

CHANGE: Chef and His Garden

Легенда, описанная далее, переформулирована и упрощена переводчиком, чтобы читатель мог лучше понять условие задачи. Оригинальную легенду вы можете прочитать на странице задачи в контексте.

Условие:

Шеф работает в компании Go Green Initiative. Ранее он посадил n деревьев в ряд напротив ресторана. Сегодня высота i -того дерева равна соответственно h_i футов. Дерево растет со скоростью m_i футов в день.

Шеф знает, что деревья выглядят красиво, если они формируют *zig-zag* последовательность. Деревья формируют *zig-zag* последовательность, если высота деревьев сначала возрастает или убывает, а затем чередуется между убыванием и возрастанием. Формально говоря, деревья находятся в *zig-zag* последовательности, если выполняется одно из двух условий:

- $h_1 < h_2 > h_3 < h_4$ и так далее
- $h_1 > h_2 < h_3 > h_4$ и так далее

Шеф хочет найти все интервалы времени, когда деревья сформируют *zig-zag* последовательность.

Формат ввода:

Первая строка содержит единственное целое число T – количество тестовых случаев.

Далее следует описание тестов в следующем формате:

Первая строка каждого теста содержит единственное целое число n – количество деревьев.

Каждая из следующих строк каждого теста разделенные пробелами целые числа h_i и m_i – начальную высоту дерева и скорость роста i -того дерева соответственно.

Формат вывода:

Для каждого тестового случая выведите на отдельной строке единственное целое число Q – количество последовательных отрезков времени, когда деревья образуют *zig-zag* последовательность.

Затем на следующих Q строках выведите минимальный и максимальный момент времени интервала, когда последовательность деревьев формируют *zig-zag* последовательность.

Тесты сгенерированы так, что общий вывод не превысит **2МБ**.

Ограничения & оценивание:

- $1 \leq T \leq 10^5$
- $1 \leq n \leq 10$
- Подзадача 1 (23 балла): $0 \leq h_i, m_i \leq 10$
- Подзадача 2 (77 баллов): $0 \leq h_i, m_i \leq 10^9$
- $1 \leq$ сумма всех n во всех тестовых случаях $\leq 5 \times 10^5$

Примеры тестов:

Входные данные:

```
3
3
0 1
2 2
0 3
2
2 1
1 2
3
1 1
2 2
3 3
```

Выходные данные:

```
1
0 1
2
0 0
2 Inf
0
```

Пояснения:

Тест 1: последовательность **0 2 0** уже формирует *zig-zag* последовательность, на второй секунде она уже становится **2 6 6** и больше не станет *zig-zag* последовательностью.