

Правила игры «Лавина»

Введение

«Лавина» — это игра для двух игроков. Каждый игрок играет на своём поле, представляющем собой прямоугольную сетку шириной 8 клеток и высотой 16 клеток. Изначально поле пустое. На каждом ходу на поле участника выпадает фигура из двух шаров, каждый из которых занимает ровно одну клетку и имеет один из следующих цветов: синий, красный, зелёный, фиолетовый или жёлтый. Участник может повернуть эту фигуру и выбрать место, куда она упадёт.

Если шары одного цвета образуют связную область из четырёх или более клеток, то все эти шары исчезают (две клетки считаются связными, если имеют общую сторону). При этом если над ними лежали другие шары, они упадут вниз. Если после этого вновь образуется связная область из четырёх или более шаров одного цвета, то они снова исчезают и процесс продолжается. Если образуется несколько таких областей, все они исчезают одновременно.

Если своим ходом игрок уничтожил некоторое количество шаров, то на поле противника перед следующим его ходом выпадут чёрные шары. Чёрные шары отличаются от цветных тем, что они не уничтожаются при образовании связной области из них. Зато при уничтожении области из цветных шаров соседние с ней чёрные шары также уничтожаются. Подробнее про чёрные шары будет сказано в следующем разделе.

Если после хода игрока некоторые шары оказываются выше верхней границы поля, то этот игрок считается проигравшим. Если оба игрока проигрывают одновременно, то победителем считается тот, на чьём поле за время игры появилось меньше чёрных шаров. В случае равенства этого показателя побеждает тот, поле которого содержит меньше чёрных шаров на момент окончания игры. Если и их одинаковое количество, партия заканчивается вничью.

Чёрные шары

Процесс уничтожения шаров одного цвета будем называть лавиной. Под длиной лавины будем понимать общее количество шагов, каждым из которых уничтожаются одноцветные области и обваливаются шары, лежащие выше.

Тогда общее количество выпавших на поле противника чёрных шаров равно сумме бонусов за каждый шаг лавины и бонуса за длину лавины.

Бонус за каждый шаг лавины равен количеству уничтоженных шаров, минус количество уничтоженных областей, минус 2.

Бонус за длину лавины равен:

- 0, если длина лавины 1;
- 3, если длина лавины 2;
- 6, если длина лавины 3;
- 12, если длина лавины 4;
- 24, если длина лавины 5;
- 50, если длина лавины 6;
- 100, если длина лавины 7 и более.

Таким образом, при уничтожении одной области из четырёх шаров на поле противника упадёт один чёрный шар. А если, например, на первом шаге лавины было уничтожено две области по четыре шара, а на втором — ещё одна область из пяти шаров, то на поле противника выпадет $(2 \cdot 4 - 2 - 2) + (5 - 1 - 2) + 3 = 9$ шаров.

Место падения каждого чёрного шара выбирается случайно и равновероятно среди всех столбиков, не заполненных на этот момент доверху.

Также время от времени обоим игрокам будет выпадать одинаковое количество чёрных шаров в качестве подарка от жюри. Жюри обещает вам, что это будет происходить не чаще чем раз в сто ходов.

Описание протокола взаимодействия с тестирующей системой

Участник турнира должен предоставить своё решение жюри в виде одного файла с исходным текстом программы, написанным на одном из поддерживаемых языков программирования: C++, Java, C# или Pascal.

Программа участника должна общаться с тестирующей системой через стандартные потоки ввода/вывода (stdin/stdout).

В начале игры программе приходит запрос, состоящий из двух строк. В первой строке записано слово **Start**, во второй строке — целое число x_0 . Ответ на него должен состоять из одной строки, содержащей имя игрока.

Затем следует последовательность ходов, каждый из которых имеет следующий вид: сначала идёт строка с единственным словом **Move**, затем описание поля игрока, затем пустая строка, затем описание поля противника.

Описание каждого из полей состоит из 16 строк по 8 символов. Символами могут быть «.», если соответствующая клетка поля пуста, «*», если в этой клетке находится чёрный шар, или цифра от 1 до 5, если в этой клетке находится цветной шар (цифра — номер его цвета).

Если нумеровать ходы, начиная с единицы, то на i -м ходу игроку выпадает фигура, первый шар которой имеет цвет $x_{2i-2} \bmod 5 + 1$, а второй шар — цвет $x_{2i-1} \bmod 5 + 1$. Последовательность x_n задаётся рекуррентной формулой $x_{i+1} = (x_i \cdot 17235617 + 4311271) \bmod 87465851$, где x_0 — число, данное программе участника в начале игры.

В качестве ответа программа участника должна вернуть **ровно одну строку, завершающуюся переводом строки**. В строке должны быть записаны целые числа s и r через пробел. s — позиция левого шара из двух (целое число от 0 до 7). r — положение шаров друг относительно друга:

- 0, если второй шар должен упасть справа от первого;
- 1, если второй шар должен упасть снизу от первого;
- 2, если второй шар должен упасть слева от первого;
- 3, если второй шар должен упасть сверху от первого.

В конце игры программе приходит запрос **Quit**, после которого она должна немедленно завершить свою работу. Если программа не завершит работу после запроса **Quit** или завершит работу до этого запроса, то ей будет зачтено поражение.

После вывода каждого ответа обязателен вызов процедуры **flush**:

C++	Java	C#	Pascal
<code>fflush(stdout), cout.flush()</code>	<code>System.out.flush()</code>	<code>Console.Out.Flush()</code>	<code>Flush(Output)</code>

Ограничения на программу участника

Программе засчитывается техническое поражение, если:

- имя игрока длиннее 25 символов или содержит символы с кодами меньше 32;
- программа делает некорректный ход;

- программа не укладывается в ограничение по времени — **10 секунд на каждые 100 ходов**;
- программа не укладывается в ограничение по памяти — **64 мегабайта**;
- программа завершает работу, не получив команду `Quit`;
- после получения команды `Quit` программа не завершает работу.

Участники могут быть дисквалифицированы, если:

- программа запускает другие программы или создаёт новые процессы;
- программа использует сетевые средства или работает с файловой системой;
- программа работает с внешними устройствами (принтером, звуковой картой и т. д.);
- программа работает с любыми GUI-элементами (окнами, диалогами и т. д.);
- программа выполняет любые другие действия, которые могут нарушить работу проверяющей системы соревнования.

Регламент проведения игрового тура

Во время игрового тура участникам будет предоставлена система для локального проведения и визуализации игр (арена), а также несложные примеры программ-игроков на всех доступных языках программирования (сэмпл-боты). Участники могут улучшить сэмпл-бот или написать полностью новое решение, используя сэмпл-бот лишь как пример взаимодействия с проверяющей системой.

Решение получит вердикт `Accepted`, если оно успешно скомпилировалось.

В ходе игрового тура (через 2 часа после начала) состоится промежуточный турнир. Он будет проводиться по швейцарской системе в 2–5 туров. Результаты промежуточного турнира будут доступны всем командам и не будут учитываться при подведении итогов тура. В случае непредвиденных обстоятельств жюри имеет право отказаться от публикации этих результатов.

По окончании игрового тура будет проведён турнир среди отправленных решений. В нём будет участвовать только **последнее** решение команды, получившее вердикт `Accepted`.

Этапы турнира:

1. Турнир по швейцарской системе в 9–12 туров для определения 16 лучших решений. В случае победы программа получает 1 очко, в случае ничьей — 0.5 очка, в случае поражения — 0 очков. При игре по швейцарской системе в каждом туре встречаются соперники с наиболее близким количеством набранных очков. При этом двое соперников играют друг с другом не более одного раза.
2. Плей-офф по олимпийской системе (система игры навывлет), в результате которого будут распределены первые 8 мест. Если программы играют вничью, то победитель определяется жребием.