



## Prisoner Challenge

В тюрьме находятся 500 заключенных. Начальник тюрьмы решил предложить им шанс выбраться на свободу. Он положил в комнату две сумки с деньгами, сумку А и сумку Б. В каждой из сумок находится от 1 до  $N$  монет, включительно. В сумках находится **различное** число монет. Начальник предлагает им испытание. Задача заключенных состоит в том, чтобы определить сумку, в которой находится меньше монет.

В комнате помимо двух сумок с деньгами присутствует также маркерная доска. В любой момент времени на доске должно быть написано ровно одно целое число. Изначально на доске записано число 0.

После этого начальник приглашает заключенных в комнату по одному. Каждый из заключенных не знает, кто именно и сколько заключенных посетили комнату до него. Каждый раз, когда заключенный заходит в комнату, он видит число, написанное на доске. После того, как заключенный прочитал число на доске, он может выбрать сумку А или сумку Б. Затем заключенный **заглядывает** в выбранную сумку и узнает число монет в ней. После этого заключенный обязан выполнить одно из двух **действий**:

- Заменить число, записанное на доске, на неотрицательное целое число и покинуть комнату. Обратите внимание, что он может как заменить число на другое, так и оставить то же самое число. После того, как заключенный покинул комнату, испытание продолжается (за исключением случая, когда в комнате уже побывали все 500 заключенных).
- Сообщить начальнику, в какой сумке находится меньше монет. После этого испытание завершается.

Начальник никогда не приглашает в комнату заключенного, который в ней уже побывал.

Испытание считается пройденным, если один из заключенных успешно определит сумку, в которой находится меньше монет. Испытание считается непройденным, если один из них указал на неправильную сумку, или же все 500 заключенных побывали в комнате, но никто из них не попытался угадать, в какой сумке меньше монет.

Перед началом испытания заключенные собираются вместе, чтобы определить общую **стратегию** из трех шагов:

- Они выбирают неотрицательное число  $x$ , обозначающее максимальное число, которое любой из заключенных будет писать на доске.

- Они выбирают для каждого числа  $i$ , которое может быть написано на доске ( $0 \leq i \leq x$ ), в какую сумку должен заглянуть заключенный, если видит на доске число  $i$ .
- Они выбирают, что должен сделать заключенный после того, как заглянул в выбранную сумку. Более конкретно, для каждого из чисел  $i$ , которое может быть записано на доске ( $0 \leq i \leq x$ ), и для каждого числа монет  $j$  ( $1 \leq j \leq N$ ), увиденного в выбранной сумке, заключенные выбирают
  - либо какое число между 0 и  $x$  (включительно) заключенный запишет на доске,
  - либо какую сумку заключенный должен выбрать для финального ответа как ту, в которой меньше монет.

После прохождения испытания начальник выпустит всех заключенных через  $x$  дней.

Ваша задача состоит в том, чтобы помочь заключенным придумать стратегию, гарантирующую прохождение испытания (вне зависимости от числа монет в сумках А и Б). Оценка вашего решения зависит от числа  $x$  (смотрите секцию Subtasks).

## Implementation Details

Вы должны реализовать функцию

```
int[][] devise_strategy(int N)
```

- $N$ : максимальное число монет в каждой из сумок
- Эта функция должна возвращать массив  $s$ , состоящий из массивов целых чисел длины  $N + 1$ , описывающих вашу стратегию. Значение  $x$  определяется как длина массива  $s$ , минус один. Для каждого  $i$ , где  $0 \leq i \leq x$ , массив  $s[i]$  должен обозначать то действие, которое должен выполнить заключенный, если видит на доске число  $i$ :
  1. Значение  $s[i][0]$  должно быть равно 0, если заключенный должен заглянуть в сумку А, или 1, если заключенный должен заглянуть в сумку Б.
  2. Пусть  $j$  обозначает число монет, которое заключенный обнаружил в выбранной сумке. Заключенный должен сделать следующее:
    - Если значение  $s[i][j]$  равно  $-1$ , то заключенный должен выбрать сумку А как сумку для финального ответа, в которой находится меньшее число монет.
    - Если значение  $s[i][j]$  равно  $-2$ , то заключенный должен выбрать сумку Б как сумку для финального ответа, в которой находится меньшее число монет.
    - Если значение  $s[i][j]$  является неотрицательным, то заключенный должен записать его на доске. Обратите внимание, что  $s[i][j]$  не должно превосходить  $x$ .
- Эта функция будет вызвана ровно один раз.

## Example

Рассмотрим следующий вызов функции:

```
devise_strategy(3)
```

Пусть  $v$  обозначает то число, которое увидит заключенный при входе в комнату. Одна из допустимых стратегий такая:

- Если  $v = 0$  (в том числе в самом начале испытания), загляни в сумку А.
  - Если в ней 1 монета, то выбери сумку А для финального ответа.
  - Если в ней 3 монеты, то выбери сумку Б для финального ответа.
  - Если в ней 2 монеты, то запиши число 1 на доске (стерев 0).
- Если  $v = 1$ , загляни в сумку Б.
  - Если в ней 1 монета, то выбери сумку Б для финального ответа.
  - Если в ней 3 монеты, то выбери сумку А для финального ответа.
  - Если в ней 2 монеты, то запиши число 0 на доске (стерев 1). Обратите внимание, что такая ситуация не может произойти никогда, так как в этом случае обе сумке содержат по 2 монеты, что невозможно по условию.

Для того, чтобы выдать описанную стратегию, функция должна вернуть  $[[0, -1, 1, -2], [1, -2, 0, -1]]$ . Длина возвращаемого массива равна 2, таким образом, значение  $x$  для данного массива определяется как  $2 - 1 = 1$ .

## Constraints

- $2 \leq N \leq 5000$

## Subtasks

1. (5 баллов)  $N \leq 500$ , значение  $x$  не должно превосходить 500.
2. (5 баллов)  $N \leq 500$ , значение  $x$  не должно превосходить 70.
3. (90 баллов) Значение  $x$  не должно превосходить 60.

Если в каком-либо тесте результат функции `devise_strategy` не задает корректную стратегию, вы получите за эту подзадачу 0 баллов.

В позадаче 3 вы можете получить частичный балл. Пусть  $m$  обозначает максимально значение  $x$  среди всех выданных стратегий на тестах данной позадачи. Ваш балл за эту подзадачу будет определяться согласно следующей таблице:

Ограничение	Баллы
$40 \leq m \leq 60$	20
$26 \leq m \leq 39$	$25 + 1.5 \times (40 - m)$
$m = 25$	50
$m = 24$	55
$m = 23$	62
$m = 22$	70
$m = 21$	80
$m \leq 20$	90

## Sample Grader

Доступный вам грайдер считывает данные в следующем формате:

- строка 1:  $N$
- строка  $2 + k$  ( $0 \leq k$ ):  $A[k] B[k]$
- последняя строка:  $-1$

Каждая из строк, за исключением первой и последней, обозначает один тестовый пример. В дальнейшем мы будем называть тестовый пример, описанный в строке  $2 + k$ , как пример  $k$ . В примере  $k$  сумка А содержит  $A[k]$  монет, а сумка Б содержит  $B[k]$  монет.

Грайдер вызывает функцию `devise_strategy(N)`. Значение  $x$  определяется как длина результирующего массива, минус один. После этого если грайдер обнаруживает, что результат функции `devise_strategy` не соответствует требованиям из условия, то он печатает одно из следующих сообщений об ошибке и завершает работу:

- `s` is an empty array:  $s$  является пустым массивом (и не является корректной стратегией).
- `s[i]` contains incorrect length: существует такой индекс  $i$  ( $0 \leq i \leq x$ ), для которого длина массива  $s[i]$  не равна  $N + 1$ .
- First element of `s[i]` is non-binary: существует такой индекс  $i$  ( $0 \leq i \leq x$ ), для которого  $s[i][0]$  не равно 0 или 1.
- `s[i][j]` contains incorrect value: существуют такие индексы  $i, j$  ( $0 \leq i \leq x, 1 \leq j \leq N$ ), что  $s[i][j]$  не находится в диапазоне от  $-2$  до  $x$ .

В противном случае, грайдер выводит два набора выходных данных.

Во-первых, грайдер выводит результат выполнения вашей стратегии на тестовых примерах.

- строка  $1 + k$  ( $0 \leq k$ ): результат выполнения стратегии для сценария  $k$ . В случае, если следуя стратегии результатом является выбор сумки А, грейдер выводит символ А. В случае, если следуя стратегии результатом является выбор сумки Б, грейдер выводит символ В. В случае, если следуя стратегии ни один из заключенных не выберет сумку для финального ответа, грейдер выводит символ Х.

Во-вторых, грейдер создает файл `log.txt` в текущей директории, в который он выводит данные в следующем формате:

- строка  $1 + k$  ( $0 \leq k$ ):  $w[k][0] w[k][1] \dots$

Последовательность в строке  $1 + k$  соответствует примеру  $k$  и описывает последовательность чисел, записанных на доске во время выполнения стратегии для примера  $k$ . Более конкретно,  $w[k][l]$  обозначает число, записанное  $(l + 1)$ -ым по счету заключенным (в том порядке, в котором они заходили в комнату).