Keep Your Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

VegiCat is on a mine-sweeping mission. There are n mines on a number line. Mine i is at the integer coordinate a_i. The mines are numbered in order of their coordinates, so $a_1 \le a_2 \le \cdots \le a_n$. Mine coordinates may repeat.

VegiCat tests each mine once, in the order $1, 2, \ldots, n$. For mine i, he chooses an integer coordinate b_i as the test location, which may be negative or zero. The error of this test is $|a_i - b_i|$.

For each $2 \le i \le n$, an integer l_i is given, specifying the minimum required distance between the test locations for mines $i-1$ and i. These locations must satisfy

$$|b_i - b_{i-1}| \ge l_i.$$

Help VegiCat minimize the largest test error; that is, find

$$\min_{\substack{b_1,\ldots,b_n \in \mathbb{Z} \\ |b_i - b_{i-1}| \ge l_i \quad (2 \le i \le n)}} \left(\max_{1 \le i \le n} |a_i - b_i| \right).$$

Input

The first line contains an integer T ($1 \le T \le 10^4$), the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($2 \le n \le 10^5$), the number of mines.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the coordinate of mine i. It is guaranteed that $a_1 \le a_2 \le \cdots \le a_n$.

The third line contains $n-1$ integers $l_2, l_3, \ldots, l_n$ ($1 \le l_i \le 10^9$), where l_i is the minimum distance between the test locations for mines $i-1$ and i.

It is guaranteed that the sum of n over all test cases does not exceed 10^5.

Output

For each test case, output an integer — the minimum possible value of the largest test error.

Example

standard input	standard output
2 2 1 1 1 7 2 6 12 14 14 16 19 11 8 4 2 10 9	1 6

Note

In the first test case, choosing $b_1 = 1$ and $b_2 = 2$ gives a maximum error of 1. An error of 0 would require $b_1 = b_2 = 1$, which violates the distance requirement.

In the second test case, one optimal choice is $b = [-4, 7, 15, 10, 8, 22, 13]$. All required distances are satisfied, and the maximum error is 6.