The *Agency for Cross-Constellation and Interstellar Space Travel* (ACIS) is ready to offer its clients space travel among several planets across the universe.

ACIS offers a list of flight options consisting of an origin planet, a destination planet, a cost, and a duration. One of the "killer" features ACIS will offer to its clients is that of being able to plan a trip between two planets under the constraint of a maximum number of stops. That is, given a natural number n, ACIS would like to offer each client the cheapest possible trip from an origin planet to a destination planet with at most n stops. Since interstellar in-flight sleep is not pleasant, it is also important to minimize the amount of time spent in a trip.

Can you help ACIS in finding an efficient algorithm for such a task?

Input

The input consists of several test cases. Each test case begins with a line with three blank-separated integers p, f, and q ($1 \leq p \leq 300$, $0 \leq f \leq 5000$, and $0 \leq q \leq 1000$), indicating the number of planets, flights, and queries, respectively. The next p lines each contains a planet name s ($1 \leq |s| \leq 30$). The next f lines each contains two planet names and two integers s_o, s_d, c, and t (separated by a blank), denoting that there is a direct flight from s_o to s_d costing c dollars ($0 \leq c \leq 10^5$) with a duration of t units of time ($0 \leq t \leq 10^5$). The next line contains a planet name s_i indicating the initial planet for the trip. The next q lines each contains a query with a destination planet name s_f for the trip and a natural number n, both separated by a blank ($0 \leq n \leq 300$). You can assume that planet names consist only of alphabetic characters, and that s_o, s_d, s_i, and s_f are in the list of p planet names.

Output

For each query s_i, s_f, n output two blank-separated integers indicating the minimum cost and the corresponding minimum travel time for this cost of an interstellar trip from s_i to s_f with at most n stops. If this is not possible, then print two blank-separated asterisks ('`*`').

Print a line with a single period ('`.`') between consecutive test cases.

Sample Input

```
2 3 1
Earth
Mars
Earth Mars 2 3
Earth Mars 4 1
Earth Earth 3 2
Earth
Mars 0
3 3 5
Tatooine
Endor
Geonosis
Tatooine Endor 300 15
Endor Geonosis 10 78
Geonosis Tatooine 1 1
Endor
Endor 0
Geonosis 0
Geonosis 4
Tatooine 0
Tatooine 1
5 5 8
Earth
Kaishin
Namek
Vegeta
NewNamek
Earth Kaishin 10 10
Kaishin Namek 10 5
Kaishin Vegeta 15 30
Earth Vegeta 25 50
NewNamek Earth 100 1
Earth
Kaishin 0
Kaishin 1
Kaishin 2
Namek 0
Namek 1
Vegeta 0
Vegeta 1
NewNamek 5
```

Sample Output

```
2 3
.
0 0
10 78
10 78
* *
11 79
.
10 10
10 10
10 10
* *
20 15
25 50
25 40
* *
```