Invasion

`invasion.c`, `invasion.C`, `invasion.java`, `invasion.p`

Alien invasion began, and scary man-eating aliens are establishing their bases all over the country. You are only safe on the places that are sufficiently far away from all current alien bases. You need to quickly write a program to help you determine where to move.

Input Specification

The input contains several instances, each of them consisting of several lines. The first line of each instance contains integers N ($1 \leq N \leq 10\,000$), M ($0 \leq M \leq 100\,000$), A ($0 \leq A \leq N$) and K ($1 \leq K \leq 100$) separated by spaces, giving the number of towns in the country, the number of roads between them, the number of bases that the aliens are going to build, and the minimum safe distance from alien bases, respectively. The towns are assigned numbers $1, \ldots, N$.

The following M lines describe the roads; each of them contains integers T_1, T_2 ($1 \leq T_1 < T_2 \leq N$) and D ($1 \leq D \leq 100$) separated by spaces, where D is the length of the road between towns T_1 and T_2. There is at most one direct road between any two towns. All roads can be used in both directions.

The following A lines describe the positions of alien bases; the i-th of them contains the number B_i ($1 \leq B_i \leq N$) of the town where the aliens build their i-th base.

Each instance is followed by one empty line. The empty line after the last instance is followed by a line containing four zeros.

Output Specification

The output for each input instance consists of A lines. On i-th line, write the number of towns that are safe after the aliens build their i-th base. The town is safe if its distance from *any* of the towns $B_1, B_2, \ldots, B_i$ is at least K.

Print one empty line after each instance.

Sample Input

```
7 6 3 3
1 2 1
1 3 1
2 5 1
3 6 1
1 4 1
4 7 2
2
1
4

1 0 1 1
1

0 0 0 0
```

Output for Sample Input

```
2
1
0

0
```