

## Bud words

A word  $W$  in a language  $L$  is called a **bud word** if the length of any other word of  $L$  is bigger than or equal to the length of  $W$ .

### The task

Find the lexicographically smallest bud word among all words accepted by both given finite automata  $A$ ,  $B$ .

---

### Input

The first line contains a single string specifying the alphabet  $\Sigma$  of two automata  $A$  and  $B$ . The alphabet is a subset of small English letters  $\{'a', 'b', \dots, 'z'\}$ . The next part of input is the specification of automata  $A$  and  $B$ . The specification format is the same for both automata. The first line of specification contains two integers  $N$  and  $M$  separated by space.  $N$  is the number of states of the automaton,  $M$  is the number of transitions. The states are labeled by integers  $0, 1, \dots, N-1$ . Next, there are  $M$  input lines, each represents one transition. A transition is described by three items  $S, C, T$  separated by space.  $S$  and  $T$  are labels of some states,  $C$  is one of alphabet  $\Sigma$  characters. Let us denote the automaton transition function by symbol  $\delta$ . Then it holds  $T \in \delta(S, C)$ . Last line of the automaton specification contains a list of labels of all final states of the automaton, the items in the list are separated by spaces. The label of automaton start state is always 0. There are no blank lines in the input, the specification of automaton  $A$  is followed immediately by the specification of automaton  $B$ .

It holds  $N \leq 1000$ ,  $M \leq 4000$ ,  $2 \leq |\Sigma| \leq 10$ . The automata are not necessarily deterministic.

### Output

The output is one line containing a single string representing the lexicographically smallest bud word among all words accepted by both input automata. The bud words are unempty in all cases in this problem.

#### Example 1

##### Input

```
ab
4 7
3 b 3
2 a 3
3 a 2
2 b 1
1 b 2
1 a 1
0 b 2
1
5 7
4 b 1
3 b 1
3 a 4
2 a 3
1 a 2
0 a 0
0 b 1
3 4
```

##### Output

```
baabaa
```

#### Example 2

##### Input

```
abcd
3 6
0 b 1
1 b 2
2 a 2
2 b 2
2 c 2
2 d 2
2
8 14
0 a 1
1 b 2
2 c 3
3 d 4
0 d 5
5 c 6
6 b 7
7 a 4
4 a 4
4 b 4
4 c 4
4 d 4
0 b 0
0 c 0
4 6 7
```

##### Output

```
bbdc
```

---

### Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.

[Link to public data set](#)