CODEFORCES β
Sponsored by Telegram

15-295 Fall #7: Shortest Paths

A. Dijkstra?

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given a weighted undirected graph. The vertices are enumerated from 1 to n. Your task is to find the shortest path between the vertex 1 and the vertex n.

Input

The first line contains two integers n and m ($2 \le n \le 10^5$, $0 \le m \le 10^5$), where n is the number of vertices and m is the number of edges. Following m lines contain one edge each in form a_i, b_i and w_i ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^6$), where a_i, b_i are edge endpoints and w_i is the length of the edge.

It is possible that the graph has loops and multiple edges between pair of vertices.

Output

Write the only integer `-1` in case of no path. Write the shortest path in opposite case. If there are many solutions, print any of them.

Sample test(s)

input
5 6 1 2 2 2 5 5 2 3 4 1 4 1 4 3 3 3 5 1
output
1 4 3 5

input
5 6 1 2 2 2 5 5 2 3 4 1 4 1 4 3 3 3 5 1
output
1 4 3 5

B. String Problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Boy Valera likes strings. And even more he likes them, when they are identical. That's why in his spare time Valera plays the following game. He takes any two strings, consisting of lower case Latin letters, and tries to make them identical. According to the game rules, with each move Valera can change **one** arbitrary character A_i in one of the strings into arbitrary character B_i, but he has to pay for every move a particular sum of money, equal to W_i. He is allowed to make as many moves as he needs. Since Valera is a very economical boy and never wastes his money, he asked you, an experienced programmer, to help him answer the question: what minimum amount of money should Valera have to get identical strings.

Input

The first input line contains two initial non-empty strings s and t, consisting of lower case Latin letters. The length of each string doesn't exceed 10^5. The following line contains integer n $(0 \le n \le 500)$ — amount of possible changings. Then follow n lines, each containing characters A_i and B_i (lower case Latin letters) and integer W_i $(0 \le W_i \le 100)$, saying that it's allowed to change character A_i into character B_i in any of the strings and spend sum of money W_i.

Output

If the answer exists, output the answer to the problem, and the resulting string. Otherwise output `-1` in the only line. If the answer is not unique, output any.

Sample test(s)

input
uayd uxxd 3 a x 8 x y 13 d c 3

output
21 uxyd

input
a b 3 a b 2 a b 3 b a 5

output
2 b

input
abc ab 6 a b 4 a b 7 b a 8 c b 11 c a 3 a c 0

output
-1

C. Roads in Berland

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n cities numbered from 1 to n in Berland. Some of them are connected by two-way roads. Each road has its own length — an integer number from 1 to 1000. It is known that from each city it is possible to get to any other city by existing roads. Also for each pair of cities it is known the shortest distance between them. Berland Government plans to build k new roads. For each of the planned road it is known its length, and what cities it will connect. To control the correctness of the construction of new roads, after the opening of another road Berland government wants to check the sum of the shortest distances between all pairs of cities. Help them — for a given matrix of shortest distances on the old roads and plans of all new roads, find out how the sum of the shortest distances between all pairs of cities changes after construction of each road.

Input

The first line contains integer n ($2 \le n \le 300$) — amount of cities in Berland. Then there follow n lines with n integer numbers each — the matrix of shortest distances. j-th integer in the i-th row — $d_{i,j}$, the shortest distance between cities i and j. It is guaranteed that $d_{i,i} = 0$, $d_{i,j} = d_{j,i}$, and a given matrix is a matrix of shortest distances for some set of two-way roads with integer lengths from 1 to 1000, such that from each city it is possible to get to any other city using these roads.

Next line contains integer k ($1 \le k \le 300$) — amount of planned roads. Following k lines contain the description of the planned roads. Each road is described by three space-separated integers a_i, b_i, c_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$, $1 \le c_i \le 1000$) — a_i and b_i — pair of cities, which the road connects, c_i — the length of the road. It can be several roads between a pair of cities, but no road connects the city with itself.

Output

Output k space-separated integers q_i ($1 \le i \le k$). q_i should be equal to the sum of shortest distances between all pairs of cities after the construction of roads with indexes from 1 to i. Roads are numbered from 1 in the input order. Each pair of cities should be taken into account in the sum exactly once, i. e. we count unordered pairs.

Sample test(s)

input
2 0 5 5 0 1 1 2 3
output
3

input
3 0 4 5 4 0 9 5 9 0 2 2 3 8 1 2 1
output
17 12

D. Volleyball

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya loves volleyball very much. One day he was running late for a volleyball match. Petya hasn't bought his own car yet, that's why he had to take a taxi. The city has n junctions, some of which are connected by two-way roads. The length of each road is defined by some positive integer number of meters; the roads can have different lengths.

Initially each junction has exactly one taxi standing there. The taxi driver from the i-th junction agrees to drive Petya (perhaps through several intermediate junctions) to some other junction if the travel distance is not more than t_i meters. Also, the cost of the ride doesn't depend on the distance and is equal to c_i bourles. Taxis can't stop in the middle of a road. **Each taxi can be used no more than once. Petya can catch taxi only in the junction, where it stands initially.**

At the moment Petya is located on the junction x and the volleyball stadium is on the junction y. Determine the minimum amount of money Petya will need to drive to the stadium.

Input

The first line contains two integers n and m ($1 \le n \le 1000, 0 \le m \le 1000$). They are the number of junctions and roads in the city correspondingly. The junctions are numbered from 1 to n, inclusive. The next line contains two integers x and y ($1 \le x, y \le n$). They are the numbers of the initial and final junctions correspondingly. Next m lines contain the roads' description. Each road is described by a group of three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$) — they are the numbers of the junctions connected by the road and the length of the road, correspondingly. The next n lines contain n pairs of integers t_i and c_i ($1 \le t_i, c_i \le 10^9$), which describe the taxi driver that waits at the i-th junction — the maximum distance he can drive and the drive's cost. The road can't connect the junction with itself, but between a pair of junctions there can be more than one road. All consecutive numbers in each line are separated by exactly one space character.

Output

If taxis can't drive Petya to the destination point, print "-1" (without the quotes). Otherwise, print the drive's minimum cost.

Please do not use the %lld specificator to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the %I64d specificator.

Sample test(s)

input
4 4 1 3 1 2 3 1 4 1 2 4 1 2 3 5 2 7 7 2 1 2 7 7
output
9

Note

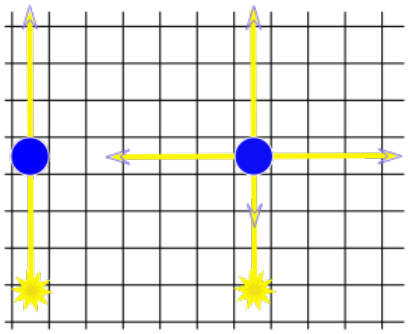
An optimal way — ride from the junction 1 to 2 (via junction 4), then from 2 to 3. It costs 7+2=9 bourles.

E. Chamber of Secrets

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

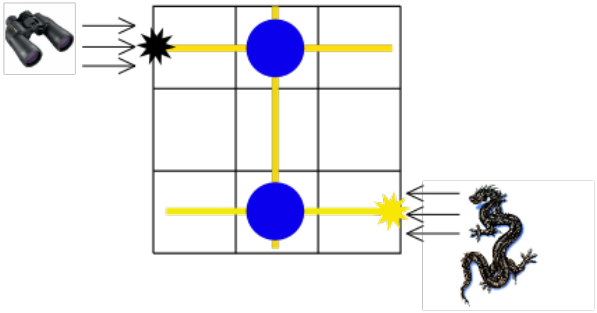
"The Chamber of Secrets has been opened again" — this news has spread all around Hogwarts and some of the students have been petrified due to seeing the basilisk. Dumbledore got fired and now Harry is trying to enter the Chamber of Secrets. These aren't good news for Lord Voldemort. The problem is, he doesn't want anybody to be able to enter the chamber. The Dark Lord is going to be busy sucking life out of Ginny.

The Chamber of Secrets is an $n \times m$ rectangular grid in which some of the cells are columns. A light ray (and a basilisk's gaze) passes through the columns without changing its direction. But with some spell we can make a column magic to reflect the light ray (or the gaze) in all four directions when it receives the ray. This is shown in the figure below.



The left light ray passes through a regular column, and the right ray — through the magic column.

The basilisk is located at the right side of the lower right cell of the grid and is looking to the left (in the direction of the lower left cell). According to the legend, anyone who meets a basilisk's gaze directly dies immediately. But if someone meets a basilisk's gaze through a column, this person will get petrified. We know that the door to the Chamber is located on the left side of the upper left corner of the grid and anyone who wants to enter will look in the direction of its movement (in the direction of the upper right cell) from that position.



This figure illustrates the first sample test.

Given the dimensions of the chamber and the location of regular columns, Lord Voldemort has asked you to find the minimum number of columns that we need to make magic so that anyone who wants to enter the chamber would be petrified or just declare that it's impossible to secure the chamber.

Input

The first line of the input contains two integer numbers n and m ($2 \le n, m \le 1000$). Each of the next n lines contains m characters. Each character is either "." or "#" and represents one cell of the Chamber grid. It's "." if the corresponding cell is empty and "#" if it's a regular column.

Output

Print the minimum number of columns to make magic or -1 if it's impossible to do.

Sample test(s)

input
3 3 .#. ... .#.
output

```
2
```

input

```
4 3
##.
...
.#.
.#.
```

output

```
2
```

Note

The figure above shows the first sample test. In the first sample we should make both columns magic. The dragon figure represents the basilisk and the binoculars represent the person who will enter the Chamber of secrets. The black star shows the place where the person will be petrified. Yellow lines represent basilisk gaze moving through columns.