

Problem J. Setting Maps

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 1024 mebibytes

I intend to install maps on the trail. The trail can be thought of as a graph consisting of N vertices, numbered from 1 to N , and M directed edges. There are no self-loops and no multiple directed edges.

The trail has a starting point S and a destination E . For better convenience, I want to put maps in such a way that all routes from S to E go through at least K maps.

Only one map can be installed per vertex (including starting and ending points), and the cost of installing a map at vertex v is C_v .

I want to complete the map installation at the lowest possible cost. Determine if the maps can be installed to meet the conditions, and if it is possible, print out in which vertices the maps should be installed so that the total cost is minimal possible.

Input

In the first line of the input, the number of vertices N , the number of edges M , and the minimum number of maps K are given ($2 \leq N \leq 200$, $1 \leq M \leq 500$, $1 \leq K \leq 5$).

In the second line, the starting point S and the destination point E are given ($1 \leq S, E \leq N$, $S \neq E$).

The next line contains N integers C_1, C_2, \dots, C_N : the costs of installing a map in vertices $1, 2, \dots, N$ ($1 \leq C_i \leq 10^7$).

The next M lines describe edges. The j -th of these lines contains two integers u_j and v_j : the source and destination of the j -th edge ($1 \leq u_j, v_j \leq N$, $u_j \neq v_j$).

It is guaranteed that the trail contains no self-loops and no multiple directed edges.

Output

If it is not possible to install maps to satisfy the conditions, -1 must be printed on the first line.

If map installation is possible, on the first line, print the number of vertices P at which the map should be installed so that the total cost is minimal possible. On the second line, print the labels of these P vertices, separated by spaces, in any order. If there are several possible answers, print any one of them.

Examples

standard input	standard output
3 2 5 1 3 1 60 35 1 2 2 3	-1
7 11 1 1 7 100 5 7 16 11 12 100 1 2 1 3 1 4 1 5 2 3 2 6 3 6 4 3 4 7 5 7 6 7	3 5 6 4