

## Задача 1. Расстановка ферзей

Источник:	основная
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Дано прямоугольное клеточное поле из  $M$  строк и  $N$  столбцов. Нужно расставить ферзей на этом поле согласно правилам:

1. В каждой строке поля должен стоять один ферзь.
2. Никакие два ферзя не должны бить друг друга.

Ферзи на поле бьют всё, что стоит на той же горизонтали, вертикали или диагонали (как в шахматах). В некоторые клетки поля ставить ферзя запрещено.

### Формат входных данных

В первой строке дано два целых числа:  $M$  — количество строк и  $N$  — количество столбцов ( $1 \leq M, N \leq 12$ ).

В остальных  $M$  строках записана карта поля. Каждая из этих строк содержит ровно  $N$  символов. Символ вопроса '?' означает, что в этой клетке можно ставить ферзя, а можно не ставить. Символ точки '.' означает, что в этой клетке ферзя ставить запрещено.

### Формат выходных данных

Если искомого решения не существует, нужно вывести слово NO, и больше ничего не выводить.

Если решение существует, то нужно вывести слово YES в первую строку, и решение в остальные  $M$  строк. Решение должно быть выведено в виде карты поля, в том же формате, как во входных данных. Только каждый символ вопроса надо заменить на X (заглавную букву «икс»), если в этой клетке стоит ферзь, и точку — если не стоит.

Если решений несколько, можно вывести любое из них.

### Пример

input.txt	output.txt
5 8 ?.?...?? ?????.?? .....?. ???.??.? ?????..?	YES X..... ..X..... .....X. .X..... ...X....
3 3 ??? ??? ???	NO

## Задача 2. Выбор предметов

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Иммануил играет в компьютерную игру, вероятно собственного приготовления. На уровне игры разбросано  $N$  ценных предметов. Для каждого предмета известно, сколько он весит в граммах, и сколько золотых можно выручить при его продаже. К сожалению, персонаж Иммануила может нести только  $W$  граммов груза: больше таскать игра не позволяет.

Иммануилу интересно, какие предметы нужно взять, чтобы получить максимальное количество золотых при их продаже.

### Формат входных данных

В первой строке входного файла задано два целых числа:  $N$  — количество предметов и  $W$  — сколько граммов можно унести ( $1 \leq N \leq 20$ ,  $1 \leq W \leq 10^{18}$ ).

Далее описаны предметы, по одному в строке. Для каждого предмета указано два целых числа: его вес  $w_i$  в граммах и его цена  $c_i$  в золотых ( $1 \leq w_i, c_i \leq 10^{16}$ ).

### Формат выходных данных

В первую строку выходного файла требуется вывести два целых числа:  $A$  — сколько максимум золотых получится выручить, и  $K$  — сколько предметов при этом нужно брать.

Во второй строке должно находиться  $K$  целых чисел — номера предметов, которые нужно брать, чтобы получить  $A$  золотых.

Предметы нумеруются в порядке описания во входном файле, начиная с единицы. Номера предметов должны быть упорядочены по возрастанию. Гарантируется, что в каждом тесте есть только один правильный ответ.

### Пример

<code>input.txt</code>	<code>output.txt</code>
4 42 15 37 11 16 37 45 23 40	77 2 1 4

### Комментарий

**Подсказка:** рассмотреть все целые числа от 0 до  $2^N - 1$  включительно как битовые маски, то они будут задавать все возможные подмножества  $\{0, 1, 2, 3, \dots, N - 1\}$ .

## Задача 3. Sentinel: Descendants in Time

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	разумное

В этой задаче предлагается решить головоломку из Myst-подобной компьютерной игры “Sentinel: Descendants in Time”. Головоломка отлично показано в этом видео. Ближе к концу видео становится ясно, что игрок решает головоломку рекурсивным перебором. При этом используются осмысленные отсечения (и даже некоторые приоритеты).

Есть  $M$  лампочек и  $N$  пультов, на каждом пульте  $K$  переключателей. На каждой лампочке нужно добиться напряжения в ровно  $L$  единиц. Изначально все переключатели выключены, и напряжение на лампочках нулевое.

На каждом пульте требуется включить **ровно один** переключатель (все остальные должны быть выключены). При включении любого переключателя напряжение на некотором подмножестве лампочек вырастает на единицу. Это подмножество своё для каждого переключателя, и дано во входном файле.

Требуется определить для каждого пульта, какой переключатель нужно на нём включить, чтобы добиться в точности требуемого напряжения на всех лампочках.

### Формат входных данных

В первой строке дано четыре целых числа:  $N$  — количество пультов,  $K$  — количество переключателей на каждом пульте,  $M$  — количество лампочек и  $L$  — требуемое напряжение на каждой лампочке ( $1 \leq N, K \leq 10$ ,  $3N \leq M \leq 100$ ,  $0 \leq L \leq N$ ).

Далее описано влияние всех переключателей на все лампочки. Записано  $N \cdot K$  строк, по  $M$  символов в каждой. Если  $j$ -ый переключатель на  $i$ -ом пульте увеличивает напряжение на  $t$ -ой лампочке, то  $t$ -ый символ в строке под номером  $(i \cdot K + j)$  равен 'X' (большая буква «икс»), а если нет — то '.' (символ точки).

Могут быть добавлены дополнительные пустые строки в описании.

### Формат выходных данных

Если искомого решения не существует, нужно вывести слово NO, и больше ничего не выводить.

Если решение существует, то нужно вывести слово YES в первую строку, и решение в остальных  $N$  строках. В каждой  $i$ -ой из этих строк нужно вывести номер переключателя, который нужно включить на  $i$ -ом пульте (переключатели нумеруются с единицы).

Если решений несколько, можно вывести любое из них.

### Комментарий

В подобных задачах почти бессмысленно оценивать время работы асимптотически — сильно многое зависит от качества составления тестов. В данной задаче тесты сгенерированы случайным образом без попытки кого-то специально подловить.

### Пример

input.txt	output.txt
1 1 5 1 X.X.X	NO

## Пояснение к примеру

Пример в точности соответствует ситуации из игры, если пульты и переключатели пронумеровать слева направо, а лампочки на мосте пронумеровать сверху вниз — сначала левую половину, потом правую. Приведено единственное решение этой задачи, к которому и приходит игрок в видео.

## Задача 4. Судоку

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	разумное

Предлагается написать программу, которая будет автоматически решать судоку.

Судоку — это головоломка на клеточном поле размера 9 на 9. Изначально какие-то клетки поля пустые, а в каких-то клетках записаны цифры (в пределах от 1 до 9). Требуется в каждую пустую клетку вписать цифру (от 1 до 9), так чтобы было верно:

1. В каждой строке поля каждая цифра встречается ровно один раз.
2. В каждом столбце поля каждая цифра встречается ровно один раз.
3. В каждом блоке поля каждая цифра встречается ровно один раз.

Заметим, что поле можно разбить на 9 строк, на 9 столбцов, или на 9 квадратных блоков размера 3 на 3. В каждой из этих частей ровно 9 клеток, и в них должны быть записаны в точности все цифры от 1 до 9.

### Формат входных данных

В первой строке дано одно целое число  $N$  — количество полей судоку, которые нужно решить ( $1 \leq N \leq 100$ ). Далее приведено  $N$  полей, каждое поле — отдельная головоломка.

Каждое поле описывается в виде девяти строк, в каждой строке девять символов. Символ может быть либо цифрой от 1 до 9, либо точкой (точка обозначает пустую клетку). Поля **могут** быть отделены друг от друга пустыми строками.

Гарантируется, что на каждом поле существует решение. Скорее всего, оно также будет единственным.

### Формат выходных данных

Для каждого записанного во входных данных поля нужно вывести решение. Решения можно отделять друг от друга дополнительными пустыми строками.

Каждое решение — это девять строк по девять символов в каждой. Все символы должны быть цифрами от 1 до 9.

Если решений на головоломку несколько, можно вывести любое из них.

## Пример

input.txt	output.txt
3	483921657
..3.2.6..	967345821
9..3.5..1	251876493
..18.64..	548132976
..81.29..	729564138
7.....8	136798245
..67.82..	372689514
..26.95..	814253769
8..2.3..9	695417382
..5.1.3..	
	861357294
..1..7.9.	597482361
59..8...1	432619785
.3.....8.	916275843
.....58..	358964127
.5..6..2.	274138956
..41.....	789541632
.8.....3.	143826579
1...2..79	625793418
.2.7..4..	
	487312695
48.3.....	593684271
.....71	126597384
.2.....	735849162
7.5....6.	914265837
...2..8..	268731549
.....	851476923
..1.76...	379128456
3.....4..	642953718
....5....	

## Комментарий

Подсказка:

1. Клетки можно перебирать в разном порядке.
2. Попробуйте решить несколько простых sudoku самостоятельно.