

Задача 7. «Ёжик в тумане»

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

*Над рекой поднялся туман, и грустная белая Лошадь утонула в нем по грудь.
 – Вот, – сказал Ёжик. – Ничего не видно. И даже лапы не видно.
 – Лошадь! – позвал он. Но лошадь ничего не сказала. "Где же Лошадь?" – подумал Ёжик.
 (С.Г. Козлов)*

Ёжик спустился в туман и оказался в прямоугольной долине размером N на M метров, по которой бродит Лошадь. Ёжик хочет ее найти. Будем считать, что в каждый момент времени и Ёжик, и Лошадь находятся в одной из $N \times M$ клеток. Туман настолько густой, что Лошадь не видно, даже если она находится в той же самой клетке, что и Ёжик. К счастью Ёжик обладает очень острым слухом и может понять, в каком направлении сместилась Лошадь. Он также может позвать Лошадь, и, если она находится в одной клетке с Ёжиком, то Лошадь его услышит и обязательно ответит.

В каждый момент времени Ёжик может сместиться в соседнюю клетку по горизонтали, вертикали или диагонали. Потом он отчетливо слышит, куда сместилась Лошадь относительно своего старого местоположения. Лошадь за единицу времени смещается на одну клетку только по горизонтали или вертикали (влево, вверх, вправо или вниз). При этом Лошадь не выходит за границы долины, поэтому и Ёжик не должен этого делать.

Требуется написать программу, которая поможет Ёжику, знающему свое начальное положение и следящему за передвижениями Лошади, как можно быстрее ее найти.

Это интерактивная задача. В процессе тестирования программа-решение будет взаимодействовать с использованием стандартных потоков ввода/вывода с программой, моделирующей поведение лошади.

Протокол взаимодействия программ

Сначала программа-решение должна прочитать из стандартного потока ввода натуральные числа N и M , записанные в первой строке, а из второй строки координаты начального местоположения Ёжика – два натуральных числа: x_0 – номер столбца, y_0 – номер строки ($1 \leq x_0 \leq M$, $1 \leq y_0 \leq N$). Числа в каждой строке разделены пробелом.



Затем программа-решение начинает взаимодействие с программой, моделирующей поведение лошади, в соответствии со следующим протоколом:

1. Программа выводит в стандартный поток вывода одну строку, описывающую ход Ёжика, которая содержит три числа: его перемещение в виде указания смещения по горизонтали dx ($dx = -1, 0$ или 1) и по вертикали dy ($dy = -1, 0$ или 1), а также число 1 , если Ёжик зовет Лошадь в клетке, в которую он при этом попадет, или 0 – если не зовет. Вывод должен завершаться переводом строки и сбросом буфера потока вывода. Для этого используйте

- `flush(output)` в паскале или Delphi;
- `fflush(stdout)` или `cout.flush()` в C/C++;
- `Console.out.flush()` в Visual Basic.

2. После этого программа должна считать из стандартного потока ввода ответ программы, сообщаемой о действии Лошади. Ответ состоит из трех чисел, расположенных в одной строке через пробел. Первое число ответа может быть равно 0 или 1 , где

- 0 означает, что Ёжик не пытался позвать Лошадь либо позвал, но Лошади в его клетке нет. В этом случае следующие два числа обозначают очередное смещение Лошади по горизонтали dx ($dx = -1, 0$ или 1) и по вертикали dy ($dy = -1, 0$ или 1), при этом хотя бы одно из значений dx или dy равно нулю;

	1	2	3	4						M
1										
2						$\begin{matrix} d \\ x \end{matrix}$	-1	0	1	
3						-1		\uparrow		
	$\begin{matrix} d \\ x \end{matrix}$	-1	0	1	0	\leftarrow		\rightarrow		
	-1	\leftarrow	\uparrow	\nearrow	1		\downarrow			
	0	\leftarrow		\rightarrow						
	1	\leftarrow	\downarrow	\searrow						
N										

- 1 означает, что Ёжик позвал Лошадь, и она действительно оказалась в той же клетке, что и он. В этом случае другие два числа равны 0, и программа-решение должна закончить свою работу.

Программа-решение не должна делать более 10 000 ходов.

Пример взаимодействия

стандартный ввод	стандартный вывод
2 3	
1 2	0 0 1
0 1 0	1 -1 0
0 0 0	1 0 1
1 0 0	

Комментарий

Ёжик находился в клетке (1, 2). Сначала он попробовал позвать Лошадь в той же клетке (вывод: 0 0 1), но Лошади там не оказалось, и она сместилась вправо (ввод: 0 1 0). Ёжик сместился по диагонали, но Лошадь звать не стал (вывод: 1 -1 0), а Лошадь осталась на месте (ввод: 0 0 0). Ёжик сместился вправо и позвал Лошадь (вывод: 1 0 1). Лошадь оказалась в той же клетке и отозвалась (ввод: 1 0 0). Значит, изначально Лошадь находилась в клетке (2, 1), а встретились они в клетке (3, 1). Ёжик при этом сделал три хода и дважды запросил местоположение Лошади.

Подзадачи и система оценки

В данной задаче две подзадачи. Каждый тест в обеих подзадачах оценивается отдельно. Оценка за тест вычисляется по формуле $\min\{10, \text{round}(10 \times (J/S)^2)\}$, где 10 – оценка в баллах за тест, S – количество ходов, которое потребовалось программе-решению, чтобы обнаружить Лошадь, J – количество ходов, которое требуется заданному эталонному решению при том же начальном положении Ёжика. Округление ведется по правилам математики.

Подзадача 1 (оценивается из 40 баллов)

$$2 \leq N, M \leq 10.$$

Подзадача 2 (оценивается из 60 баллов)

$2 \leq N, M \leq 30$. В этой подзадаче количество запросов о том, есть ли Лошадь в текущей клетке, не должно превышать $N \times M$.

Вспомогательная программа

Для тестирования своего решения вы можете использовать вспомогательную программу «runpair», которая находится у вас в каталоге «с:\work\runpair».

Она позволяет запустить две программы и перенаправить стандартный поток вывода первой программы на стандартный поток ввода второй программы и наоборот.

Для тестирования с ее помощью программы-решения вам придется помимо программы-решения написать программу, моделирующую поведение лошади. Тогда можно запустить одновременно программу, моделирующую поведение лошади, и программу-решение, с помощью команды

runpair «исполнимый файл лошади» «исполнимый файл ёжика»

и на экране будет отображен их диалог.