Задача A. Выпукая оболочка

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

Рассмотрим бесконечный лист клетчатой бумаги. Закрасим некоторое непустое множество клеток в черный цвет. Теперь мы хотим закрасить минимальное количество клеток, так, чтобы множество черных клеток стало выпуклым.

Напомним, что геометрическая фигура Φ называется выпуклой, если для любых точек $A \in \Phi$ и $B \in \Phi$ с вещественными координатами отрезок $[AB] \in \Phi$.

Формат входного файла

В первой строке входного файла содержатся два числа N $(1 \leq N \leq 100)$ и M $(1 \leq M \leq 100)$ — размеры куска бумаги, куда попали все черные клетки. В каждой из следующих N строк содержится M символов «`*`» или «`.`». Символ «*» обозначает черную клетку, а «.» белую.

Формат выходного файла

Выведете выпуклое множество, содержащее минимальное количество дополнительно покрашенных черных клеток, в ровно N строках по M символов «*» или «.» в каждой.

Примеры

convex.in	convex.out
2 4 ..*. .**.	.**. .**.
4 3 .*. .*. .*. .*.	.*. .*. .*. .*.

Задача B. Единичный НОД

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

Рассмотрим два числа вида (в десятичной системе счисления)

$$\underbrace{11\cdots11}_{n\text{ штук}} \text{ и } \underbrace{11\cdots11}_{m\text{ штук}}$$

Вам требуется найти НОД этих чисел.

Напомним, что НОД (наибольший общий делитель) двух чисел a и b — это такое максимальное число c, что b делится на c и a делится на c.

Формат входного файла

В единственной строке входного файла записаны два целых числа $N, M, 1 \le N, M \le 2000$.

Формат выходного файла

В единственной строке выведите ответ, без ведущих нулей.

Примеры

ones.in	ones.out
1 1	1
1 2	1

Задача C. Головоломка про ферзей

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

Многие из Вас, наверно, хотя бы раз в жизни играли в шахматы. Поэтому, Вы наверняка знаете, что ферзь — это шахматная фигура, которая может перемещаться по вертикали, горизонтали и диагонали на любое количество полей.

Слава недавно начал заниматься шахматами. На первом занятии учитель рассказал ему, как ходят шахматные фигуры. Слава сразу понял, что ферзь — самая "мощная" шахматная фигура.

Кроме обычной игры в шахматы, Славе сразу полюбились шахматные головоломки. В недавно купленной им книге «Шахматные головоломки для детей от 9 до 99» он обнаружил такую головоломку: "Расставить k ферзей на доске 8 на 8 так, чтобы хотя бы одно свободное поле не было атаковано."

Напишите программу, которая поможет Славе решить эту головоломку.

Поле называется *атакованным*, если хотя бы одна фигура на доске может совершить ход на это поле.

Формат входного файла

Входной файл содержит натуральное число k $(0 \le k \le 64)$.

Формат выходного файла

В первой строке выходного файла выведите слово YES, если искомая расстановка возможна, и слово NO в противном случае. В случае положительного ответа, выведите соответствующую расстановку.

Для этого выведите 8 строк по 8 символов в каждой. Строки соответствуют горизонталям шахматной доски, столбцы — вертикалям. Пустое поле обозначайте символом . (точка), поле, на котором стоит ферзь — символом Q.

Если существует несколько расстановок ферзей, выведите любую.

Примеры

<table>
<tr><th>queens.in</th><th>queens.out</th></tr>
<tr><td>1</td><td>YES
....
....
....
....
...Q....
....
....
........</td></tr>
<tr><td>60</td><td>NO</td></tr>
<tr><td>5</td><td>YES
Q......Q
....
....
.......Q
....
....
....
Q......Q</td></tr>
</table>

Задача D. Строки

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

Задано множество строк $S = \{s_1, s_2, s_3, \ldots, s_n\}$. Необходимо найти количество строк $s_i \in S$, представимых в виде конкатенации двух строк $s_i \in S, s_k \in S$ ($s_i = s_j s_k$, j и k при этом могут совпадать).

Формат входного файла

Входной файл содержит множество S — по одному элементу на строке. i-ая строка входного файла содержит s_i. Последняя строка входного файла содержит строку `ENDOFINPUT`. Она обозначает конец входных данных и не входит в множество S.

Все s_i состоят только из маленьких букв латинского алфавита и имеют длину не более 100 символов. Во входном файле не более 239 строк.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

strings.in	**strings.out**
aa aaaa ab abaa ENDOFINPUT	2
abc bcd def ENDOFINPUT	0

Задача E. Судоку

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

Судоку размера n называется квадрат со стороной n^2, разделенный на n^2 средних квадратов со стороной n, каждый из которых разделен на n^2 маленьких квадратов. В каждом маленьком квадрате записано число от 1 до n^2.

Судоку называется *правильным*, если в каждом столбце, каждой строке и каждом среднем квадрате встречаются все числа от 1 до n^2.

Недавно Вася нарисовал Судоку размера n. Ваша задача — помочь ему определить правильный ли он.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 10$). В следующих n^2 строчках содержится по n^2 чисел, задающих нарисованный Васей Судоку.

Все числа во входном файле целые и не превосходят 10^9 по модулю.

Формат выходного файла

Если Судоку правильный, то выведите слово «`Correct`», иначе выведите «`Incorrect`»

Примеры

sudoku.in	**sudoku.out**
3 1 3 2 5 4 6 9 8 7 4 6 5 8 7 9 3 2 1 7 9 8 2 1 3 6 5 4 9 2 1 4 3 5 8 7 6 3 5 4 7 6 8 2 1 9 6 8 7 1 9 2 5 4 3 5 7 6 9 8 1 4 3 2 2 4 3 6 5 7 1 9 8 8 1 9 3 2 4 7 6 5	Correct
1 10	Incorrect

Задача F. «Сердца столиц соединяя...»

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

Между двумя крупнейшими городами нашей страны Санкт-Петербургом и Москвой ежедневно совершают рейсы n поездов. Для каждого поезда известно его время отправления из Петербурга и время прибытия в Москву.

Найдите самый быстрый поезд и его скорость в предположении, что длина железной дороги между Санкт-Петербургом и Москвой равна 650 км.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 100$). Каждая из последующих n строк описывает ровно один поезд.

Описание поезда состоит из его название, времени отправления и времени прибытия. Название поезда — строка длиной не более 50 символов, заключенная в кавычки. Она может содержать буквы латинского алфавита, пробелы, цифры, символы тире («—») и подчеркивания («_»). Времена отправления и прибытия заданы в формате `ЧЧ:ММ`. Строчные и заглавные буквы в названиях поездов различаются.

Время в пути для каждого из поездов составляет хотя бы одну минуту и не превышает 24 часов.

Гарантируется, что самый быстрый поезд определяется единственным образом.

Формат выходного файла

В выходной файл выведите название самого быстрого поезда и его скорость. Скорость выводите в километрах в час и округляйте до целых. Следуйте формату вывода, приведенному в примерах.

Примеры

trains.in
```
3
"ER-200" 06:43 10:40
"Red Arrow" 23:55 07:55
"Express" 23:59 08:00
```
trains.out
```
The fastest train is "ER-200".
It’s speed is 165 km/h, approximately.
```
trains.in
```
3
"Train1" 00:00 00:00
"Train2" 00:00 00:01
"Train3" 00:01 00:01
```
trains.out
```
The fastest train is "Train2".
It’s speed is 39000 km/h, approximately.
```
trains.in
```
2
"Slow Train 1" 10:00 09:59
"Slow Train 2" 10:00 10:00
```
trains.out
```
The fastest train is "Slow Train 1".
It’s speed is 27 km/h, approximately.
```

Задача G. Треугольные страны

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

Эта история происходила на одной плоской планете. С незапамятных времен на ней существовал город N, находящийся в точке x_N, y_N. Кроме этого, в разное время на этой же планете существовали страны, каждая из которых имела форму треугольника.

Теперь перед историками встала серьезная задача — по имеющимся у них данным о треугольных странах определить, в какие страны мог входить город N. Город мог входить в страну, если он находится строго внутри нее.

Формат входного файла

Первая строка входного файла содержит два числа: x_N и y_N — координаты города N. Вторая строка входного файла содержит количество k треугольных стран $(1 \le k \le 1000)$. Последующие k строк каждая описывают одну треугольную страну. Описание треугольной страны состоит из шести целых чисел $x_1, y_1, x_2, y_2, x_3, y_3$, где (x_1, y_1), (x_2, y_2), x_3, y_3 — координаты вершин этой страны.

Гарантируется, что все страны имеют ненулевую площадь. Все координаты не превосходят 10000 по абсолютной величине.

Формат выходного файла

В первой строке выходного файла выведите количество стран, в которые мог входить город N. Во второй строке выведите через пробел номера этих стран в возрастающем порядке. Страны нумеруются с единицы в том порядке, в каком они заданы во входном файле.

Примеры

tri.in	**tri.out**
0 1 2 -2 0 2 0 0 2 -3 0 3 0 0 3	2 1 2
0 2 2 -2 0 2 0 0 2 -3 0 3 0 0 3	1 2

Задача H. Две последовательности

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

Определим последовательности a_n и b_n следующим образом: $a_1 = 2$, $a_2 = 3$, $a_3 = 4$, $a_4 = 7$, $a_n = b_{n-1} + b_{n-3}, n > 4$, b_n — последовательность чисел, не входящих в a_n, записанных в возрастающем порядке.

Таким образом, последовательность a_n будет выглядеть следующим образом: $2, 3, 4, 7, 13, 15, \ldots$, а последовательность b_n — $1, 5, 6, 8, 9, 10, \ldots$.

Ваша задача состоит в том, чтобы найти a_n и b_n.

Формат входного файла

Входной файл содержит целое число n $(1 \le n \le 10000)$.

Формат выходного файла

В первой строке выходного файла выведите a_n, во второй — b_n.

Примеры

twoseq.in	twoseq.out
4	7 8
10	25 16
6578	19731 9868
10000	29995 15000