

7518 Palindromic Obsession

Countries on planet Pali follow the following rules for labeling one-way roads between cities: 1) each road is labeled by a single lower-case letter, 2) for every city the roads leaving it receive different labels (roads that enter a city can have identical labels). All roads on Pali are one-way. This system makes it very easy to give driving directions — they are just strings of lower-case letters. Dezider wants to drive from his city to the capital but he has the following peculiar set of constraints on the drive:

- 1) it has to take exactly l days,
- 2) each day he will drive on just one road,
- 3) the directions for the drive must be a palindrome (that is, a string that reads the same forward and backward).

He asked Ružena (an agent in a trip planning agency) to give him a list of all strings that satisfy his constraints. Ružena told him that there are too many such strings. Now Dezider wants to know exactly how many strings satisfy his constraints. Please, help Ružena and write an algorithm that will count the strings that satisfy Dezider's constraints.

Input

The first line contains k , the number of countries on planet Pali. Each country is described on several lines. The first line contains four numbers:

- 1) $n \leq 10$, the number of cities,
- 2) Dezider's starting city (we refer to the cities by numbers $1, \dots, n$),
- 3) the capital,
- 4) $l \leq 100$, the length of the drive.

The next line contains $m \leq 100$, the number of roads. Then m lines follow, each describing a road. Each line describing a road consists of

- 1) the label of the road (a letter chosen from 'a', 'b', ..., 'z'),
- 2) the city where the road starts, and
- 3) the city where the road ends.

Output

The output contains one line for each country. The line contains the number of palindromes of length l that are valid directions from Dezider's starting city to the capital.

Explanation:

There is no even length path from city 1 to city 2 in the first example.

In the second example all palindromes of length 99 that use letters a, b are valid.

Sample Input

```
3
2 1 2 10
4
a 1 2
a 2 1
b 1 2
b 2 1
2 1 2 99
4
a 1 2
a 2 1
b 1 2
b 2 1
```

Sample Output

```
0
1125899906842624
```