

## Правила игры «Футбол»

### Введение

«Футбол» — это пошаговая игра для двух игроков. Игра ведётся на поле, представляющем собой прямоугольную сетку шириной 20 клеток и высотой 14 клеток. Узлы сетки пронумерованы сверху вниз и слева направо числами от 0 до 14 и от 0 до 20 соответственно. Изначально в центральном узле сетки с координатами (7, 10) находится мяч. Также на левой и правой сторонах поля стоят ворота первого и второго игроков соответственно. *Ворота* представляют собой отрезки с координатами (5, 0) – (9, 0) и (5, 20) – (9, 20). Узлы, расположенные строго внутри ворот, называются *финишными*. Все остальные узлы на границе поля называются *граничными*. В течение игры мяч перемещается по узлам сетки. *Смежными* узлами для узла  $(x, y)$  являются узлы, у которых  $x$ - и  $y$ -координаты отличаются от  $x$ - и  $y$ -координат исходного узла не более чем на 1.

Ход представляет собой последовательность узлов, по которым перемещается мяч, такую что:

- первый узел — тот, в котором стоит мяч на момент начала хода;
- любые соседние узлы в последовательности являются смежными;
- для всех узлов последовательности, кроме последнего, верно, что мяч уже успел побывать в этом узле в течение игры или узел является граничным;
- мяч ещё не находился в последнем узле последовательности и этот узел не является граничным;
- ломаная, по которой будет перемещаться мяч в результате хода, не имеет общих рёбер с ломаными предыдущих ходов и не проходит по одному ребру два или более раз;
- ни один из отрезков этой ломаной не лежит на границе поля.

Игроки по очереди совершают ходы. Первый ход делает игрок, ворота которого расположены слева. Игрок проигрывает, если не может сделать ход либо в его ворота залетает мяч.

### 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 одной строки, в которой написано слово `Name`. Ответ на него должен состоять из одной строки, содержащей имя игрока.

Далее программе поступают запросы на совершение хода. Запрос представляется одной строкой и имеет следующий вид.

- Если программа должна совершить первый ход в партии, то строка содержит одно слово `Start`
- В противном случае строка состоит из нескольких чисел, разделённых пробелами, описывающих ход соперника. Первое число  $n$  — количество узлов в ходе. Последующие  $2n$  чисел — это координаты узлов хода.

В качестве ответа программа участника должна вернуть **ровно одну строку, завершающуюся переводом строки**. В строке должен быть записан ответный ход в точно таком же формате, как описано выше.

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 или больше 127;
- программа делает некорректный ход;
- программа не укладывается в ограничение по времени — **10 секунд на каждые 100 ходов**;
- программа не укладывается в ограничение по памяти — **256 мегабайт**;
- 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 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. В плей-офф пара команд играет два раза, первыми в играх ходят разные игроки. Если каждый игрок в паре одержит по одной победе, победителем пары считается тот, который одержал свою победу за меньшее число ходов. Если победы будут одержаны за одинаковое число ходов, победитель определяется жребием.