Задача A. Морской бой — 3

Имя входного файла:	`battle3.in`
Имя выходного файла:	`battle3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Петя с Васей играют в «Морской бой». Как известно, «Морской бой» — это игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника (говорят, что игрок произвел выстрел в эту клетке). Если у соперника по этим координатам имеется корабль, то корабль или его часть «топится», а попавший получает право сделать еще один ход. Цель игрока — первым поразить все корабли противника.

Игра у Васи с Петей происходит на клетчатом поле размером $n \times m$, причем на этом поле размещены корабли, имеющие форму прямоугольников размером $1 \times k$ или $k \times 1$. Уже сделано несколько ходов в игре, причем все выстрелы, которые произвел Вася оказались неудачными, то есть он ни разу не назвал клетку, в которой находится часть корабля, который поставил Петя.

Для того, чтобы прикинуть свои шансы на победу в игре, Вася хочет найти число способов поставить на поле один корабль размером $1 \times k$ или $k \times 1$. При этом ни в одну клетку, которую будет занимать корабль, не должен был быть Васей произведен выстрел.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($2 \le n, m \le 100$). Последующие n строк описывают поле — каждая из них содержит по m символов: j-ый символ i-ой из этих строк есть «`#`», если в ходе игры Вася производил выстрел в эту клетку, и «`.`» — в противном случае.

Последняя строка входного файла содержит целое число k ($1 \le k \le 5$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

battle3.in	battle3.out
4 4 2	24
3 2 .. .# ## 2	2

Задача B. Четырехугольник

Имя входного файла:	`quadr.in`
Имя выходного файла:	`quadr.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

На плоскости заданы четыре точки (возможно, некоторые из них совпадают). Необходимо выяснить, являются ли они четырьмя вершинами некоторого квадрата. Стороны этого квадрата необязательно должны быть параллельны осям координат.

Формат входного файла

Входной файл содержит четыре строки, каждая из которых содержит по два целых числа — координаты одной из заданных точек. Все числа во входном файле не превосходят 1000 по абсолютной величине.

Формат выходного файла

Если заданные точки являются вершинами некоторого квадрата, то выведите в выходной файл слово `YES`, иначе — выведите слово `NO`.

Примеры

quadr.in	quadr.out
0 0 1 1 0 1 1 0	YES
0 0 0 0 1 1 0 1	NO

Задача C. Робот

Имя входного файла:	`robot.in`
Имя выходного файла:	`robot.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Робот *P-2008-2009* предназначен для исследования просторов Флатландии, которые, как известно, представляют собой части плоскости, разбитые на единичные квадраты (клетки) вертикальными и горизонтальными прямыми. Программы для этого робота достаточно просты, так как написаны на языке программирования, который содержит всего четыре команды.

Эти команды таковы:

- сдвинуться на клетку вверх — `U`;
- сдвинуться на клетку вниз — `D`;
- сдвинуться на клетку влево — `L`;
- сдвинуться на клетку вправо — `R`.

Ваша задача состоит в написании программы, которая будет анализировать некоторые свойства программы для робота P-2008-2009, — а именно, предположим, что исследуемая область представляет собой бесконечную во всех четырех направлениях плоскость. Задана программа для робота P-2008-2009. Необходимо найти число клеток плоскости, которые он посетит более одного раза.

Заметим, что это число не зависит от того, в какой клетке изначально находится робот.

Формат входного файла

Первая строка входного файла содержит программу для робота. Она состоит только из символов `U`, `D`, `L`, `R`. Ее длина положительна и не превосходит 1000 символов.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

robot.in	robot.out
ULDR	1
URLD	2

Задача D. Последовательность

Имя входного файла:	`sequence.in`
Имя выходного файла:	`sequence.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Вася очень любит последовательности. Целыми днями он выписывает их в тетрадке на полях во время уроков. Но Вася выписывает не случайные числа. Вася выписывает последовательность, заданную следующими соотношениями:

$$a_1 = w$$

$$a_i = (x \cdot a_{i-1} + y) \mod z, \text{ если } i > 1$$

Здесь x, y, z и w — некие натуральные числа, которые первыми пришли Васе на ум. Не один и не два, а три дня Вася выписывал на полях свою последовательность и, наконец, понял, что выписал n чисел.

И все бы хорошо, если бы Васин друг Петя, не спросил, какое число является k-ым по порядку, если эту последовательность упорядочить по неубыванию.

И тут на Васю напала лень, поэтому он обратился к Вам за помощью. Ответьте на Петин вопрос.

Формат входного файла

В первой строке находятся четыре числа x, y, z и w. Эти числа положительны и не превосходят 10000. Во второй строке находятся два целых числа n и k ($1 \leq k \leq n \leq 10^5$).

Формат выходного файла

В выходной файл выведите ответ на Петин вопрос.

Примеры

sequence.in	sequence.out
1 1 100 2 5 2	3
1 1 100 200 3 2	2

Задача E. Клавиша Shift

Имя входного файла:	`shift.in`
Имя выходного файла:	`shift.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

У Коли беда. На его новенькой клавиатуре перестала работать клавиша *Shift*. За время использования клавиатуры Коля всего лишь набрал свое сочинение по английскому языку, используя при этом обычную английскую раскладку и только левый *Shift* (Коля пока еще достаточно плохо набирает на клавиатуре). При покупке клавиатуры Коле сообщили, что каждая клавиша выдерживает $\frac{2^{2^{2^2}}-1}{3^{1^{3^1}}}$ нажатий.

Теперь Коле интересно, какое минимальное число раз он мог нажать *Shift*, чтобы набрать тот же текст, чтобы знать может ли он потребовать замену клавиатуры. При написании Коля мог бы зажимать *Shift* и поочередно нажимать несколько других клавиш, не отпуская *Shift*, и это считалось бы одним нажатием. Клавишей *Caps Lock* Коля не пользуется принципиально.

Формат входного файла

Во входном файле дан текст набранный Колей. Он состоит из строчных и заглавных латинских букв, пробелов и переводов строк. Размер входного файла на превышает 10^3 байт.

Формат выходного файла

В выходной файл выведите одно число — минимальное необходимое число нажатий на *Shift*.

Примеры

shift.in	shift.out
`TestLib`	`2`
`PCMS`	`1`
`PCMS Client`	`1`

Задача F. Счастливый билетик — 3

Имя входного файла:	`ticket3.in`
Имя выходного файла:	`ticket3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Назовем *билетиком* последовательность цифр длины $2 \cdot n$. Билетик A называется *счастливым*, если выполняется хотя бы один из следующих критериев:

- сумма цифр первой половины билетика равна сумме цифр второй половины $(\sum\limits_{i=1}^{n} a_i = \sum\limits_{i=n+1}^{2n} a_i)$;
- сумма цифр на нечетных позициях равна сумме цифр на четных позициях $(\sum\limits_{i=1}^{n} a_{2i} = \sum\limits_{i=1}^{n} a_{2i-1})$;
- сумма четных цифр билетика равна сумме его нечетных цифр.

Ваша задача — проверить, является ли заданный билетик счастливым.

Формат входного файла

Первая строка входного файла содержит число n $(1 \le n \le 5000)$. Во второй строке содержатся цифры a_1, a_2, $\ldots$, a_{2n} $(0 \le a_i \le 9)$, разделенные пробелами.

Формат выходного файла

В выходной файл выведите `YES`, если билетик счастливый, и `NO` — в противном случае.

Примеры

ticket3.in	ticket3.out
3 2 1 1 1 1 2	YES
3 2 1 1 1 1 3	NO

Задача G. Трехдиагональная матрица

Имя входного файла:	`tridiag.in`
Имя выходного файла:	`tridiag.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Сегодня на уроке Станиславу рассказывали, что такое матрицы. К сожалению, большую часть урока Станислав играл в новую совершенно потрясающую игру для телефона, и совершенно ничего не запомнил. Поэтому когда он обнаружил, что на дом задали проверить, являются ли некоторые матрицы трехдиагональными, ему ничего не оставалось, кроме как идти за советом к лучшему другу, то есть к Вам.

Вы-то знаете, что квадратная матрица A называется *трехдиагональной*, если все ее ненулевые элементы находятся либо на главной диагонали, либо на диагоналях, находящихся непосредственно выше и ниже ее (то есть для любого ненулевого элемента A_{ij} верно, что $|i - j| \le 1$).

У Станислава не было времени разбираться в этом, так как он еще не успел пройти свою игру. Поэтому он попросил Вас написать для него программу, решающую за него задачу, заданную ему на дом.

Формат входного файла

В первой строке входного файла содержится целое число n — размер матрицы ($1 \le n \le 100$). Далее следует n строк, каждая из которых содержит n целых чисел. j-е число на i-й строке соответствует элементу матрицы A_{ij} ($-10000 \le A_{ij} \le 10000$).

Формат выходного файла

В выходной файл выведите «`YES`», если матрица, заданная во входном файле, является трехдиагональной, и «`NO`» в противном случае.

Примеры

tridiag.in	tridiag.out
2 1 1 1 1	YES
3 1 1 1 1 1 1 0 1 1	NO
2 1 0 1 1	YES
4 1 2 0 0 3 4 5 0 0 6 7 8 0 0 9 0	YES

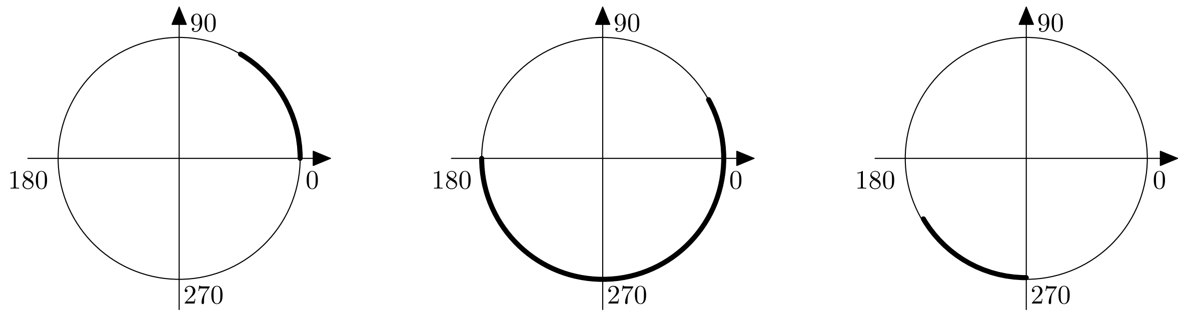
Задача H. Две дуги

Имя входного файла:	`twoarcs.in`
Имя выходного файла:	`twoarcs.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В этой задаче речь пойдет о дугах одной и той же окружности.

Каждая дуга будет задаваться градусными мерами ее двух концов. Эти градусные меры являются целыми числами от $0°$ до $359°$ включительно и измеряются против часовой стрелки от луча, направленного вправо. Так как в общем случае существуют две дуги с заданными концами, мы будем считать, что дуга проходит из первой точки в сторону возрастания угла (с учетом перехода через отметку $0°$).

На рисунке изображены дуги, задаваемые соответственно как $(0; 60)$, $(180; 30)$, $(210; 270)$:



В случае совпадения концов дуг условимся считать, что таким образом задана дуга, состоящая из одной точки.

Заданы две дуги окружности. Необходимо проверить, имеют ли они хотя бы одну общую точку.

Формат входного файла

В первой строке входного файла находится два целых числа S_1, F_1 — описание первой дуги. Во второй строке находится описание второй дуги в том же формате.

Все числа во входном файле неотрицательны и строго меньше 360.

Формат выходного файла

В выходной файл выведите «`YES`», если дуги имеют хотя бы одну общую точку, или «`NO`», если не имеют общих точек.

Примеры

twoarcs.in	twoarcs.out
0 60 210 270	NO
210 270 180 30	YES
180 30 0 60	YES