

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 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

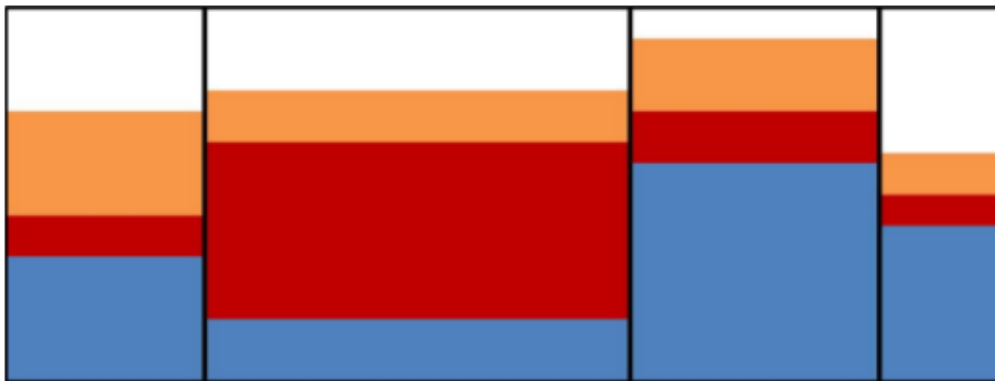
standard input	standard output
5 3 10 5 4 8 3	2
5 3 10 11 12 13 14	0

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 ($0 < v \leq w \cdot h$) — имеющееся 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 максимально 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 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 заиклены, 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.CC @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

Problem K. Knight Jumps

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

Шахматный конь ходит на два поля в одном направлении, затем на одно поле в перпендикулярном. При этом для того, чтобы ход был возможен, достаточно того, чтобы то поле, на которое делается ход, было свободно.

Вам дана доска $m \times n$, поле, на котором находится конь и поле, на которое он должен пойти. Некоторые поля этой доски заняты, и коня туда ставить нельзя. Выясните, за какое минимальное количество ходов конь сможет добраться до цели (если он вообще туда сможет добраться).

Input

В первой строке входа содержатся два целых числа n и m ($2 \leq n, m \leq 100$), задающие высоту и ширину доски. Каждая из последующих n строк содержит ровно m символов, задающих доску; допустимыми символами являются '.' — обозначение для свободного поля, '#' — для занятого поля, 'K' для поля, в котором расположен конь и 'X' для поля, куда конь должен прийти. Гарантируется, что в каждом входном файле будет ровно один символ 'K' и ровно один символ 'X'.

Output

Выведите одно число — наименьшее количество ходов, за которое конь сможет добраться до цели, или -1 , если не существует способа до неё пойти.

Examples

standard input	standard output
5 4 K...#.. .#.. ...X	5
3 3 K.. .X.. ...	-1

Problem L. Lost Digit

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

Известный археолог попросил Вас помочь ему в расшифровке рунической записи чисел. Археологу известно, что применялась десятичная система счисления, что использовались только целые числа и что в записи чисел не встречались ведущие нули (кроме ситуации с записью числа 0, записываемого одной цифрой 0). Он самостоятельно нашёл соответствие рунам нескольких цифр, а также некоторых знаков арифметических операций.

Очередная найденная им запись представляет собой простейшее математическое выражение в формате (операнд1) (операция) (операнд2) = (результат). В качестве операций могут использоваться только сложение (+), вычитание (-) и умножение (*). Также в записи чисел может использоваться унарный минус; числа принимают значение от -999999 до 999999. Одна из цифр, встречающихся в записи, на данный момент неизвестна профессору; она обозначается знаком вопроса (?). При этом неизвестная цифра ровно одна, и все её вхождения в выражение заменены на знак вопроса.

По заданному выражению найдите минимальную цифру, на которую можно заменить все знаки вопроса так, чтобы выражение было верным. Если такой цифры не существует, выведите -1.

Input

На входе задаётся одна строка — выражение в форме $\langle number \rangle \langle op \rangle \langle number \rangle = \langle number \rangle$. $\langle number \rangle$ состоит только из цифр '0'-'9', а также, возможно, знаков вопроса и ведущего минуса. Число может начинаться с 0 только в случае, если оно равно 0 (в этом случае 0 — единственный символ в записи); число не может начинаться с -0, также число не может состоять более, чем из шести цифр, включая и знаки вопроса. $\langle op \rangle$ разделяет первое и второе числа, и может быть одним из знаков '+', '-' или '*', обозначая сложение, вычитание или умножение соответственно. Знак равенства '=' всегда разделяет второе и третье числа. Гарантируется, что запись не содержит пробельных символов; также гарантируется, что в строке присутствует как минимум один знак вопроса.

Output

Выведите одно число, равное минимальной цифре, при подстановке которой вместо знака вопроса выражение остаётся верным; если такой цифры не существует, выведите -1.

Examples

standard input	standard output
1+1=?	2
123*45?=5?088	6
-5?* -1=5?	0
19--45=5?	-1
??*??=302?	5

Problem M. Multiple Top N

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

Среди обозревателей, следящих за многочисленными студенческими лигами в США, популярно составление различных рейтингов типа Топ 25 (или, в общем случае, Топ n) команд в стране. Учитывая, что команды из разных лиг между собой могут и не встречаться, данные списки субъективны и часто различаются, однако в целом они более или менее похожи. Ваша задача — сравнить два таких списка одинаковой длины, составленные из одних и тех же команд, и выяснить, в чём они похожи.

Более формально, Вы должны разбить списки на непустые множества, состоящие из одинакового числа позиций в обоих списках и обладающие следующим свойством: набор команд, занимающих эти позиции в обоих списках, одинаков. При этом каждое множество должно содержать как можно меньше команд. Например, для совпадающих списков длины n Вы должны получить n множеств, каждое из которых содержит одну позицию (так как в совпадающих списках все позиции совпадают).

Input

Первая строка входа содержит целое число n ($1 \leq n \leq 10^6$) — количество команд в списках. Следующие n строк содержат первый список, перечисленный сверху вниз по одной команде в строке. Последующие n строк содержат в аналогичном формате второй список. Названия команд непусты, состоят из не более, чем из 8 заглавных латинских букв и не содержат других символов. Первый и второй списки состоят из одних тех же команд; внутри списков имена команд не повторяются.

Output

Выведите размер каждого множества в порядке следования сверху вниз, по одному в строке.

Examples

standard input	standard output
5 A B C D E A C D B E	1 3 1
3 FIRST SECOND THIRD FIRST SECOND THIRD	1 1 1
UNITED CITY CHELSEA CHELSEA UNITED CITY	3

Problem N. New Contest Director

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

Только что закончилось голосование на выборах нового директора регионального конкурса. Поданные голоса уже переписаны в отдельный файл и избирательная комиссия поручила Вам посчитать голоса и сообщить имя того, кто набрал наибольшее количество голосов. Если два и более кандидата набрали одинаковое число голосов, Вы должны вывести имена всех этих кандидатов в алфавитном порядке.

Input

В первой строке входа задано одно целое число n ($1 \leq n \leq 1000$) — общее количество голосов. Каждая из последующих n строк содержит имя кандидата, за которого подан голос; имя состоит только из заглавных латинских букв, непусто и имеет длину не более 20 символов.

Output

Выведите имя кандидата, набравшего наибольшее количество голосов. В случае ничьей выведите имена всех кандидатов, разделивших первое место, в алфавитном порядке по одному имени на строку.

Examples

standard input	standard output
5 BILL MIKE BILL BILL MIKE	BILL
5 BILL MARSHA BILL MARSHA ANONYMOUS	BILL MARSHA