

# Горки, греки, гэги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как известно, в Древней Греции были положены начала философии, благодаря которой, в частности, удалось формализовать понятие науки. Одной из главных наук того времени считалась математика: Пифагор, Евклид, Архимед — вы точно слышали эти имена! А уж о том, что, в некотором смысле вследствие развития математики этими знаменитыми философами, вы читаете условие этой задачи, говорить не приходится...

Тем не менее, даже таким выдающимся учёным иногда приходится спускаться с небес на землю и просить архитекторов строить здания для своих школ.

Архитекторы — люди инженерного склада, и потому здесь и там у них возникают задачи, связанные с математикой. Вот и на этот раз перед ними встала следующая проблема.

У них есть  $n$  строительных блоков, стоящих на одной прямой. Блок с номером  $i$  имеет высоту  $h_i$ ; длину и ширину всех блоков будем считать единичными. Назовем *горкой* непрерывную подпоследовательность с  $l$ -го по  $r$ -й блок такую, что найдется  $m : l < m < r$  и  $h_l < h_{l+1} < \dots < h_m > h_{m+1} > \dots > h_r$ .

Архитекторам потребовалось узнать, сколько существует *горок* в этой последовательности блоков. Они очень рассчитывают на вашу помощь!

## Формат входных данных

Первая строка содержит целое число  $t$  ( $1 \leq t \leq 100$ ) — количество наборов входных данных.

Следующие  $(2 \cdot t)$  строк описывают наборы входных данных.

Первая строка каждого набора содержит целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество блоков.

Вторая строка каждого набора содержит  $n$  целых чисел  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 10^9$ ) — высоты блоков.

Гарантируется, что  $\sum n \leq 2 \cdot 10^5$ .

## Формат выходных данных

Для каждого набора входных данных выведите одно целое число — количество *горок* в последовательности блоков.

## Пример

стандартный ввод	стандартный вывод
5	4
5	0
1 2 3 2 1	0
1	0
1000000000	6
1	
1	
6	
1 2 3 3 2 1	
10	
1 5 8 4 6 3 8 10 14 7	