Problem C. Yet Another Shortest Path Query

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	1024 mebibytes

In graph theory, a planar graph is a graph that can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn so that no edges cross each other. Such a drawing is called a plane graph or a planar embedding of the graph. A plane graph can be defined as a planar graph with a mapping from every node to a point on the plane, and from every edge to a plane curve on that plane, such that the extreme points of each curve are the points mapped from its end nodes, and all curves are disjoint except for their extreme points.

You will be given an undirected planar graph with n vertices and m edges. The vertices are labeled by $1, 2, \ldots, n$. The i-th edge connects vertices u_i and v_i, and its length is w_i.

You will then be given q queries. In the i-th query, you will be given two integers s_i and t_i. Please write a program to figure out the length of the shortest path from vertex s_i to vertex t_i such that the path contains **no more than three edges**, or determine that there is no such path.

Input

The first line of the input contains two integers n and m ($2 \le n \le 10^6$, $1 \le m \le 10^6$) denoting the number of vertices and the number of edges.

In the next m lines, the i-th line contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le w_i \le 10^8$) describing the i-th edge. It is guaranteed that there is at most one edge between each pair of vertices. It is also guaranteed that the graph is planar.

The next line contains a single integer q ($1 \le q \le 10^6$) denoting the number of queries.

In the next q lines, the i-th line contains two integers s_i and t_i ($1 \le s_i, t_i \le n$, $s_i \ne t_i$) describing the i-th query.

Output

For each query, print a single line containing an integer denoting the length of the shortest path containing at most three edges. If there is no such path, please print "`-1`" instead.

Examples

standard input	*standard output*
6 9 1 2 4 2 3 6 3 6 5 6 5 3 5 4 2 4 1 3 3 4 9 1 3 100 5 3 1 5 1 3 1 6 3 4 3 5 2 5	6 8 3 1 7
6 4 1 2 1 2 3 1 3 4 1 4 5 1 3 1 4 1 5 1 6	3 -1 -1