

# Killing the Brute-force

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         512 megabytes

Chenjb is the task author of the 71-st Zhejiang Provincial Collegiate Programming Contest. He created an NP-Hard problem on a graph with  $n$  vertices, the intended solution of which is DFS with branching and pruning. Chenjb is an experienced task author, he knows that the constraints can't be too tight, so he will set the time limit  $t$  to be  $3x$ , where  $x$  is the running time of the intended solution on the slowest test case.

Chenjb heard that the contest would be unfortunately held on Potato OJ, whose hardware is not so good. To protect Potato OJ from being overloaded and the contest being a failure, Chenjb plans to cut down the input size of his task. He has tested the running time of the intended solution and the brute-force solution for various values of  $n$ . Note that for simplicity, we assume the running time of the solutions on a test case only depends on the value of  $n$ .

Please help Chenjb find the smallest value of  $n$  such that  $1 \leq n \leq m$ , and the brute-force solution won't pass the problem. Note that if the time limit is  $t$ , we assume that a solution will pass if and only if its running time on each test case is smaller than or equal to  $t$ .

## Input

The input contains multiple cases. The first line of the input contains a single integer  $T$  ( $1 \leq T \leq 50$ ), the number of cases.

For each case, the first line of the input contains a single integer  $m$  ( $1 \leq m \leq 50$ ), denoting the upper bound of  $n$ .

The second line contains  $m$  integers  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 100, a_i \leq a_{i+1}$ ), the  $i$ -th of which denotes the running time of the intended solution when  $n = i$ .

The third line contains  $m$  integers  $b_1, b_2, \dots, b_m$  ( $1 \leq b_i \leq 100, b_i \leq b_{i+1}$ ), the  $i$ -th of which denotes the running time of the brute-force solution when  $n = i$ .

## Output

For each case, print a single line containing a single integer denoting the minimum possible value of  $n$ . If there is no solution, print  $-1$  instead.

## Example

standard input	standard output
2	3
4	-1
1 2 7 20	
2 5 30 40	
3	
2 3 3	
5 7 8	