Map 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Noah has played an RPG game in which he steps on the back of an enormous moose to find a secret shipyard. The wayfinding system was terrible. Noah keeps dreaming about what would happen if the game had proper map navigation.

The map in the game can be represented by a simple polygon M. Noah cannot walk out of M. The shipyard is located at a specific point p inside (or on the boundary of) M. A proper map navigation system should always display the shortest path between p and Noah's current location. Given M, p, and several locations $a_1, \ldots, a_Q$ of Noah, write a program that computes the length of the shortest paths between a_i and p for all $1 \leq i \leq Q$.

Input

There are multiple test cases in a single test file. The first line contains a single integer T ($1 \leq T \leq 5000$) – the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($3 \leq n \leq 5000$) – the number of vertices of the simple polygon, M.

Each of the next n lines contains a vertex of M. The vertices are given in counterclockwise order.

The next line contains the target p.

The next line contains a single integer Q ($1 \leq Q \leq 5000$) – the number of queries.

Each of the next Q lines contains a query a_i.

Points are given by a pair of coordinates x and y separated by a single space. All coordinates are integers with absolute values no more than 2000. All points in the input are inside or on the boundary of M. It is guaranteed that M is simple, i.e., its vertices are distinct and no two edges of the polygon intersect or touch, other than consecutive edges which touch at their common vertex. The sum of n over all test cases does not exceed 5000. The sum of Q over all test cases does not exceed 5000.

Output

For each test case, output Q lines. The i-th line should contain the length of the shortest path between a_i and p. Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Example

standard input	standard output
1 6 0 5 4 4 4 -4 0 -5 6 -4 6 4 0 5 6 4 4 4 -4 6 4 6 -4 0 -5 5 -3	4.123105625618 12.123105625618 6.082762530298 12.369316876853 16.246211251235 11.194173437483

Note

Here is the illustration for the example.