

Точки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В данной задаче вам предлагается написать программу, играющую в описанную далее игру.

Игра происходит на бесконечном клетчатом поле, на котором введена прямоугольная система координат. Играют k игроков (в этой задаче $k = 2$, но не рекомендуется вбивать это как константу, т.к. потом будет проведён турнир с большим количеством игроков). Изначально у каждого игрока есть клетка, покрашенная в его цвет.

Игра длится T дней, в каждый из дней игроки совершают ходы в порядке возрастания своих номеров. В свой ход игрок должен покрасить одну из **свободных** клеток своим цветом. Клетка считается **свободной**, если в ней не располагается стена, её не закрывает туман, и она не покрашена никаким цветом. Если после очередного хода игрока какие-то его клетки огораживают *замкнутую область*, эта область (включая клетки противника, но не включая стены) красится в его цвет. *Замкнутой областью* называется связное множество клеток, от клеток которого нельзя выйти за пределы видимой области поля, перемещаясь только в соседние по стороне клетки, не принадлежащие текущему игроку (для лучшего понимания см. примеры и пояснения).

Изначально игроку доступны для просмотра и покраски клетки, у которых обе координаты не превышают N по модулю, на остальных клетках лежит туман. Каждые P дней доступный квадрат расширяется на одну клетку влево, вправо, вверх и вниз.

Побеждает игрок, под чьим контролем по истечении T дней будет наибольшая территория.

ТЛ на информатиксе указан в шапке условия. При запуске турнира изначально у вас есть 1 секунда на все ходы. Каждый ход лимит по времени увеличивается на 30 миллисекунд. Жюри оставляет за собой право менять эти значения в случае крайней необходимости.

Формат входных данных

В первой строке содержатся 5 чисел: T ($1 \leq T \leq 10^4$) — длительность игры в днях, N ($1 \leq N \leq A$) — изначальное ограничение на модуль координат видимых клеток, P ($1 \leq P \leq T + 1$) — периодичность расширения поля, k — количество игроков, A ($1 \leq A \leq 1000$) — **длина стороны** поля после всех увеличений поля.

Во второй строке находится число num ($1 \leq num \leq k$) — номер игрока.

В следующих $2 \cdot N + 1$ строках содержится по $2 \cdot N + 1$ числу — описание видимого поля. На таблице введена система координат таким образом, информация о клетке $(0; 0)$ находится на пересечении строки $N + 1$ и столбца $N + 1$, о клетке $(1; 1)$ — строки N и столбца $N + 2$. Значение клетки равно 0, если она пуста, или -1 , если там находится стена.

Протокол взаимодействия

На каждом своём ходу игрок должен вывести в одну строку 2 числа: x и y — координаты очередной покрашенной клетки. Клетка должна быть свободна и доступна. Перед этим ему необходимо получить от интерактора информацию о ходах других игроков и расширении поля.

Ход другого игрока выводится в формате $n \ x \ y$, где n — номер сходявшего, x и y — координаты хода. Обратите внимание, что ходы окончивших игру будут пропускаться. Гарантируется, что в течение одного дня оставшиеся игроки ходят в порядке увеличения номера.

Информация о расширении поля начинается и заканчивается строками $-1 \ -1 \ -1$. Между ними вводится несколько (возможно, ноль) строк вида $0 \ x \ y$, где x и y — координаты новой стены, расположенной в открывшейся области. Гарантируется, что расширение происходит в начале каждого из дней под номерами $P, 2P, 3P, \dots$, первый день имеет номер 1.

Каждый сеанс ввода интерактора будет заканчиваться строкой $-2 \ -2 \ -2$, уведомляющей о необходимости сделать ход.

Когда игрок закончит игру, интерактор выведет ему строку $-3 \ -3 \ -3$. После её получения программа участника должна немедленно завершиться.

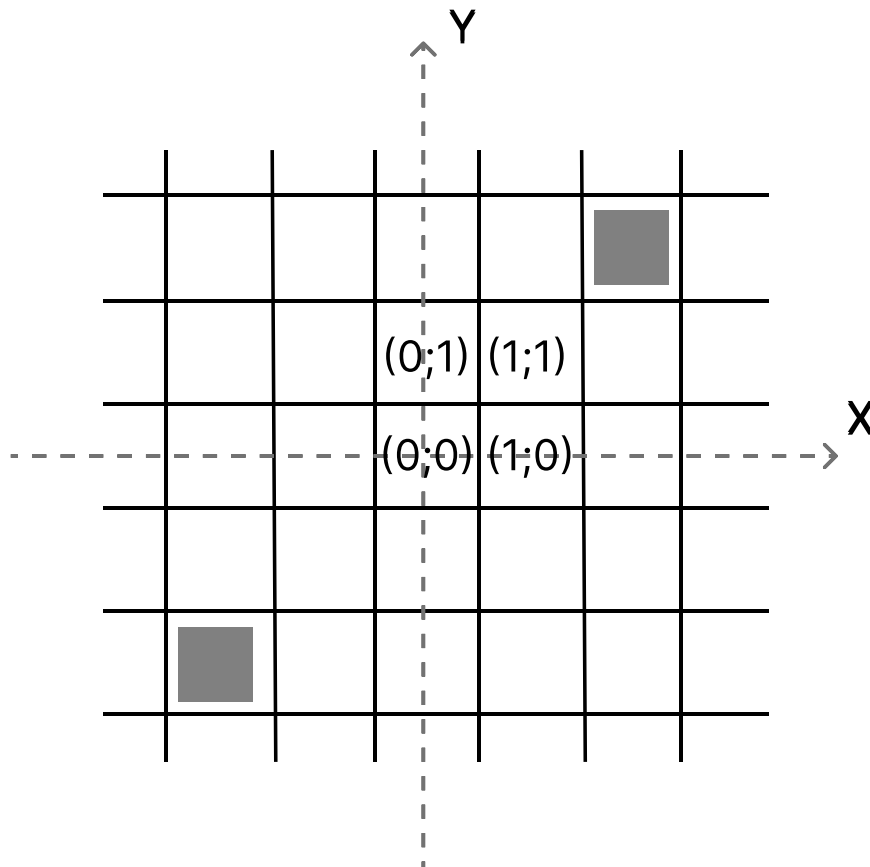
Любое некорректное действие игрока будет расцениваться как проигрыш. Любое некорректное действие игрока будет расцениваться как проигрыш. Если вы хотите сдать, в ответ на запрос интерактора выведите два числа, задающих заведомо некорректное действие и ждите $-3 - 3 - 3$ от интерактора.

Обратите внимание, что программа участника должна выводить свой ход только после получения строки $-2 - 2 - 2$ от интерактора (в частности, интерактор сразу после вывода исходного поля выводит игроку под номером `num` ходы всех игроков с номерами, меньшими `num`, затем выводит $-2 - 2 - 2$, и только после этого участник должен выводить свой ход).

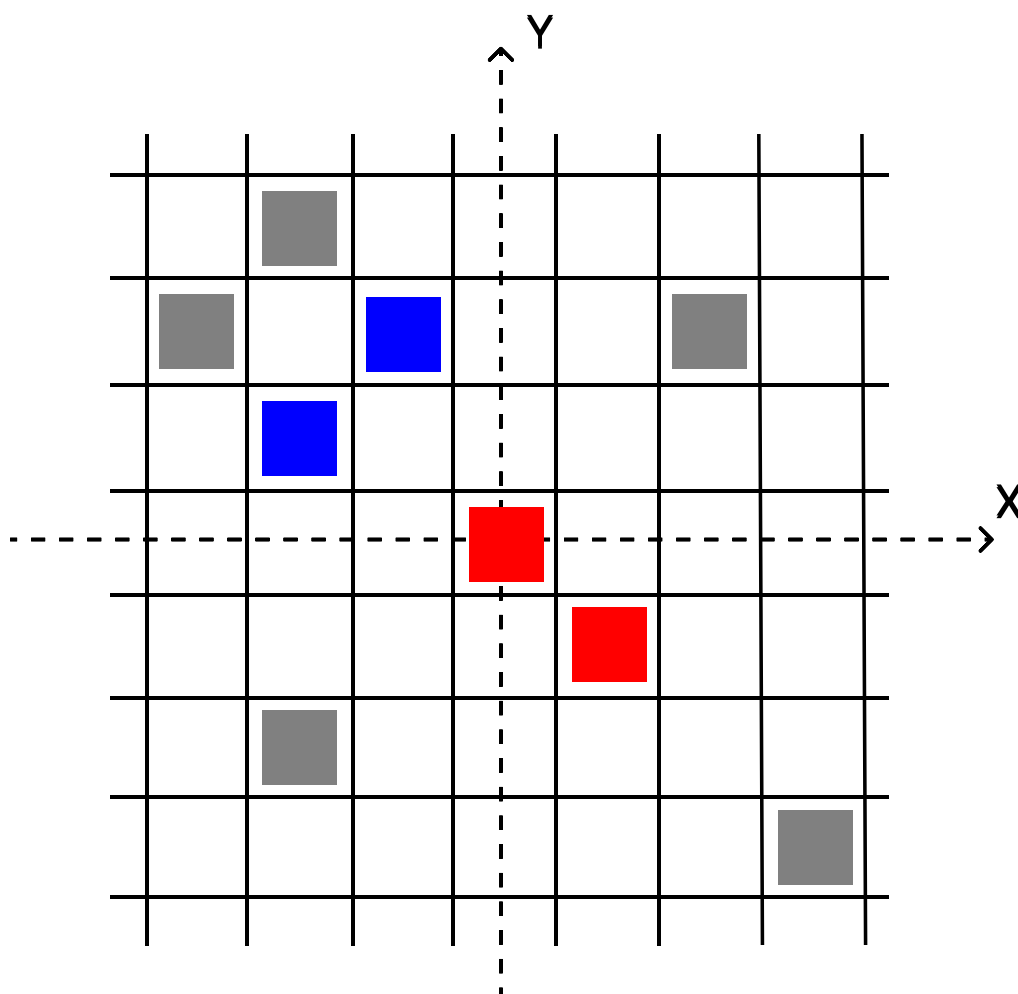
Не забывайте переводить строку и очищать буфер после каждого вывода (`std::cout « std::endl` в C++).

Замечание

Приведём пример взаимодействия. Пусть игра длится 7 дней, играют 2 игрока на поле с изначальным размером 5×5 (т.е. $N = 2$), которое расширяется каждые 3 дня. Исходное поле:

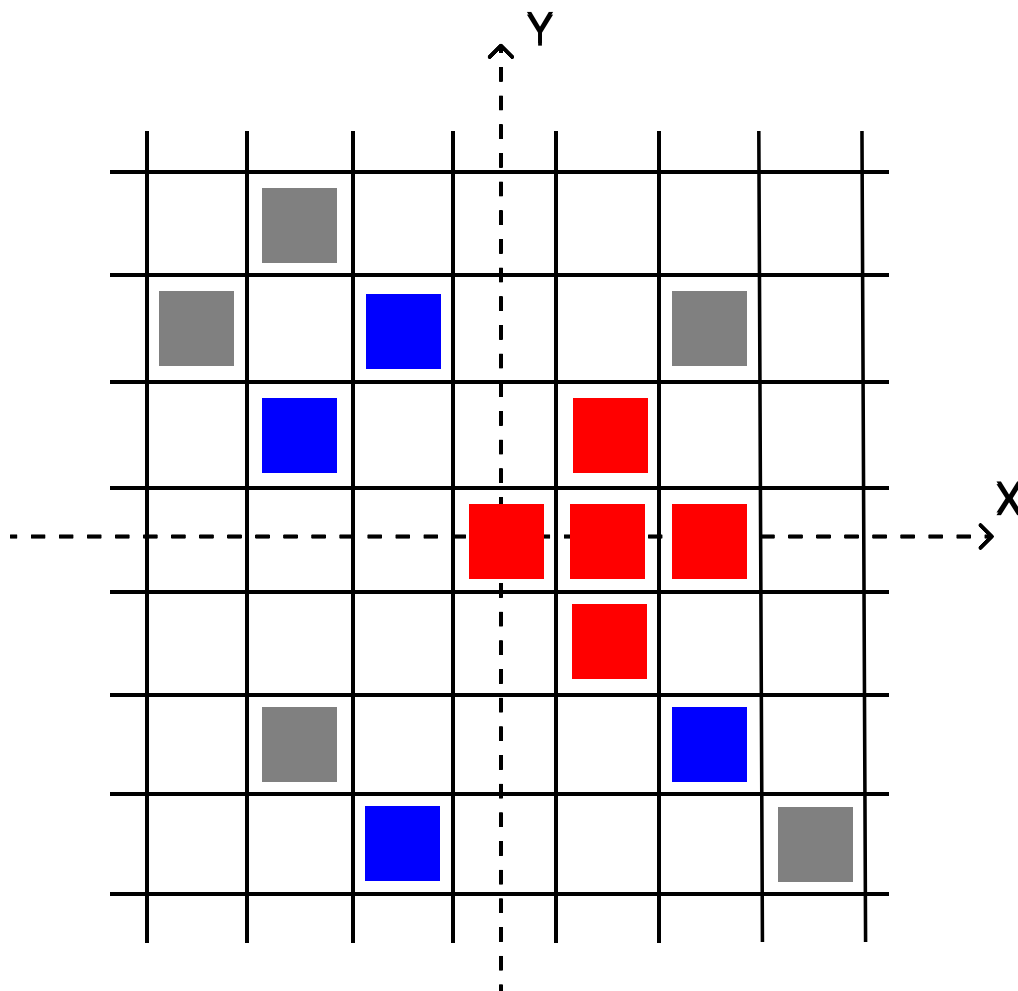


Игроки по очереди делают ходы. Первый — $(1; -1)$, второй — $(-2; 1)$, первый — $(0; 0)$, второй — $(-1; 2)$. После второго хода второго игрока (т.е. после конца второго и начала третьего дня) поле расширится и будет выглядеть так:



Обратите внимание, что в расширенной области тоже могут быть стены.

Далее первый игрок делает ход $(1; 1)$, второй — $(-1; -3)$, первый — $(2; 0)$. На этом момента клетка $(1; 0)$ оказывается в замкнутой области и переходит первому. Обратите внимание, что ещё ход назад область не замыкалась. Теперь, когда второй игрок сделает ход $(2; -2)$, после окончания четвертого дня поле будет выглядеть так:



Приведём начало протокола взаимодействия интерактора с первым игроком:

```

7 2 3 2 9 // информация о поле (начало)
1
0 0 0 0 -1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
-1 0 0 0 0 // информация о поле (конец)
-2 -2 -2 // ожидание хода
→ 1 -1 // ваш ход
2 -2 1 // ход второго игрока
-2 -2 -2
→ 0 0
2 -1 2
-1 -1 -1 // информация о расширении поля (начало)
0 -2 3
0 3 -3
0 -3 2
-1 -1 -1 // информация о расширении поля (конец)
-2 -2 -2
→ 1 1
...

```