Задача A. Файлы

Имя входного файла:	`files.in`
Имя выходного файла:	`files.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сергей увлекается цифровой фотографией. Недавно он решил распечатать часть своих снимков и повесить их на стену в прихожей. Для этого он выписал имена файлов всех фотографий и пронумеровал их в порядке появления в этом списке.

Номер	Имя файла
1	idx1001.jpg
2	idx1000.jpg
3	idx1014.jpg

После этого Сергей несколько часов выбирал лучшие фотоснимки. Обычно Сергею нравилось сразу несколько подряд идущих фотографий, поэтому он выписывал номер первой и последней из них. В итоге у него получился список интервалов номеров понравившихся ему фотографий.

Номер первой фотографии	Номер последней фотографии	Имена соответствующих файлов
2	3	idx1000.jpg idx1014.jpg
1	2	idx1001.jpg idx1000.jpg

Теперь Сергей хочет получить список имен файлов всех понравившихся ему фотографий. В этом списке сначала должны быть перечислены в порядке возрастания номеров имена файлов фотографий из первого интервала, затем из второго и т.д.

Имена файлов
idx1000.jpg
idx1014.jpg
idx1001.jpg
idx1000.jpg

Этот список может быть очень большим, поэтому Сергей просит вас ему помочь.

Формат входного файла

Первая строка входного файла содержит целое число n — число фотографий ($1 \le n \le 1000$). Следующие n строк содержат имена файлов фотографий, состоящие из строчных и прописных латинских букв, цифр и символов «_» и «.». Длины имен файлов не превышают 100 символов. Следующая строка содержит m — число интервалов номеров фотографий ($1 \le m \le 1000$). Следующие m строк содержат по два целых числа l_i и r_i — номер первой и последней фотографии в i-м интервале ($1 \le l_i \le r_i \le n$).

Формат выходного файла

Выведите искомый список, по одному имени файла на строке.

Примеры

files.in	files.out
3 idx1001.jpg idx1000.jpg idx1014.jpg 2 2 3 1 2	idx1000.jpg idx1014.jpg idx1001.jpg idx1000.jpg

Примечание

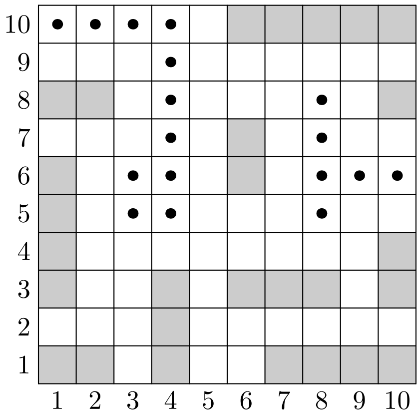
Решения, правильно работающие только при $l_i = r_i$, будут оцениваться 40 баллов.

Задача B. Морской бой

Имя входного файла:	`battleship.in`
Имя выходного файла:	`battleship.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Маленький Ян пишет искусственный интеллект для игры в «морской бой».

«Морской бой» — игра для двух игроков, в которой у каждого игрока есть клетчатое прямоугольное поле размером $n \times m$, на котором они располагают несколько «кораблей». Каждый корабль — это горизонтальный прямоугольник с высотой 1 или вертикальный прямоугольник с шириной 1, занимающий целиком несколько клеток. Корабли должны быть расположены таким образом, чтобы не иметь общих точек даже на границе.



Один из модулей программы Яна должен в некоторый момент игры рассчитывать количество возможных расположений вражеского корабля размером 1×1. В этот момент игры уже известны положения некоторых других кораблей и необходимо проверить, не нарушил ли правила расположения кораблей соперник, и если не нарушил, то сколько существует возможных позиций для расположения еще одного единичного корабля.

Помогите маленькому Яну написать эту часть программы.

Формат входного файла

Первая строка входного файла содержит три целых числа n, m и k ($1 \le n, m \le 1000$; $0 \le k \le 10000$) — размеры игрового поля и количество известных вражеских кораблей.

Следующие k строк содержат по четыре целых числа x_1, y_1, x_2, y_2 ($1 \le x_1 \le x_2 \le n$; $1 \le y_1 \le y_2 \le m$) — координаты клеток, в которых расположены левый верхний и правый нижний углы корабля, соответственно.

Если корабль расположен вертикально, то $y_1 = y_2$, если горизонтально, $x_1 = x_2$.

Формат выходного файла

Если противник нарушил правила расположения кораблей (у двух кораблей есть общие точки), то выведите в выходной файл единственное слово «`INCORRECT`».

Иначе выведите единственное целое число — количество возможных расположений вражеского единичного корабля.

Примеры

battleship.in	battleship.out
10 10 10 1 1 1 2 1 4 3 4 1 7 1 10 3 1 6 1 3 6 3 8 3 10 4 10 6 6 7 6 8 1 8 2 8 10 8 10 10 6 10 10	17
3 3 2 1 1 1 1 2 2 2 3	INCORRECT

Примечание

На рисунке точками изображены возможные расположения вражеского корабля размером 1×1 в первом примере.

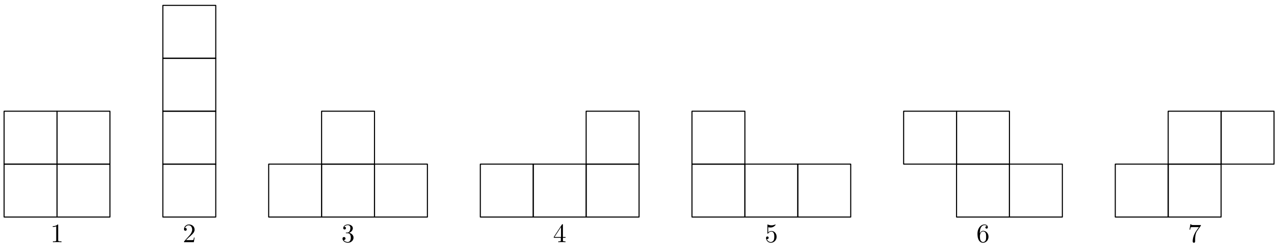
Решения, работающие при $n, m, k \leq 100$, будут оцениваться в 50 баллов.

Задача C. Тетрис

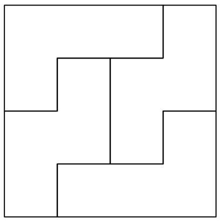
Имя входного файла:	`tetris.in`
Имя выходного файла:	`tetris.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Маленький Ян очень много играл в компьютерные игры, поэтому родители запретили мальчику играть в его любимый «Тетрис».

Но Ян не отчаивается. Он смастерил свою собственную игру — «Настольный Тетрис». Правила этой игры очень просты. Игра производится на прямоугольном поле размера $n \times m$. Изначально некоторые клетки поля заняты, а остальные свободны. Игроку требуется набором фигурок из тетриса покрыть все свободные клетки, при этом фигурки не должны накладываться друг на друга или на уже занятые клетки. Так как все фигурки с одной стороны покрашены, а с другой нет, то переворачивать их нельзя, однако можно их поворачивать.



Изначально Ян хотел выпилить бесконечно много фигурок каждого типа, но он очень быстро устал, поэтому у него есть только a_i фигурок типа i.



Ян смастерил несколько полей для игры, и теперь ему интересно, можно ли их покрыть фигурками. Помогите Яну решить эту задачу.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($1 \le n, m \le 6$) — размеры игрового поля. В следующей строке заданы семь целых чисел a_i ($0 \le a_i \le 10$). Следующие n строк входного файла содержат описание поля. Каждая строка содержит m символов. Символ '`.`' означает, что клетка свободна, '`#`' — занята.

Формат выходного файла

Если покрыть свободные клетки фигурками нельзя выведите в выходной файл единственное слово «`NIE`» (нет по-польски). Иначе в первой строке выведите единственное слово «`TAK`» (да по-польски). В следующих n строках выведите описание покрытого поля. Каждая строка описания должна содержать m целых чисел — номера фигурок, которыми покрыты соответствующие клетки поля. Фигурки должны иметь номера от 1 до 9, при этом разные фигурки должны иметь разные номера. Для изначально занятых клеток требуется выводить 0.

Примеры

tetris.in	tetris.out
4 5 3 0 0 0 0 0 0 ....# ....# #..## #..##	TAK 1 1 2 2 0 1 1 2 2 0 0 3 3 0 0 0 3 3 0 0
4 5 2 1 0 0 0 0 0 ....# ....# #..## #..##	NIE
4 4 1 1 0 3 0 2 0	TAK 1 1 1 3 1 2 3 3 2 2 3 4 2 4 4 4

Примечание

Решения, работающие в случае наличия фигур только первого типа, будут оцениваться в 30 баллов.

Задача D. Тикал

Имя входного файла:	`tikal.in`
Имя выходного файла:	`tikal.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В городе Тикал индейцы майя построили прекрасный храм. Храм представляет собой правильный n-угольник, все стороны которого неотличимы. По сложившейся традиции в храм были занесены k одинаковых фигурок идолов. Шаманы утверждают, что фигурки должны стоять у стен храма — сторон n-угольника. При этом у разных стен может стоять разное число фигурок. По обычаям индейцев каждое утро Главный Шаман должен переставлять фигурки в какую-то новую конфигурацию.

После постройки храма духи явились индейцам и сообщили, что, как только конфигурация фигурок в храме повториться, небеса упадут на землю, моря выйдут из берегов и случится конец света. Также духи уточнили, что конфигурации идолов, получаемые поворотом храма считаются одинаковыми.

Индейцы очень не хотят конца света, поэтому Главный Шаман просит сообщить ему, сколько существует различных конфигураций фигурок идолов в храме. Так как это число может быть очень большим, Главный Шаман хочет знать его по модулю простого числа p.

Формат входного файла

В первой строке входного файла заданы три целых числа n, k и p ($3 \leq n \leq 500\,000$; $1 \leq k \leq 500\,000$; $10^6 < p < 10^9$). Гарантируется, что число p — простое.

Формат выходного файла

В выходной файл выведите единственное целое число — число различных расстановок k идолов в храме с n стенами с точностью до поворота по модулю p.

Примеры

tikal.in	tikal.out
`4 1 1000003`	`1`
`6 3 1000003`	`10`

Примечание

Решения, работающие при $n, k \leq 10$, будут оцениваться в 40 баллов.

Возможные расположения идолов во втором примере:

(3,0,0,0,0,0) (2,1,0,0,0,0) (2,0,1,0,0,0) (2,0,0,1,0,0) (2,0,0,0,1,0)
(2,0,0,0,0,1) (1,1,1,0,0,0) (1,1,0,1,0,0) (1,1,0,0,1,0) (1,0,1,0,1,0)