

Problem J. Modified LCS

Program: **sequence.**(cpp|java)
Input: **sequence.in**
Balloon Color: **Purple**

LCS stands for longest common subsequence, and it is a well known problem. A sequence in this problem means a list of integers, and a sequence X is considered a subsequence of another sequence Y , when the sequence X can be obtained by deleting zero or more elements from the sequence Y without changing the order of the remaining elements.

In this problem you are given two sequences and your task is to find the length of the longest sequence which is a subsequence of both the given sequences.

You are not given the sequences themselves. For each sequence you are given three integers N , F and D , where N is the length of the sequence, F is the first element in the sequence. Each element except the first element is greater than the element before it by D .

For example $N = 5$, $F = 3$ and $D = 4$ represents the following sequence: [3, 7, 11, 15, 19].

There will be at least one integer which belongs to both sequences and it is not greater than 1,000,000.

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T , the number of test cases ($1 \leq T \leq 100$). Followed by the test cases, each test case is described in one line which contains 6 integers separated by a single space $N1 F1 D1 N2 F2 D2$ ($1 \leq N1, N2 \leq 10^{18}$) and ($1 \leq F1, D1, F2, D2 \leq 10^9$) representing the length of the first sequence, the first element in the first sequence, the incremental value of the first sequence, the length of the second sequence, the first element in the second sequence and the incremental value of the second sequence, respectively.

Output

For each test case, print a single line which contains a single integer representing the length of the longest common subsequence between the given two sequences.

Examples

<code>sequence.in</code>	<code>stdout</code>
3	4
5 3 4 15 3 1	3
10 2 2 7 3 3	50
100 1 1 100 1 2	