

# Яндекс. Тренировки по алгоритмам 2.0, занятие 6 (А)

## А. Горячо-Холодно

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Это интерактивная задача.

Егор и Пётр играют в игру «Горячо-Холодно» на двумерной плоскости. В начале игры Егор загадывает точку с неотрицательными целыми координатами, не превышающими  $10^9$ . После чего Пётр пытается угадать эту точку: на  $i$ -м ходу он выбирает некоторую точку с целыми координатами  $(x_i, y_i)$  и сообщает её Егору. Если эта точка ближе к загаданной, чем предыдущая  $(x_{i-1}, y_{i-1})$ , Егор отвечает “1”. В противном случае (в том числе и на самый первый вопрос Петра в начале игры) Егор отвечает “0”.

Когда Пётр считает, что у него достаточно информации, он останавливает игру и сообщает ответ (это действие ходом уже не считается). Если ответ правильный и Пётр сделал не более 500 ходов, Пётр считается победителем. В противном случае выиграл Егор. Пётр просит Вас написать программу, которая гарантированно будет выигрывать у Егора.

Гарантируется, что Егор играет честно и не будет менять координаты загаданной точки в процессе игры.

### Протокол взаимодействия

Когда игрок делает ход, он должен вывести два целых числа от 0 до  $10^9$ , разделённых пробелом — координаты очередной точки, о которой он спрашивает. Если игрок хочет остановить игру и сообщить ответ, он должен вывести символ 'A', а после него через пробел — два целых числа:  $x$  и  $y$ -координаты загаданной точки, после чего завершить выполнение программы.

После каждого вывода вы обязаны выводить один символ перевода строки, делать команду `flush`, очищая поток вывода, и считывать ответ. Если ваша программа получит на стандартный вход EOF, она обязана завершить выполнение. В противном случае возможно получение ошибки `TimeLimitExceeded`.

На каждый ход программа жюри выводит “1” в случае, когда названная программой-игроком точка ближе к загаданной, чем предыдущая, и “0” в противном случае (то есть когда названная точка не ближе предыдущей или предыдущей точки названо не было, то есть если ход первый).

Гарантируется, что координаты загаданной точки целые, неотрицательные и не превосходят  $10^9$ .

### Пример взаимодействия

Система Участник

```
1 1
0
0 0
0
20 20
1
20 20
0
17 239
1
17 240
0
A 17 239
```

## Примечания

В приведённом примере загадана точка  $(x = 17, y = 239)$  и приведён следующий сценарий.

1. Участник называет точку  $(1, 1)$ , программа жюри отвечает 0, так как это первый ход.
2. Участник называет точку  $(0, 0)$ , которая дальше от точки  $(x = 17, y = 239)$ , чем точка  $(1, 1)$ , так что программа жюри снова отвечает 0.
3. Следующая попытка участника —  $(20, 20)$ , и сейчас ответ — 1, так как точка ближе к загаданной.
4. Участник снова называет  $(20, 20)$ , ответ — 0, так как в случае одинакового расстояния точка не ближе.
5. Участник называет точку  $(17, 239)$ ... которая и была загадана, но просто получает ответ 1 — если правильная точка названа не как ответ, то игра продолжается.
6. Программа жюри отвечает 0 на точку  $(17, 240)$ .
7. Участник решает рискнуть и называет точку  $(17, 239)$  в качестве ответа. Заметим, что он не имел нужной информации. Ему просто повезло. Вам на такое везение рассчитывать не стоит.

Функция `flush` для очистки потока вывода в некоторых языках:

В C (или C++ с использованием `stdin`) используйте `fflush(stdout);`

В C++ используйте `cout.flush();`

В Java используйте `System.out.flush();`

В Pascal используйте `Flush(Output).`

Язык Python 3.12.1

Набрать здесь

Отправить файл

```
1 lf_x = 0 # левая граница бин поиска x
2 rg_x = 10**9 # правая граница бин поиска x
3 while lf_x != rg_x: # пока бин поиск не сошелся
4     print(lf_x, 0) # проверяем левую границу
5     _ = int(input()) # ответ системы
6     print(rg_x, 0) # проверяем правую границу
7     answer = int(input()) # ответ системы
8     # если правая граница ближе, то переносим левую в mid + 1, иначе переносим правую границу в mid
9     if answer == 1:
10         lf_x = (rg_x + lf_x) // 2 + 1
11     else:
12         rg_x = (rg_x + lf_x) // 2
13
14 lf_y = 0 # левая граница бин поиска y
15 rg_y = 1000000000 # правая граница бин поиска y
16 while lf_y != rg_y: # пока бин поиск не сошелся
17     print(lf_x, lf_y) # проверяем левую границу
18     _ = int(input()) # ответ системы
19     print(lf_x, rg_y) # проверяем правую границу
20     answer = int(input()) # ответ системы
21     if answer == 1: # если правая граница ближе, то переносим левую в mid + 1, иначе переносим правую границу в mid
22         lf_y = (rg_y + lf_y) // 2 + 1
23     else:
24         rg_y = (rg_y + lf_y) // 2
25
26 # ответ
27
```

Отправить

Следующая