

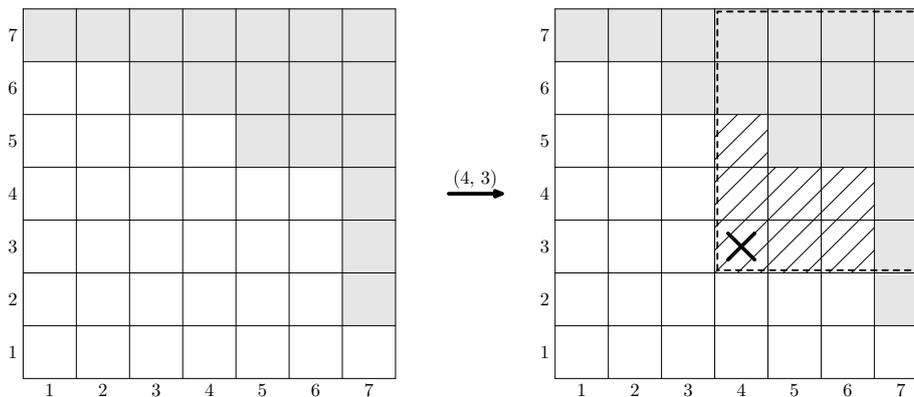
## Задача 3. Игра с тайным смыслом

Ограничение по времени: 3 секунды  
Ограничение по памяти: 1024 мегабайта

Это интерактивная задача с двойным запуском.

Один шпион добыл важное сообщение, и ему необходимо передать его в центр. Так как сообщение важное, было решено, что необходимо скрыть даже сам факт передачи какой-либо информации, и поэтому передача сообщения будет замаскирована под стрим набирающей популярность игры «Щёлк».

Правила игры «Щёлк» очень простые. На квадратном поле размера  $n \times n$  изначально ни одна клетка не закрашена. Два игрока ходят по очереди. За один ход игрок может выбрать любую ещё незакрашенную клетку и закрасить её и все незакрашенные клетки прямоугольника, левым нижним углом которого является выбранная клетка, а правым верхним — правый верхний угол поля (см. рисунок). Игрок, закрасивший левый нижний угол поля, проигрывает.



Шпион будет играть на сайте, предоставляющем возможность сыграть с ботом. Бот играет следующим образом: каждый раз он случайно равновероятно выбирает один из ходов, которые закрасивают не более  $k$  новых клеток, кроме хода в левый нижний угол. Если же ход в левый нижний угол — единственный оставшийся, то бот делает его и проигрывает игру.

Шпион может сыграть столько игр, сколько необходимо, однако для целей конспирации лучше, чтобы их было как можно меньше. Кроме того, чтобы все выглядело как можно менее подозрительно, лучше как можно больше игр выиграть. С помощью записи ходов, сделанных во всех сыгранных играх, шпион и планирует передать важное сообщение.

Вам необходимо написать программу, которая будет запущена два раза. При первом запуске программа будет исполнять роль шпиона: зная секретное сообщение, она сыграет с ботом несколько игр в «Щёлк», а при втором запуске, получив только списки ходов этих игр, восстановит секретное сообщение.

### Протокол взаимодействия

Ваше решение будет запущено два раза.

В первой строке входных данных записано одно число 1 или 2 — номер запуска.

При обоих запусках решения во второй строке записаны три числа:  $n$  — размер стороны квадратного поля ( $n = 32$  во всех тестах жюри),  $k$  — максимальное количество клеток, которые может закрасить бот ( $k = 8$  во всех тестах жюри),  $m$  — длина сообщения в битах ( $m = 100\,000$  во всех тестах, кроме примера).

При первом запуске в третьей строке находится битовое представление передаваемого сообщения — строка длины  $m$ , состоящая из символов «0» и «1». После его считывания решение должно начать взаимодействие с программой жюри, реализующей стратегию бота. Взаимодействие состоит в обмене ходами. Ход задается парой чисел — номерами столбца и строки выбранной клетки соответственно. Столбцы нумеруются слева направо от 1 до  $n$ , а строки — снизу вверх от 1 до  $n$ . Выбранная клетка не должна быть закрашенной, иначе решение будет прервано и получит вердикт «WA». После того, как один из игроков сделал проигрышный ход «1 1», игра заканчивается, и тут же

начинается новая. В первой игре первый ход делает ваша программа, а далее первый ход чередуется, то есть во второй игре первой будет ходить программа жюри, в третьей снова ваша программа и так далее.

Когда решение считает, что записи имеющихся игр достаточно для передачи сообщения, вместо своего первого хода в игре необходимо вывести единственное число 0, и завершить работу решения. Даже если в этой игре первый ход делает бот, можно вывести 0 вместо своего первого хода и завершить работу решения. В таком случае первый ход бота не будет записан в лог, доступный решению на втором запуске. В середине игры прервать ее нельзя.

Если решение во время первого запуска сыграет с соблюдением протокола некоторое число игр, уложившись в ограничения по времени и памяти, оно будет запущено второй раз. Иначе второго запуска не будет, и вердикт при первом запуске («WA», «RE», «TL», «ML» или «IL») будет объявлен результатом проверки на этом тесте.

При втором запуске в третьей строке будет находиться количество игр, которые решение сыграло на первом запуске. После этого на вход будет подан список ходов сделанных во всех играх, как решением, так и ботом, в порядке, в котором эти ходы происходили. При втором запуске решение должно вывести одну строку длины  $m$  из «0» и «1» — секретное сообщение, которое было доступно первому запуску.

При втором запуске вам необходимо сначала считать все данные из стандартного потока ввода, и только затем вывести ответ.

Если второй запуск состоялся, и вердикт второго запуска «RE», «TL», «ML» или «IL», — этот результат будет объявлен результатом проверки на этом тесте. Иначе выведенный программой ответ будет проверен. Если он совпадает с переданным программе при первом запуске секретным сообщением, результат проверки на тесте будет «OK», иначе — «WA». В обоих запусках после каждого вывода вашей программы выводите перевод строки и делайте сброс потока вывода.

Если вы используете «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`Console.WriteLine`» в C#, «`writeln`» в Паскале, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае. Для сброса потока вывода можно использовать «`fflush(stdout)`» в C и C++, «`flush(output)`» в Паскале, «`System.out.flush()`» в Java, «`sys.stdout.flush()`» в Python, «`Console.Out.Flush()`» в C#.

Для удобства локального тестирования, вы можете использовать вариант программы жюри с визуализацией игры, доступной в рабочем каталоге. Программу можно запустить командой «`java -jar visualizer.jar`» из консоли, или двойным кликом по файлу в файловом менеджере. Вы можете также скачать визуализатор в тестирующей системе на вкладке «Файлы».

Ваше решение может взаимодействовать с программой жюри, используя для ввода-вывода файлы, которые будут указаны после нажатия на кнопку «Начать игру». У решения может не получиться открыть файлы до того, как запущена программа жюри, поэтому следует нажимать кнопку «Начать игру» и только после этого запускать решение. После завершения взаимодействия будет предложено сохранить лог игры в формате входных данных для второго запуска и секретное сообщение, чтобы вы могли проверить корректность его восстановления вашей программой. Кроме проверки взаимодействия выданная вам программа позволяет посмотреть как проходила игра между вашим решением и программой жюри. Для этого нужно перед запуском поставить настройку «Визуализировать игру». Также можно сыграть с программой жюри самостоятельно, выбрав соответствующую опцию в интерфейсе.

## Система оценивания

Ваше решение запускается на 10 тестах, за каждый из которых можно получить до 10 баллов.

Секретное сообщение в каждом тесте зафиксировано и имеет длину  $m = 100\,000$ . В каждом тесте сообщение было получено генератором случайных чисел, каждый символ независимо с равной вероятностью был выбран равным «0» или «1».

Ходы, которые делает решение жюри в каждом тесте, случайны, в соответствии с описанием в условии, но при повторных запусках, если программа участника будет делать те же самые ходы, то программа жюри также будет делать те же самые ходы в этом тесте.

Если решение не смогло правильно восстановить сообщение на втором запуске или нарушило протокол взаимодействия на первом запуске, за тест будет выставлено 0 баллов.

Иначе балл за тест зависит от количества сыгранных игр при первом запуске, а также от количества игр, которые были выиграны программой-решением.

Пусть  $W = 1$ , если проиграно не более одной игры, и  $W = 0$  в противном случае. Пусть  $G$  — количество сыгранных игр,  $B = \frac{m}{G}$  — среднее количество бит, переданных за одну игру. Числа  $b_i$  определяют значение  $B$ , необходимое для получения  $i$  баллов, и представлены в таблице ниже.

$b_0$	$b_1$	$b_2$	$b_3$	$b_4$	$b_5$	$b_6$	$b_7$	$b_8$	$b_9$
0	1	100	300	450	600	700	750	800	825

В случае если  $B \geq b_9$ , решение получает  $W + 9$  баллов за тест. Иначе, пусть  $s$  таково, что  $b_s \leq B < b_{s+1}$ . В таком случае, решение получит  $W + s + \frac{B - b_s}{b_{s+1} - b_s}$  баллов за тест. Баллы за отдельные тесты будут просуммированы, и уже итоговая сумма будет округлена до ближайшего целого числа.

Вашим результатом по этой задаче будет результат наилучшей попытки в целом.

### Пример

Приведенный ниже пример предназначен только для того, чтобы ознакомиться с форматом ввода и вывода. В частности, не предполагается, что в указанном примере шпион действительно передал какую-либо информацию с помощью приведенной там последовательности ходов.

Ваше решение не будет запущено на тесте из примера при отправке в тестирующую систему.

Первый запуск.

стандартный ввод	стандартный вывод
1	
32 8 16	
1111000011110000	
	3 2
30 1	
	1 2
28 1	
	2 1
1 1	
31 31	
	0

В примере ввод и вывод отформатированы пустыми строками, чтобы было видно, какие запросы соответствуют каким ответам программы жюри. В реальном взаимодействии необходимо переводить строку после каждого запроса, но выводить пустые строки не надо.

Второй запуск.

стандартный ввод	стандартный вывод
2	1111000011110000
32 8 16	
1	
3 2	
30 1	
1 2	
28 1	
2 1	
1 1	