Outsmarting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Given an undirected bipartite graph $G = (U, V, E)$, where U and V are sets of vertices of the two parts and $E \subseteq U \times V$ is the set of edges. Some vertices are marked as "disabled", while the others are "enabled". There is a piece on the graph. Two players, the first player and the second player, move the piece alternately. The first player starts from vertex 1 in the left part U of the bipartite graph. The movement rules are as follows:

- On a player's turn, if the piece is on vertex x, the player must choose an edge (x, y) such that vertex y is enabled;
- The player moves the piece to y and marks x as disabled;
- If a player cannot find a movable edge, that player loses and the opponent wins.

Before the game starts, a "flip" operation can be applied to some vertices: changing an enabled vertex to disabled, or a disabled vertex to enabled. Each vertex can be flipped at most once, and the starting vertex cannot be disabled.

Determine the minimum number of flips needed to guarantee a win for the first player, assuming both players play optimally. If it is impossible for the first player to win, output "-1".

Input

There is only one test case in each test file.

The first line contains three integers n_U, n_V and m ($1 \le n_U, n_V \le 100$, $0 \le m \le n_U \cdot n_V$), representing the size of the left part U, the size of the right part V, and the number of edges connecting the two parts, respectively.

The next m lines each contain two integers u and v ($1 \le u \le n_U$, $1 \le v \le n_V$), representing an edge connecting vertex u in the left part and vertex v in the right part.

The next line contains n_U integers $s_{u,1}, s_{u,2}, \ldots, s_{u,n_U}$ ($s_{u,i} \in \{0, 1\}$, $s_{u,1} = 1$). If $s_{u,i} = 0$, the i-th vertex in the left part is disabled; otherwise, it is enabled.

The next line contains n_V integers $s_{v,1}, s_{v,2}, \ldots, s_{v,n_V}$ ($s_{v,i} \in \{0, 1\}$). If $s_{v,i} = 0$, the i-th vertex in the right part is disabled; otherwise, it is enabled.

Output

Output a single integer representing the minimum number of flips required to guarantee a win for the first player. If it is impossible to guarantee a win, output "-1".

Examples

standard input	standard output
4 4 8 1 1 2 1 2 2 3 1 3 3 4 2 4 3 4 4 1 1 1 1 1 1 1 0	1
4 2 5 1 1 2 1 2 2 3 2 4 2 1 1 1 1 1 1	1
1 1 0 1 0	-1