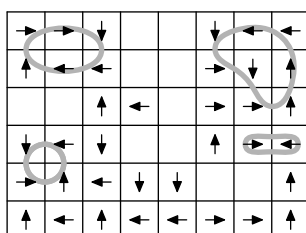


## Задача А. Стрелки

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

Костя нашел головоломку, в которую его дедушка играл в детстве — доску  $n \times m$  клеток, на некоторых из которых прикреплены вращающиеся стрелки (Костя предполагает, что раньше стрелки были на всех клетках). К сожалению, дедушка уже не помнит, что нужно было сделать в этой головоломке, поэтому Костя придумывает правила сам.

Он заметил следующую особенность. Направим каждую стрелку на одну из соседних клеток, на которой тоже есть стрелка (оказалось, что одиноко стоящих стрелок на доске нет). Теперь на доске появились циклы, идя по которым, мы всегда будем идти по направлению стрелок. Например, при такой расстановке:



есть четыре цикла.

Костя придумал сразу два задания: «поставить стрелки так, чтобы число циклов было минимально», и «поставить стрелки так, чтобы число циклов было максимально». Пока Костя ищет решение вручную, напишите программу, находящую требуемые расстановки.

### Формат входного файла

Первая строка входного файла содержит числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ). Следующие  $n$  строк по  $m$  символов содержат описание доски: символ «?» соответствует клетке со стрелкой, а символ «.» — пустой клетке.

### Формат выходного файла

Выведите в выходной файл сначала расстановку, в которой число циклов минимально, а затем расстановку, в которой число циклов максимально. Разделите расстановки одной пустой строкой.

Расстановку стрелок выводите, заменив в описании доски символы «?» на символы, соответствующие направлениям стрелок. Используйте символы «R», «L», «U» и «D» для обозначения направлений вправо, влево, вверх и вниз соответственно.

### Пример

arrows.in	arrows.out
3 4	RRRD
????	UUDL
????	.UL.
.??.	RLDD
	RLUU
	.RL.

## Задача В. Шоколадки

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

Рома играет сам с собой в очень интересную игру. Для нее нужна коробка конфет, в которой конфеты расположены прямоугольником  $n \times m$  штук. В игре участвуют конфеты из темного и белого шоколада. Сначала коробка заполняется конфетами произвольным образом. Далее Рома повторяет следующие операции. Он находит три конфеты одного цвета, лежащие рядом (в ряд, или в виде буквы «Г»), съедает их и заполняет освободившиеся места новыми конфетами произвольным образом. Если же он не находит трех конфет одного цвета, лежащих рядом, то игра заканчивается.

Посчитайте, сколько различных комбинаций может остаться на доске (то есть, в коробке) после окончания игры. Например, если  $n = 2$ ,  $m = 3$ , то может остаться восемь различных комбинаций:

W	W	B
B	B	W

B	W	B
W	B	W

B	W	W
W	B	B

B	W	B
B	W	B

B	B	W
W	W	B

W	B	W
B	W	B

W	B	B
B	W	W

W	B	W
W	B	W

(здесь символами «B» и «W» обозначены конфеты из темного и белого шоколада соответственно)

### Формат входного файла

Входной файл содержит два числа —  $n$  и  $m$ . ( $1 \leq n, m \leq 1000$ ).

### Формат выходного файла

Выведите в выходной файл одно число — ответ на вопрос задачи.

### Пример

choko.in	choko.out
2 3	8

## Задача С. Доставка грибов

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

Леша работает в службе доставки сушеных грибов. Некоторые клиенты жалуются, что грибы доставляются слишком долго. Леша хочет проверить, не является ли это следствием неэффективной системы доставки.

Система доставки устроена следующим образом. Существуют  $n$  распределительных пунктов, между некоторыми из которых есть односторонние каналы доставки. Грибы могут быть перевезены из пункта  $a_i$  в пункт  $b_i$  за время  $t_i$ . С основной базы грибы поступают на пункт номер 1 и далее доставляются до нужного пункта по каналам доставки. Система каналов довольно сложна, так что есть несколько путей, по которым можно доставить грибы до некоторых пунктов. При этом никто не следит за тем, чтобы выбранный путь был самым коротким.

Леша хочет узнать для каждого пункта, какое максимальное время до него могут идти грибы от первого пункта при условии, что на промежуточных пунктах грибы не задерживаются.

### Формат входного файла

В первой строке входного файла содержится числа  $n$  и  $m$  — количество распределительных пунктов и количество каналов доставки ( $2 \leq n \leq 100$ ,  $1 \leq m \leq 10000$ ).

Далее следуют  $m$  троек  $a_i, b_i, t_i$  ( $a_i \neq b_i$ ,  $1 \leq t_i \leq 100$ ).

### Формат выходного файла

Выведите в выходной файл  $n - 1$  число — максимальное время доставки грибов до второго, третьего, ...,  $n$ -го пункта. Если грибы до пункта могут идти сколь угодно долго, выведите вместо соответствующего числа  $-1$ .

### Пример

<code>delivery.in</code>	<code>delivery.out</code>
6 7	2 5 6 8 -1
1 2 2	
2 3 3	
1 3 1	
3 4 1	
4 5 2	
5 3 1	
3 6 1	

## Задача D. Убить Вову

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

Таня хочет убить Вову. Разумеется, виртуально. Сначала Вова выбирает одну из  $n$  тактик защиты, потом Таня, на зная, какую тактику выбрал Вова, атакует несколько раз, каждый раз выбирая одну из  $m$  тактик нападения. Для каждой тактики защиты известно, от каких тактик нападения она защищает. Помогите Тане узнать, за сколько атак она может гарантированно убить Вову и какие тактики нападения ей нужно для этого использовать.

### Формат входного файла

Первая строка входного файла содержит два числа —  $n$  и  $m$ . ( $1 \leq n, m \leq 20$ ). Далее идут  $n$  строк по  $n$  чисел,  $j$ -е число в  $i$ -ой строке равно 1, если тактика защиты  $i$  спасает от тактики нападения  $j$  и равно 0 в противном случае.

### Формат выходного файла

Выведите в выходной файл число  $k$  — минимальное число атак, за которое можно гарантированно убить Вову и далее  $k$  чисел — номера стратегий нападения, которые нужно использовать. Если Вову убить невозможно, выведите строку «Impossible».

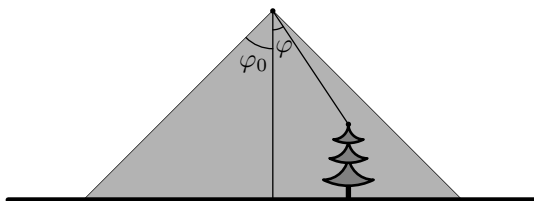
### Пример

kill.in	kill.out
4 5 1 0 0 0 0 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0	3 2 4 5
3 3 1 1 1 0 1 0 0 0 1	Impossible

## Задача Е. Громоотвод

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

В одном маленьком городке летом часто бывают грозы, поэтому жители решили на крыше одного из зданий установить громоотвод, защищающий все дома от ударов молнии. Громоотвод защищает точку в пространстве, если угол  $\varphi$  между направлением от вершины громоотвода до этой точки и отвесом не превышает некоторого угла  $\varphi_0$ .



Для большей устойчивости громоотвода жители хотят сделать его минимально возможной высоты (высота измеряется от крыши дома, на котором устанавливается громоотвод). Помогите жителям найти минимально возможную высоту громоотвода, защищающего от молнии все здания города.

### Формат входного файла

Первая строка входного файла содержит два целых числа: количество зданий  $n$  ( $1 \leq n \leq 10$ ) и величину угла  $\varphi_0$  в градусах ( $1 \leq \varphi_0 \leq 89$ ). Следующие  $n$  строк описывают здания. Все здания имеют форму параллелипипедов со сторонами, параллельными координатным осям (ось  $z$  направлена вертикально вверх). Таким образом, каждое здание описывается пятеркой  $x_1, y_1, x_2, y_2, h$ , где  $x_1, y_1$  и  $x_2, y_2$  — координаты противоположных углов здания,  $h$  — его высота.  $0 \leq x_1, y_1, x_2, y_2 \leq 1000$ ,  $1 \leq h \leq 1000$ ,  $x_1 < x_2$ ,  $y_1 < y_2$ , все числа целые.

### Формат выходного файла

Выведите в выходной файл минимально возможную высоту громоотвода, защищающего от молнии все здания города, с точностью не менее  $10^{-2}$ .

### Пример

<code>lighting.in</code>	<code>lighting.out</code>
2 30 0 0 3 3 1 3 3 4 4 1	4.898979485794484

## Задача F. Остатки

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

Саша устроился программистом в одну серьезную компанию. Однажды к нему зашел начальник и сказал: «Саша, что за бред выдает твоя программа? В одном месте у нее получается, что число  $3 \cdot n$  — нечетное, а в другом — что число  $5 \cdot n$  делится на 10. Ну и как такое может быть? У тебя что по математике в школе было?». По математике у Саши была твердая тройка, поэтому, чтобы больше не попадать в такие неприятные ситуации, он просит вас написать программу, проверяющую, может ли число  $a \cdot n$  давать остаток  $b$  по модулю  $c$ , и в то же время число  $d \cdot n$  давать остаток  $e$  по модулю  $f$ .

### Формат входного файла

Входной файл содержит шесть целых чисел:  $a, b, c, d, e$  и  $f$  ( $1 \leq a, c, d, f \leq 10^9$ ,  $0 \leq b \leq c$ ,  $0 \leq e \leq f$ ).

### Формат выходного файла

Выведите в выходной файл строку «YES», если это возможно и «NO» в противном случае.

### Пример

mod.in	mod.out
3 1 2 5 0 10	NO
3 1 2 5 5 10	YES

## Задача G. Распределение участков

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

В одном маленьком городке начинают работать  $n$  крупных компаний. Для начала они хотят поделить между собой  $n$  земельных участков. По расчетам экономистов, компания  $i$  может получить с участка  $j$  доход  $a_{i,j}$ . Каждая компания, разумеется, хочет получить наибольший доход.

Распределением участков занимается лично мэр города. Распределение происходит следующим образом. Каждая компания сообщает мэру, какой участок ей бы хотелось получить. После этого, если на участок претендует несколько компаний, то он отдается той из них, которая может получить с него наибольший доход (чем больше доход, тем больше налогов), а если таких несколько, то той, у которой наименьший номер (меньшие номера, разумеется, имеют знакомые мэра).

Зная, как происходит распределение, представители компаний решили собраться вместе и решить, кто на какой участок будет претендовать. Чтобы избежать жульничества, они хотят сделать так, чтобы доход компании не мог увеличиться, если она отклонится от выбранного плана, а все остальные будут ему следовать.

Оказалось, однако, что найти такой план не так-то просто. Помогите им сделать это.

### Формат входного файла

В первой строке входного файла содержится число  $n$  ( $1 \leq n \leq 300$ ). Далее идут  $n$  строк по  $n$  чисел.  $j$ -е число в  $i$ -ой строке — это  $a_{i,j}$ , доход  $i$ -ой компании с  $j$ -го участка ( $1 \leq a_{i,j} \leq 10^9$ ).

### Формат выходного файла

Выведите в выходной файл  $n$  чисел.  $i$ -е число — это участок, на который должна претендовать  $i$ -я компания. Если ответ не однозначен, выведите любой.

### Пример

<code>parcels.in</code>	<code>parcels.out</code>
3	1 3 2
5 4 4	
5 2 3	
2 1 2	

## Задача Н. Закупка носков

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

В одной военной части было принято революционное решение перейти от портянок к носкам. По такому случаю прапорщику Недалекому было поручено закупить  $n$  пар носков. Однако предложенная прапорщиком смета не удовлетворила начальство, и прапорщику было предложено очень-очень быстро переделать ее так, чтобы затраты были минимально возможными. Помогите бедному прапорщику составить такую смету.

Изучение рынка показало, что разные поставщики продают носки разными пачками и по разным ценам. Пачка, содержащая  $a_i$  пар носков продается за  $b_i$  рублей.

### Формат входного файла

В первой строке входного файла содержится числа  $n$  и  $m$  ( $1 \leq n \leq 10000$ ,  $1 \leq m \leq 100$ ). Далее идут  $m$  пар чисел  $a_i, b_i$  ( $1 \leq a_i \leq 10000$ ,  $1 \leq b_i \leq 10000$ ).

### Формат выходного файла

Выведите в выходной файл минимальную сумму денег, которую нужно потратить на покупку  $n$  пар носок.

### Пример

socks.in	socks.out
9 2 1 1 10 8	8