Mock CCC '23 2 S4 - Roger Says Yabe

Time limit: 0.25s **Memory limit:** 1G

Roger likes saying yabe. Some people think he might be trying to say eBay, but obviously, he's saying yabe.

Two strings are *equivalent* if the two strings are equal after removing all lowercase letters.

Given a graph G of N nodes and M labelled directed edges with D destination nodes, we define $L(G)$ to be the *language* of the graph. Imagine a walk that starts at node 1 and ends at some destination node. Let S be the string obtained by concatenating all the labels of the edges that are walked in order. $L(G)$ is thus defined as all possible strings S that can be generated by this process.

Compute the minimum possible value of $|s_1| + |s_2|$ where s_1 and s_2 are both in $L(G)$, $s_1 \neq s_2$, and s_1 and s_2 are equivalent.

Constraints

$1 \leq D \leq N \leq 50$

$1 \leq M \leq 52N$

$1 \leq v_1, v_2 \leq N$

If two edges go out of a common vertex, they are labelled with distinct letters.

Input Specification

The first line contains three integers, N, D, and M.

Each of the next D lines contains a unique positive integer less than or equal to N. These represent the destination nodes.

Each of the next M lines contains an integer v_1, a lowercase or uppercase letter c, and another integer v_2, indicating a directed edge labeled with c going from vertex v_1 to vertex v_2.

Output Specification

Output the minimum possible value of $|s_1| + |s_2|$ where s_1 and s_2 are both in $L(G)$, $s_1 \neq s_2$, and s_1 and s_2 are equivalent. If no such value exists, output `-1`.

Sample Input

```
4 1 8
3
1 F 1
1 A 2
2 b 1
2 F 3
2 d 3
3 B 3
3 y 4
4 d 1
```

Sample Output

```
10
```