Гарантируется, что никакие две стороны многоугольника не пересекаются и не касаются, никакие две вершины не совпадают, никакие три точки не лежат на одной прямой и площадь многоугольника положительна.

Output

Выведите одно число с абсолютной точностью не хуже 10^{-2} — наименьшую длину стороны квадрата, соответствующего любой карте из построенной Вами системы.

Examples

standard input	standard output
4 1 1 1 5 1 5 5 4 2	4.0
6 3 -8 -8 0 -1 8 -8 1 0 0 10 -1 0	9
16 2 0 0 3 0 3 3 6 3 8 0 10 4 10 10 8 10 8 6 6 10 6 11 5 9 4 7 3 11 2 1 0 4	9.00

Problem C. Magic Checkerboard

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 512 mebibytes

Рассмотрим шахматную доску $n \times m$. Поместим в каждую её клетку целое положительное число. Значения в каждом столбце доски должны быть в строго возрастающем порядке при движении сверху вниз, а значения в каждой строке должны строго возрастать при движении слева направо.

```
1 2 3 4
3 4 5 6
5 6 7 8
7 8 9 10
```

Магическая доска имеет дополнительное ограничение. Клетки, которые граничат только по углу, должны иметь значения различной чётности. Например, приведённая ниже доска не является магической, так как 2 и 4 граничат только по углу и имеют одинаковую чётность.

```
1 4
2 6
```

Первая приведённая доска является магической. По частично заполненной магической доске заполните оставшиеся поля таким образом, чтобы сумма всех чисел доски была минимальна.

Input

В первой строке входа заданы два целых числа n и m , разделённые пробелом ($1 \leq n, m \leq 2000$) — количество строк и столбцов доски, соответственно. Каждая из последующих n строк содержит по m целых чисел c , разделённых пробелами ($0 \leq c \leq 2000$) — записанные числа. Если число равно 0, то соответствующая клетка пустует. Можно использовать любое целое положительное число, чтобы заполнять эти клетки, до тех пор, пока доска остаётся магической (в том числе и превосходящее 2000); при этом числа могут повторяться.

Output

Выведите одно целое число — минимальную сумму всех чисел на доске, которую можно получить после замены всех 0 на числа таким образом, чтобы доска была магической; если магическую доску получить заменой 0 невозможно, выведите -1 .

Examples

standard input	standard output
4 4 1 2 3 0 0 0 5 6 0 0 7 8 7 0 0 10	88
4 4 1 2 3 0 0 0 5 6 0 4 7 8 7 0 0 10	-1
2 3 0 0 0 0 0 0	18

Problem D. Extensive OR

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Рассмотрим очень большое число R , заданное в сжатом формате. Оно задано как двоичная строка s и целое число k . Для получения двоичной записи R необходимо провести следующие действия: взять пустую строку, после чего k раз дописать в конец имеющейся строки s . Гарантируется, что первым символом строки s является единица.

Требуется решить следующую задачу: сколько существует множеств, состоящих из n попарно различных целых чисел, таких, что каждое целое число находится между 0 и $R - 1$ включительно, а XOR всех этих чисел равен нулю? Так как ответ может быть очень велик, выведите его по модулю $10^9 + 7$.

Input

Входной файл состоит из двух строк. Первая строка содержит два целых числа n ($3 \leq n \leq 7$) и k ($1 \leq k \leq 10^5$), разделённых пробелами — количество чисел в множествах и коэффициент повторения строки s , который требуется для построения R . Вторая строка содержит s — непустую двоичную строку, состоящую из не более, чем 50 нулей или единиц, при этом начинающуюся с единицы.

Output

Выведите одно целое число — остаток от деления требуемого количества множеств на $10^9 + 7$.

Examples

standard input	standard output
3 1 100	1
4 3 10	1978
5 100 1	598192244

Problem E. Primal Partitions

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Задана последовательность из n целых положительных чисел. Требуется разделить последовательность на k последовательных непрерывных участков так, чтобы в каждом участке было не менее одного целого числа. Каждое разбиение оценивается следующим образом: для каждого участка находится наибольший общий **простой** делитель; если такого делителя нет (1 простым числом не является), то оценка за участок равна 0, иначе она равна этому делителю. Оценка разбиения равно минимуму оценок по всем участкам.

Требуется найти разбиение, дающее максимальную оценку.

Input

В первой строке входа заданы два целых числа n ($1 \leq n \leq 2 \cdot 10^4$) и k ($1 \leq k \leq \min(100, n)$), где n — количество целых положительных чисел в исходной последовательности и k — количество участков в разбиении. В следующей строке заданы n целых чисел v ($1 \leq v \leq 10^6$) — исходная последовательность.

Output

Выведите одно целое число — максимальную оценку, которую Вы можете получить за разбиение, удовлетворяющее условию задачи.

Examples

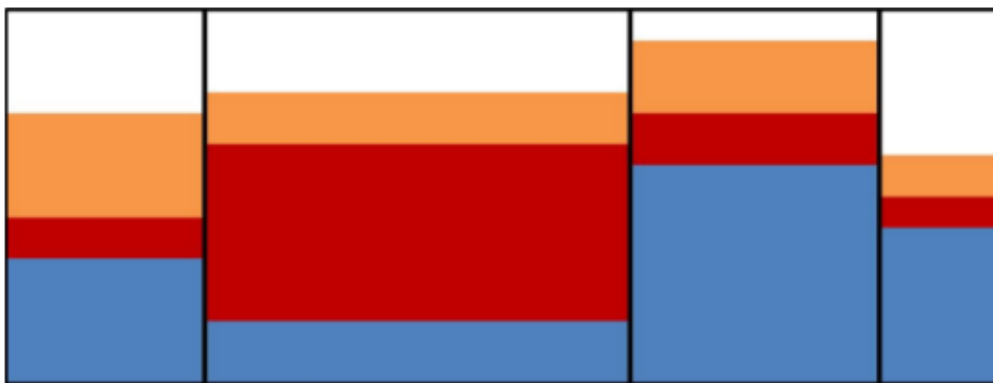
standard input	standard output
5 3 10 5 4 8 3	2
5 3 10 11 12 13 14	0

Problem F. Sand Art

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

На выставках часто предлагают самим изготовить цветные бутылочки с песком. Обычно процесс изготовления заключается в следующем: в бутылку насыпаются слои песка разной толщины и разного цвета. Однако на выставке кубистов был предложен нестандартный аттракцион: вместо бутылки используется стеклянный ящик прямоугольной формы.

Ящик имеет толщину в 1 единицу; внутри ящика есть $n - 1$ вертикальные перегородки, которые делят его на n вертикальных секций. На рисунке ящик разделён тремя перегородками на четыре секции.



Часто покупатель затрудняется в выборе композиции, однако знает, какое наименьшее и наибольшее количество песка каждого цвета он хочет видеть в каждой секции. Ваша задача — подобрать наиболее сбалансированный вариант, то есть такой, в котором разность между высотой столбиков песка в наиболее заполненной и наименее заполненной секциях минимальна.

Input

В первой строке входа заданы четыре целых числа n , m , w , h :

- n ($2 \leq n \leq 200$) — количество секций;
- m ($1 \leq m \leq 200$) — количество цветов песка;
- w , h ($1 \leq w, h \leq 5000$) — ширина и высота ящика (толщина ящика равна 1)

Далее во входе используются вещественные числа, каждое из которых содержит не более 3 цифр после десятичной точки.

Следующая строка содержит m вещественных чисел v ($0 < v \leq w \cdot h$) — имеющиеся в наличии количество песка каждого цвета. Использовать весь песок вовсе не обязательно — всё определяется максимумом и минимумом.

Следующая строка содержит $n - 1$ вещественное число x ($0 < x < w$), задающее расстояние от левой стенки до каждой перегородки. Гарантируется, что значения x отсортированы по возрастанию.

Каждая из последующих n строк содержит m вещественных чисел min ($0 \leq min \leq w \cdot h$). j -е число в i -й строке — минимальное количество песка цвета j , которое покупатель хочет видеть в секции i .

Каждая из последующих n строк содержит m вещественных чисел max ($0 \leq max \leq w \cdot h$). j -е число в i -й строке задаёт максимальное количество песка цвета j , которое покупатель хочет видеть в секции i ($min_{ij} \leq max_{ij}$)

Output

Выведите минимальную разность между максимальной и минимальной высотой песка в секциях с абсолютной точностью 10^{-3} . Гарантируется, что хотя бы одна композиция, удовлетворяющая ограничениям, существует.

Examples

standard input	standard output
2 2 5 5 2.0 2.0 4.0 1.0 0.0 0.0 1.0 1.0 0.0 0.0 2.0	0.75
2 2 5 5 2.0 2.0 4.0 1.0 0.0 0.0 1.0 1.5 0.0 0.0 2.0	0.625
2 5 11 10 3.0 4.0 4.0 9.0 2.0 4.0 2.0 2.0 1.0 0.5 0.25 0.0 2.0 0.0 4.0 1.0 2.0 2.0 3.0 4.0 0.75 0.0 2.1 0.0 5.1 1.1	0.266

Problem G. String Stretching

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Рассмотрим следующий процесс. Изначально задана строка p . Построим строку s следующим образом: начнём с пустой строки, вставим p ; затем выберем некоторую позицию в строке (в начале, в конце или между двумя символами) и вставим p ; затем повторим это действие (выбор позиции и вставку) некоторое количество раз.

Пусть p — строка “hello”. Начиная с пустой строки, строка s может иметь следующий вид (вставленная на текущем шаге строка p выделена:

- 1.
2. **hello**
3. **h**helloello
4. hh**elloel**hellolo
5. h**he**hellolloel**hell**olo

Таким образом, после пяти шагов строка s имеет вид “hhehellolloelhellolo”.

По заданной строке s найдите кратчайшую строку p , из которой могла быть построена строка s . Если таких строк несколько, выведите лексикографически минимальную.

Input

Вход состоит из строки s . Строка непуста, содержит только строчные латинские буквы, а её длина не превосходит 200.

Output

Выведите требуемую строку p .

Examples

standard input	standard output
hhehellolloelhellolo	hello

Problem H. Vending Machine

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Пройдоха Боб оказался около торгового автомата и стал наблюдать за покупателями. Спустя некоторое время он заметил, что торговый автомат сломан. Вот что увидел Боб:

1. Покупатель выбирает снек;
2. Автомат проверяет, остался ли выбранный снек;
3. Если хотя бы один снек остался, машина принимает оплату за выбранный снек;
4. После того, как деньги внесены, машина выдаёт не тот снек, который был выбран; если соответствующие снеки закончились, покупатель может вообще ничего не получить.

Пройдоха Боб заметил, что поломка является регулярной в том смысле, что при выборе снека i машина выдаёт некоторый снек $f(i)$; в частности, иногда машина может выдать заказанный снек, если для соответствующего i $f(i) = i$. Более того, Боб выяснил значение $f(i)$ для всех i ...

Боб собирается сделать небольшой бизнес: он хочет купить несколько снеков, после чего продать их на ближайшем хоккейном матче по актуальной рыночной стоимости (которая может отличаться от установленной в автомате — даже исправном — цены). Если дешёвый снек находится на позиции i , а дорогой на позиции $f(i)$, то бизнес будет ещё более доходным!

Считая, что покупателей на хоккейном матче будет достаточно, чтобы Боб продал любое количество снеков, найдите максимальную прибыль, которую Боб может получить, покупая несколько (возможно, 0) снеков в данном автомате и продавая их на хоккейном матче. Считается, что денег у Боба хватит, чтобы купить все снеки, находящиеся в данном автомате (если этого потребует бизнес).

Input

В первой строке входа задано целое число n ($1 \leq n \leq 10^5$) — количество различных снеков в автомате. i -я из последующих n строк содержит 4 целых числа f, p, m и s , описывающих параметры i -го снека следующим образом:

- f ($1 \leq f \leq n$) — значение $f(i)$, то есть номер снека, который будет выдан при выборе этого снека (и оплате за него);
- p ($1 \leq p \leq 10^6$) — цена снека в автомате;
- m ($1 \leq m \leq 10^6$) — рыночная цена снека во время матча;
- s ($1 \leq s \leq 10^6$) — количество снеков данного вида.

Output

Выведите одно целое число — максимальную прибыль, которую пройдоха Боб может получить, покупая снеки в данном автомате и продавая их на матче.

Examples

standard input	standard output
3 1 2 3 1 2 3 4 1 3 4 5 1	3
3 2 2 3 8 3 1 5 6 1 9 4 7	39
5 5 9 2 2 1 1 7 4 2 3 6 3 2 2 9 6 1 4 5 1	22

Problem I. Rainbow Zamboni

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

РасМан-у надоело бегать от монстров; так что он решил пойти на каток и потренироваться там. Однако каток надо чистить. В мире РасМан-а каток, как и всё остальное, является тороидальным, то есть если Вы выезжаете за пределы катка при движении направо, вы возвращаетесь на том же уровне, но слева, если Вы выезжаете налево — то справа, если выезжаете при движении вверх — возвращаетесь на том же расстоянии от левого края, но снизу, при движении вниз — на том же расстоянии от левого края, но сверху. Каток разбит на единичные квадраты; развёртка катка представляет собой прямоугольник размера $r \times c$ (r строк и c столбцов).

Машина для очистки льда обладает следующим свойством: когда она проезжает по льду, она не только очищает соответствующий квадрат, но и красит лёд некоторым цветом. Перед началом процесса чистки лёд является бесцветным. Когда машина проезжает по квадрату, она перекрашивает его в некоторый цвет. Каждый цвет задаётся заглавной буквой алфавита. Цвета меняются в алфавитном порядке, то есть при каждой смене цвета машина меняет его на следующий (при этом цвета зациклены, то есть если следующим цветом за Р является Q, то следующим за Z является A).

Алгоритм, по которому машина передвигается по катку, записывается следующим образом:

```
stepSize = 1
повторить numSteps раз
    Пройти stepSize шагов в текущем направлении
    Повернуться на 90 градусов по часовой стрелке
    Переключиться на следующий цвет
    stepSize = stepSize + 1
конец цикла
```

Первоначально машина направлена вверх. По заданной точке старта машины, размеру катка и количеству шагов *numSteps*, сделанных машиной, определите раскраску катка на момент окончания движения машины.

Input

Входной файл состоит из одной строки, содержащей 5 целых чисел r, c, i, j, n , где:

- r ($1 \leq r \leq 2000$) — количество строк в развёртке катка;
- c ($1 \leq c \leq 2000$) — количество столбцов в развёртке катка;
- i ($1 \leq i \leq r$) — строка, в которой стартует машина;
- j ($1 \leq j \leq c$) — столбец, в котором стартует машина;
- n ($1 \leq n \leq 10^{18}$) — количество шагов (*numSteps* в записи алгоритма).

Output

Выведите состояние катка в виде прямоугольника $r \times c$ символов. Финальное положение машины обозначьте символом '@', окрашенный квадрат обозначьте соответствующей заглавной буквой ('A'-'Z'), оставшийся неокрашенным (и нечищенным) — символом ('.').

Examples

standard input	standard output
5 5 3 3 4BBC ..A.C ...C @DDDD
5 5 3 3 7	EG... E@BBC EGA.C EG..C FGFFF

Problem J. Zig Zag Nametag

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

На слётах ниндзя уровень секретности настолько высок, что на бейджах указываются не реальные имена, а псевдонимы. Один ниндзя захотел прозвонить впечатление на своего сенсэя. Он знает, какое число сенсэй считает “счастливым”, так что ниндзя хочет писать на бейдже имя, которое кодирует это “счастливое число” следующим образом.

Пусть надпись на бейдже состоит только из строчных латинских букв. Присвоим каждой букве значение, равное её номеру в алфавите при нумерации с единицы (то есть $a = 1, b = 2, \dots, z = 26$). Числовое значение имени равно сумме модулей разностей значений пар соседних букв; например, для строки “azxb” её числовое значение равно

$$|a - z| + |z - x| + |x - b| = |1 - 26| + |26 - 24| + |24 - 2| = 49$$

Требуется найти кратчайшую строку, значение которой равно заданному “счастливому числу” k . Если строк минимальной длины более одной, то ниндзя хочет выбрать ту, которая идёт раньше по алфавиту.

Input

Вход состоит из одной строки, задающей одно целое число k ($1 \leq k \leq 10^6$) — “счастливое число” сенсэя. Гарантируется, что всегда существует как минимум одна строка, значение которой равно заданному числу.

Output

Выведите кратчайшую строку, значение которой равно заданному числу; в случае, если решений несколько, выведите лексикографически наименьшее.

Examples

standard input	standard output
1	ab
19	at
77	aoazb