Problem A. Webinar is coming!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Oups! This happened again!

Webinar on Algorithms and Data Structures is about to start, and you are still not at home. In the previous problem (L7.B) you were rather close to home — you had just to go through park. But now you are in the another city!

You've taken a taxi and driver asked you to show the way. You have a map of the country and you decided to use it to find the shortest way home. Your map is rather simple: there is no roads which end in the same city they started in and there's no more than one road between each pair of cities. All roads in your country are both-directional.

You are in a hurry, because you want to attend webinar with a cup of your favorite tea.

More formally: given simple undirected weighted graph and two vertices s, h. Find the weight of the shortest path and sequence of vertices in any shortest path $s \to h$.

Input

First line contains 2 integer numbers $1 \leq N \leq 100000, 1 \leq M \leq 200000$ — number of cities (vertices) and roads (edges) in the country (graph).

Second line contains two integer numbers: $0 \leq s, h < N$ — number of the city you are in and number of your home city, $s \neq h$.

Each of following M lines defines an edge $u_i \to v_i$ with weight w_i and contains three integer numbers: $u_i, v_i, w_i : 0 \leq u_i, v_i < N, 0 \leq w_i\ leq 10000$.

Output

On the first line print weight of the optimal path $s \to h$ or -1 if path doesn't exist.

If path exists, the second line should contain number of vertices in the path k.

If path exists, the third line should contain k integer numbers — vertices in the optimal path in the order they will be visited (first one is s, last one is h).

Examples

standard input	standard output

4 4
0 2
0 1 1
1 3 1
3 2 2
0 2 5
	4
	4
	0 1 3 2

4 2
0 2
0 1 1
2 3 1
	-1

5 7
0 2
1 2 5
1 3 2
2 3 4
2 4 3
3 4 6
0 3 20
0 4 10
	13
	3
	0 4 2

Problem B. Student ID

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Covid-19 pandemic is finally over! State borders open again and airlines return to work. You want to visit MIPT, because you need to get your student id. Also you want to see sights of Dolgoprudny.

You have K days to take student ID, because you are going to use it to get discount in K days. So, you need to get from your home city to Moscow not later than in K days. Also, you decided that you don't want to have more than one flight per day, because it's too hard. So, this means, you can't have more than K flights in your path.

Also, you want to minimize total cost of your flights. You have obtained schedules of flights (list of flights is the same every day). Now you just need to implement an algorithm which will choose the best path for you.

Input

The first line contains 4 integer numbers: $2 \le N \le 100; 0 \le s, f < N; 0 \le K < N$ — number of cities in the schedule, id of your home city (s) and id of Moscow airport (f), and K — maximum number of flights you can make.

Next N lines contain N integer numbers each and define matrix a (adjacency matrix of the graph). Value $a_{i,j}$ in i-th row j-th column is the cost of flight from city i to city j. If there's no flights from i to j, this value equals -1: $-1 \le a_{i,j} \le 10000$. It is guaranteed that a contains only -1 values on main diagonal.

Output

Print single integer number — minimum cost of the path $s \to f$ which contains not more than K flights.

Examples

standard input	standard output

5 0 4 3
-1 100 10 -1 1000
-1 -1 -1 -1 100
-1 -1 -1 10 -1
-1 -1 -1 -1 10
-1 -1 -1 -1 -1
	30

5 0 4 2
-1 100 10 -1 1000
-1 -1 -1 -1 100
-1 -1 -1 10 -1
-1 -1 -1 -1 10
-1 -1 -1 -1 -1
	200

5 0 4 1
-1 100 10 -1 1000
-1 -1 -1 -1 100
-1 -1 -1 10 -1
-1 -1 -1 -1 10
-1 -1 -1 -1 -1
	1000

4 0 3 2
-1 1 3 5
-1 -1 1 -1
-1 -1 -1 1
-1 -1 -1 -1
	4