B. Bear and Compressing

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Limak is a little polar bear. Polar bears hate long strings and thus they like to compress them. You should also know that Limak is so young that he knows only first six letters of the English alphabet: '`a`', '`b`', '`c`', '`d`', '`e`' and '`f`'.

You are given a set of q possible operations. Limak can perform them in any order, any operation may be applied any number of times. The i-th operation is described by a string a_i of length two and a string b_i of length one. No two of q possible operations have the same string a_i.

When Limak has a string s he can perform the i-th operation on s if the first two letters of s match a two-letter string a_i. Performing the i-th operation removes first two letters of s and inserts there a string b_i. See the notes section for further clarification.

You may note that performing an operation decreases the length of a string s exactly by 1. Also, for some sets of operations there may be a string that cannot be compressed any further, because the first two letters don't match any a_i.

Limak wants to start with a string of length n and perform $n - 1$ operations to finally get a one-letter string "`a`". In how many ways can he choose the starting string to be able to get "`a`"? Remember that Limak can use only letters he knows.

Input

The first line contains two integers n and q $(2 \le n \le 6, 1 \le q \le 36)$ — the length of the initial string and the number of available operations.

The next q lines describe the possible operations. The i-th of them contains two strings a_i and b_i $(|a_i| = 2, |b_i| = 1)$. It's guaranteed that $a_i \ne a_j$ for $i \ne j$ and that all a_i and b_i consist of only first six lowercase English letters.

Output

Print the number of strings of length n that Limak will be able to transform to string "`a`" by applying only operations given in the input.

Examples

input Copy

```
3 5
ab a
cc c
ca a
ee c
ff d
```

output Copy

```
4
```

input Copy

```
2 8
af e
dc d
cc f
bc b
da b
eb a
bb b
ff c
```

output Copy

```
1
```

input Copy

```
6 2
bb a
ba a
```

output Copy

```
0
```

Note

In the first sample, we count initial strings of length 3 from which Limak can get a required string "a". There are 4 such strings: "abb", "cab", "cca", "eea". The first one Limak can compress using operation 1 two times (changing "ab" to a single "a"). The first operation would change "abb" to "ab" and the second operation would change "ab" to "a".

Other three strings may be compressed as follows:

- "cab" $\rightarrow$ "ab" $\rightarrow$ "a"
- "cca" $\rightarrow$ "ca" $\rightarrow$ "a"
- "eea" $\rightarrow$ "ca" $\rightarrow$ "a"

In the second sample, the only correct initial string is "eb" because it can be immediately compressed to "a".