

## Технокубок 2018 - Отборочный Раунд 4

### А. Маша и медведи

ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт  
 ввод: стандартный ввод  
 вывод: стандартный вывод

У семьи из папы-медведя, мамы-медведя и сына-медведя есть три автомобиля. Известно, что в самый большой автомобиль папа-медведь влезает и он ему нравится. Аналогично, в средний автомобиль влезает мама-медведь, и он ей нравится. Сын-медведь влезает в самый маленький автомобиль, он ему нравится. Известно, что самый большой автомобиль по объему строго больше среднего, а тот — строго больше самого маленького.

Когда протестировать автомобили пришла Маша, она смогла влезть во все, но только самый маленький понравился ей.

Известно, что персонаж объема  $a$  влезает в автомобиль объема  $b$ , если и только если  $a \leq b$ ; если влезть в машину удалось, этот автомобиль персонажу нравится, если и только если  $2a \geq b$ .

Вам даны объемы Маши и медведей. Найдите какие-нибудь подходящие целочисленные неотрицательные объемы автомобилей.

#### Входные данные

Вам дано 4 целых числа  $V_1, V_2, V_3, V_m (1 \leq V_i \leq 100)$  — объёмы папы-медведя, мамы-медведя, сына-медведя и Маши соответственно. Гарантируется что  $V_1 > V_2 > V_3$

#### Выходные данные

Выведите 3 целых числа — размеры автомобилей папы-медведя, мамы-медведя и сына-медведя соответственно. Если существует несколько решений, выведите любое. Если решения не существует, в единственной строке выведите «-1» (без кавычек).

#### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
50 30 10 10	
<b>выходные данные</b>	
50 30 10	
<b>входные данные</b>	<a href="#">Скопировать</a>
100 50 10 21	
<b>выходные данные</b>	
-1	

#### Примечание

Легко видеть, что в первом тесте все условия на размеры автомобилей удовлетворены.

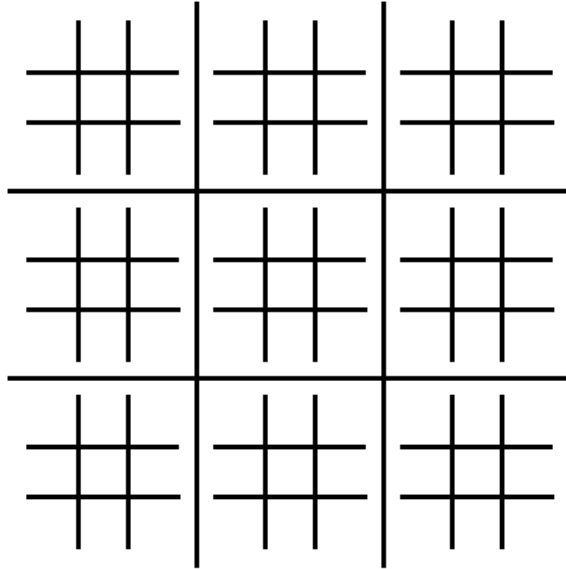
Во втором тестовом примере ответа не существует, так как в самый маленький автомобиль должна помещаться Маша (т.е. его минимальный размер — 21), но в тоже время сыну-медведю он должен нравиться, а значит его максимальный размер — 20.

## В. Крестики-нолики

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Два медведя играют в крестики-нолики по переписке. Медведям давно надоели обычные крестики-нолики на доске три на три, поэтому они играют в чуть более сложную версию этой игры. Опишем её правила.

Игра происходит на следующем поле:



Игроки ходят по очереди. На первом ходу первый игрок может поставить свою фишку в любую клетку любого маленького поля. Далее на ходы накладываются ограничения — если на предыдущем ходу оппонент поставил свою фишку в клетку с координатами  $(x_i, y_i)$  в некотором маленьком поле, то свой текущий ход игрок обязан совершить в маленькое поле с координатами  $(x_i, y_i)$ . Например, если первый игрок своим ходом поставит свою фишку в нижнюю левую клетку центрального поля, то второй игрок своим ходом поставит свою фишку в нижнее левое поле (обратите внимание на первый тестовый пример). Если в поле, в которое игрок должен ставить фишку, нет свободных клеток, то он может сделать ход в любую свободную клетку любого поля.

Вам дано текущее поле игры, а также координаты клетки, в которую был сделан последний ход. Вам необходимо определить, в какие клетки может поставить свою фишку текущий игрок.

Почтальоном в лесу работает заяц и он любит пакостить медведям. Поэтому он иногда немного перерисовывает игровое поле, но делает это так, чтобы медведи не заметили. А именно: в клетке, в которую был сделан последний ход, обязательно стоит фишка одного из игроков. Поэтому текущая позиция может быть недостижимой. Вам не нужно выяснять, достижима текущая позиция или нет, просто выведите возможные ходы следующего игрока.

### Входные данные

В первых 11 строках вам задана таблица 9 на 9, которая пробелами и переводами строк разбита на 9 маленьких полей, внутри каждого из которых описано соответствующее поле без пробелов и переводов строк. Символ «x» (ASCII-код 120) обозначает, что соответствующая клетка занята фишками первого игрока, «o» (ASCII-код 111) — фишками второго, символ «.» (ASCII-код 46) обозначает пустую клетку.

После таблицы в единственной строке записаны два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq 9$ ) — координаты клетки, в которую был сделан последний ход. Строки в таблице нумеруются сверху вниз, а столбцы — слева направо. Гарантируется, что в клетке, в которую был совершен последний ход находится либо символ «x», либо символ «o».

Также гарантируется, что существует хотя бы одна свободная клетка. Не гарантируется, что данная позиция достижима.

### Выходные данные

Выведите поле в аналогичном формате, где все клетки, в которые может поставить свою фишку очередной игрок заменены на «!» (ASCII-код 33). Все остальные клетки должны остаться без изменений.

### Примеры

входные данные	Скопировать
..... ..... .....  ..... .....	

... X.. ...

... ..

... ..

... ..

6 4

**ВЫХОДНЫЕ ДАННЫЕ**

... ..

... ..

... ..

... ..

... ..

... X.. ...

!!! ... ..

!!! ... ..

!!! ... ..

**ВХОДНЫЕ ДАННЫЕ**

Скопировать

XOO X.. X..

OOO ... ..

OOO ... ..

X.. X.. X..

... ..

... ..

X.. X.. X..

... ..

... ..

7 4

**ВЫХОДНЫЕ ДАННЫЕ**

XOO x!! x!!

OOO !!! !!!

OOO !!! !!!

x!! x!! x!!

!!! !!! !!!

!!! !!! !!!

x!! x!! x!!

!!! !!! !!!

!!! !!! !!!

**ВХОДНЫЕ ДАННЫЕ**

Скопировать

O.. ... ..

... ..

... ..

... ХХХ ...

... ХОХ ...

... ООО ...

... ..

... ..

... ..

5 5

**ВЫХОДНЫЕ ДАННЫЕ**

o!! !!! !!!

!!! !!! !!!

!!! !!! !!!

!!! ХХХ !!!

!!! ХОХ !!!

!!! ООО !!!

!!! !!! !!!

!!! !!! !!!

!!! !!! !!!

**Примечание**

Разберём первый тестовый пример.

В нём первый игрок сделал свой первый ход в нижнюю левую клетку центрального поля, поэтому второй игрок может ставить свою фишку только в нижнее левое поле.

Во втором тесте последний ход был сделан в верхнюю левую клетку нижнего центрального поля, однако все клетки в верхнем левом поле уже заняты, поэтому второй игрок может ставить свою фишку в любую свободную клетку любого поля.

В третьем тесте последний ход был сделан в центральную клетку центрального поля, однако все клетки там уже заняты, поэтому первый игрок может ставить фишку в любую свободную клетку любого поля. Обратите внимание, что такая позиция недостижима.

# С. Шокеры

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Валентин участвует в шоу, которое называется «Шокеры». Суть проста — жюри загадало некоторую букву, неизвестную Валентину. Он должен выступить с некоторой речью, но за каждое произнесённое слово, содержащее загаданную букву, его бьют током. Он может делать предположения, какая буква загадана, однако за каждую неправильную гипотезу его также будут бить током. Шоу заканчивается, когда Валентин угадывает букву правильно.

У Валентина не идеальная память, поэтому он мог угадать букву существенно позже, чем она однозначно определялась, то есть получить несколько лишних ударов током. Лишние удары током — это те удары, которые он получил после того момента, как загаданную букву можно было однозначно определить. Вам требуется найти количество таких лишних ударов.

## Входные данные

В первой строке находится единственное целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество действий Валентина.

В следующих  $n$  строках записаны его действия, по одному в строке. Каждое действие может принадлежать одному из трёх типов:

- Слово, за которое его не ударили током. Такое действие описывается строкой «. w» (без кавычек), где «.» — точка (ASCII-код 46), а w — сказанное Валентином слово.
- Слово, за которое его ударили током. Такое действие описывается строкой «! w» (без кавычек), где «!» — восклицательный знак (ASCII-код 33), а w — сказанное Валентином слово.
- Гипотеза о загаданной букве. Такое действие описывается строкой «? s» (без кавычек), где «?» — вопросительный знак (ASCII-код 63), а s — гипотеза, маленькая буква латинского алфавита.

Все сказанные слова состоят из строчных букв латинского алфавита, суммарная длина всех слов не превосходит  $10^5$ .

Гарантируется, что последним действием Валентин отгадывает загаданную букву. Также гарантируется, что ранее он её не отгадывал. Помимо того гарантируется, что в каждом слове, за которое его били током, такая буква присутствует, а в словах, за которые его не били — отсутствует.

## Выходные данные

Выведите единственное целое число — количество ударов тока, которых Валентин мог избежать, если бы назвал загаданную букву сразу после того, как она однозначно определилась.

## Примеры

входные данные	Скопировать
<pre>5 ! abc . ad . b ! cd ? c</pre>	
выходные данные	
<pre>1</pre>	

входные данные	Скопировать
<pre>8 ! hello ! codeforces ? c . o ? d ? h . l ? e</pre>	
выходные данные	
<pre>2</pre>	

входные данные	Скопировать
<pre>7 ! ababahalamaha ? a ? b ? a ? b ? a ? h</pre>	

## выходные данные

0

### Примечание

Разберём тестовые примеры.

В первом тесте после первого слова становится понятно, что загадана одна из букв  $a$ ,  $b$ ,  $c$ . Из второго слова мы узнаём, что загадана не буква  $a$ . Валентин произносит слово  $b$ , и не получает удар током. В этот момент мы понимаем, что загадана буква  $c$ , однако Валентин говорит слово  $cd$ , тем самым получает один лишний удар током.

Во втором тесте после первых двух ударов мы понимаем, что загадана либо буква  $e$ , либо буква  $o$ . Валентин начинает перебирать буквы из этих двух слов, и после второй его попытки угадать букву мы понимаем, что загадана буква  $e$ . Но Валентин делает ещё 3 попытки, прежде чем добирается до этой буквы.

В третьем тесте загаданная буква однозначно определяется только когда Валентин её угадал, поэтому он не получил лишних ударов током.

## D. Пересаживание студентов

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Студенты пришли в аудиторию писать контрольную и расселись некоторым образом. Преподаватель подумал: «Наверняка они сели так, чтобы им было удобней списывать друг у друга. Надо их пересадить так, чтобы ранее сидевшие рядом студенты рядом не сидели.»

Можно представить себе аудиторию как прямоугольное поле  $n$  на  $m$ , в каждой клетке сидит по студенту. Два студента сидят рядом друг с другом, если соответствующие им клетки смежны по стороне.

Пронумеруем студентов от 1 до  $n \cdot m$  по строчкам. Т.е. студент, изначально сидящий в клетке в  $i$ -й строчке и  $j$ -м столбце будет иметь номер  $(i - 1) \cdot m + j$ . Вам требуется найти таблицу, в которой будет  $n$  строчек и  $m$  столбцов, в которой каждое число от 1 до  $n \cdot m$  будет записано ровно один раз, а также соседние числа в изначальной таблице не должны быть соседними в получившейся, либо сказать, что такой не существует.

### Входные данные

В единственной строке находятся два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ;  $n \cdot m \leq 10^5$ ) — количество строк и столбцов в искомой таблице соответственно.

### Выходные данные

Если искомой таблицы не существует, в единственной строке выведите «NO» (без кавычек).

Если же она существует, то в первой строке выведите «YES» (без кавычек), а в следующих  $n$  строках выведите по  $m$  целых чисел — искомую таблицу.

### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
2 4	
<b>выходные данные</b>	
YES 5 4 7 2 3 6 1 8	

<b>входные данные</b>	<a href="#">Скопировать</a>
2 1	
<b>выходные данные</b>	
NO	

### Примечание

Разберём первый тест. Изначальная таблица выглядит так:

```
1 2 3 4
5 6 7 8
```

Легко видеть, что никакие две пары студентов не являются соседями одновременно в обеих таблицах.

Во втором тесте существует всего два варианта рассадить студентов, и в любом случае пара 1 2 будет сидеть рядом.

## Е. Вечеринка

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Арсений любит собирать людей на вечеринки. Однако к нему помимо друзей приходят также друзья друзей, друзья друзей друзей и так далее. Таким образом даже Арсений может не знать всех пришедших к нему гостей. Он решил исправить это недоразумение при помощи следующего процесса:

На каждом шаге выбирается один человек  $A$ , который в свою очередь попарно знакомит всех своих знакомых. После этого все знакомые  $A$  попарно становятся друзьями. Это продолжается до тех пор, пока все гости не будут попарно знакомы.

Арсению очень хочется провести поменьше времени за этим процессом, поэтому он хочет его завершить за наименьшее число шагов. Помогите ему в этом.

### Входные данные

В первой строке находятся два целых числа  $n$  и  $m$  ( $1 \leq n \leq 22$ ;  $0 \leq m \leq \frac{n \cdot (n-1)}{2}$ ) — количество человек на вечеринке, включая Арсения, и количество пар знакомых людей.

В следующих  $m$  строках записаны по два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ;  $u \neq v$ ). Каждая такая пара обозначает, что люди с номерами  $u$  и  $v$  знакомы. Гарантируется, что каждая пара знакомств описывается не более одного раза, а также гарантируется, что данный граф знакомств связан.

### Выходные данные

В первую строку запишите наименьшее количество ходов, за которое можно попарно познакомить всех людей. Во второй строке для каждого хода выведите номер гостя, который будет знакомить своих друзей на этом шаге. Если решений несколько, выведите любое.

### Примеры

входные данные	Скопировать
5 6 1 2 1 3 2 3 2 5 3 4 4 5	
выходные данные	
2 2 3	

входные данные	Скопировать
4 4 1 2 1 3 1 4 3 4	
выходные данные	
1 1	

### Примечание

В первом тестовом примере нет человека, который знал бы всех, поэтому для выполнения задачи требуется по меньшей мере два хода. После того, как второй перезнакомит всех своих друзей останутся незнакомыми только пары  $(4, 1)$  и  $(4, 2)$ . Познакомить их может либо 3, либо 5.

Во втором тестовом примере гость номер 1 знает всех, поэтому за один ход может всех попарно познакомить.



## Ф. Башня Мощи

ограничение по времени на тест: 4.5 секунд  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Жрецы культа Кетцалькоатля хотят построить величественную башню, символизирующую мощь их бога. Башня обычно строится из заряженных мощью камней. Она строится с помощью редкой магии, которая заставляет верхнюю часть башни левитировать над землёй, в то время как очередной камень пристраивается снизу башни. Если верхняя часть башни, построенная из  $k - 1$  камней, обладает мощью  $p$  и мы хотим добавить к ней камень, заряженный мощью  $w_k$ , то значение мощи новой башни будет равно  $\{w_k\}^p$ .

Камни добавляются, начиная с последнего и заканчивая нижним, поэтому для последовательности  $w_1, \dots, w_m$  значение мощи будет равно

$$w_1 \left( w_2 \left( w_3 \dots w_m \right) \right)$$

После того, как башня построена, её мощь может быть настолько колоссальной, что человеческий разум будет не в силах осознать её. Но жрецы всё ещё хотят иметь какую-то информацию о мощи башни, поэтому они хотят узнать значение приведённой мощи, которое равно истинному значению взятому по модулю  $m$ . У жрецов есть  $n$  камней пронумерованных от 1 до  $n$ . Они просят вас посчитать, каким значением приведённой мощи будет обладать башня, если её построить из камней с номерами  $l, l + 1, \dots, r$ .

### Входные данные

Первая строка ввода содержит два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^9$ ).

Вторая строка ввода содержит  $n$  целых чисел  $w_k$  ( $1 \leq w_k \leq 10^9$ ), обозначающих значение мощи камней, имеющихся у жрецов.

Третья строка содержит единственное число  $q$  ( $1 \leq q \leq 10^5$ ), равное числу вопросов, которые вам зададут жрецы.

$k^{th}$  из следующих  $q$  строк содержит два целых числа  $l_k$  и  $r_k$  ( $1 \leq l_k \leq r_k \leq n$ ).

### Выходные данные

Выведите  $q$  целых чисел.  $k$ -е из них должно быть равно величине приведённой мощи, которая будет у башни, построенной из камней с номерами  $l_k, l_k + 1, \dots, r_k$ .

### Пример

входные данные	Скопировать
6 1000000000 1 2 2 3 3 3 8 1 1 1 6 2 2 2 3 2 4 4 4 4 5 4 6	
выходные данные	
1 1 2 4 256 3 27 597484987	

### Примечание

$3^{27} = 7625597484987$