

## Правила игрового тура «Прорыв»

### Описание игры

«Прорыв» — это игра для двух игроков, которая проходит на стандартной шахматной доске. Стартовая позиция игры изображена на рисунке справа. Один из игроков играет белыми, другой — чёрными. Игроки ходят по очереди, первый ход делают белые. В игре существует единственный тип фигур — пешка. За один ход необходимо сходить одной из своих пешек.

Цель игры — дойти одной из своих пешек до противоположного края доски или срубить все пешки противника.

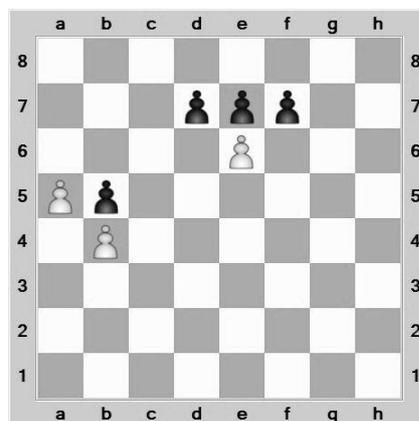
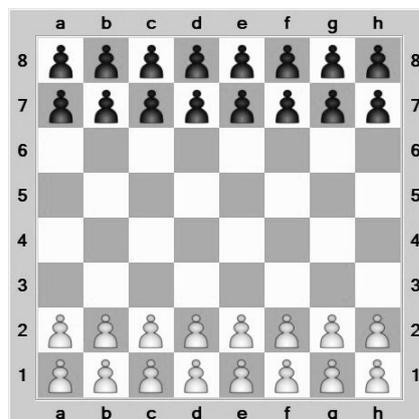
Ходы бывают двух типов — простые и взятия. При простом ходе пешка перемещается на одну клетку вперёд прямо или по диагонали, если эта клетка пуста. При взятии пешка рубит пешку противника, расположенную на одну клетку вперёд по диагонали. При этом срубленная пешка снимается с доски, а срубившая занимает её место.

На рисунке справа пешка **a5** может сделать два простых хода: на **a6** и **b6**. Пешка **b4** может сделать простой ход только на поле **c5**, так как **a5** занято своей пешкой, а **b5** — пешкой противника. У пешки **e6** нет простых ходов, зато есть два взятия: пешки **d7** и пешки **f7**.

Нотация ходов следующая: сначала записывается начальная клетка, затем — конечная.

Список возможных ходов для позиции, изображённой на рисунке, для белых: **a5a6**, **a5b6**, **b4c5**, **e6d7**, **e6f7**; для чёрных: **b5a4**, **b5c4**, **d7c6**, **d7d6**, **d7e6**, **e7d6**, **e7f6**, **f7e6**, **f7f6**, **f7g6**.

В «Прорыве» не бывает ничьих, так как все пешки движутся только вперёд и всегда можно сделать хотя бы один ход.



### Протокол взаимодействия с проверяющей системой

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

Программа участника должна общаться с проверяющей системой через стандартные потоки ввода/вывода (stdin/stdout) в режиме запрос–ответ. Основная процедура программы должна в цикле считывать со стандартного ввода по одной строке, содержащей запрос, и выводить на стандартный вывод по одной строке, содержащей ответ на этот запрос. **Строка-запрос и строка-ответ обязаны завершаться переводом строки.**

В начале игры программе приходит запрос **Name**. Ответ на него должен состоять из одной строки, содержащей имя игрока. Затем, если программа играет белыми, она получит запрос **Start**, если же чёрными, — ей на вход придёт первый ход белых. Программе необходимо выдать свой ход в качестве ответа. Ответный ход противника программа получит в следующем запросе. Гарантируется, что все ходы, которые программа получает от проверяющей системы, являются корректными. В конце игры программа получит запрос **Quit**, после которого она должна немедленно завершить свою работу. Если программа не завершит работу после того, как получит запрос **Quit**, или завершит работу до этого запроса, ей будет зачтено поражение.

## Ограничения на программу участника

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

- для C++: `fflush(stdout), cout.flush();`
- для Java: `System.out.flush();`
- для Pascal: `Flush(Output);`
- для C#: `Console.Out.Flush();`

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: **3 секунды** на ход;
- программа не укладывается в ограничение по памяти: **64 мегабайта**;
- программа обдумывает ход во время хода соперника;
- 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 реализации программ-игроков на всех доступных языках программирования (сэмпл-боты). Арена и сэмпл-боты будут выложены в папке **X:\Breakthrough\**. Участники могут улучшить сэмпл-бот, либо написать полностью новое решение, используя сэмпл-бот лишь как пример взаимодействия с проверяющей системой.

При каждой посылке решения будет автоматически проводиться две игры с сэмпл-ботом — белыми и чёрными. Программа получит вердикт **Accepted**, если она успешно скомпилировалась. Логи игр будут выложены на диск **R:**, их можно будет воспроизвести в арене.

В ходе игрового тура будут проведены два промежуточных турнира — через 2 часа и через 3 часа после начала. Промежуточные турниры будут проведены по швейцарской системе в 2–5 туров. Результаты промежуточных турниров будут доступны всем командам. Эти результаты не будут учитываться при подведении итогов. В случае непредвиденных обстоятельств Жюри имеет право отказаться от публикации этих результатов.

По окончании игрового тура будет проведён турнир среди отправленных решений. В турнире 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. При игре по швейцарской системе в каждом туре встречаются соперники с наиболее близким суммарным количеством побед. При этом двое соперников сыграют друг с другом не более одного раза.
2. Плей-офф по олимпийской системе (система игры навывлет), в результате которого будут распределены первые 8 мест. Программы будут играть по две партии — белыми и чёрными. Если счёт по партиям будет 1:1, то выиграет та программа, которая победила за меньшее число ходов. Если программы победили за равное число ходов, то победитель определяется жребием.

### Пример игры

Move	Player	Input	Output
-	White	Name	ZZZ Bot
-	Black	Name	QXX Bot
1	White	Start	a2a3
1	Black	a2a3	a7a6
2	White	a7a6	a3a4
2	Black	a3a4	a6a5
3	White	a6a5	a4b5
3	Black	a4b5	a5a4
4	White	a5a4	b5a6
4	Black	b5a6	a4a3
5	White	a4a3	a6a7
5	Black	a6a7	a3a2
6	White	a3a2	a7b8
-	White	Quit	
-	Black	Quit	