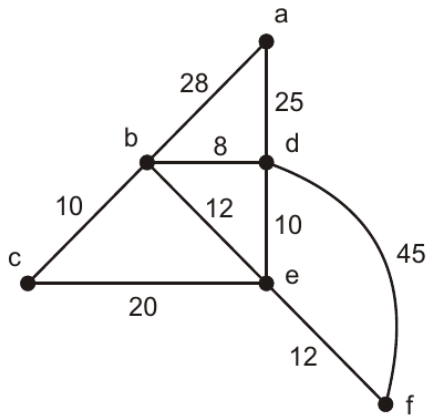
ACM-ICPC Live Archive

2977 Johnny the Taxi Driver

Johnny is a dishonest taxi driver who just loves to get more money out of unaware tourists. When he picks up a tourist, Johnny always attempts to drive more than necessary in order to be able to charge the tourist more. However, Johnny realizes that even tourists visiting the city for the first time, will be able to uncover his scheme if he keeps going around the same places too many times. Johnny must find paths with minimum revisits of the same places, and never more than 10 revisits. For that Johnny needs your help.



Johnny will give you a detailed description of the city, listing distances between intersection points. Johnny wants you to write a program that will determine a path between any two given points, such that the path will have a preset cost.

For example: In the map on the right, Rather than going from '`a`' to '`f`' using the cheapest (shortest) path '`adef`' (with *cost* = 47) Johnny, being greedy, can go '`abcebdef`' or '`adebdf`' both with *cost* = 100. The second path '`adebdf`' revisits only one point ('`d`') is better than the first '`abcebdef`' which has two revisits ('`b`' and '`e`'.)

Input

Your program will be tested on a number of test cases. The first line of the input file contains a single integer T denoting the number of test cases.

Each test case is specified on $n + q + 1$ lines. The first line of each test case specifies three integers: p (the number of intersection points,) n (the number of streets,) and q (the number of queried paths.)

Following that, the streets will be specified using n lines, one for each street. A street between $PointA$ and $PointB$ with cost C is specified using the following format:

$PointA_{\sqcup}PointB_{\sqcup}C$

Each test case has at least one street and at most 100 streets. All streets are two ways. A $Point$ is specified using a lower-case word whose length doesn't exceed 8 characters. No city will have more than 100 distinct points. C is a positive integer ≤ 100

Following the n input lines describing the streets, each test case will have q queries, each specified on a separate line using the following format:

$PointX_{\sqcup}PointY_{\sqcup}Q$

which inquires about the minimum number of revisits in a path from $PointX$ to $PointY$ having cost Q.

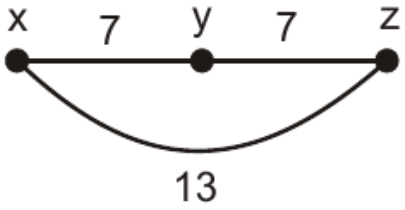
Output

For each query, write on a separate line, the following output:

$PointX_{\sqcup}PointY_{\sqcup}Q_{\sqcup}R$

Where $PointX$, $PointY$, and Q are the same as the input, and R is the minimum number of revisits in a path going from $PointX$ to $PointY$ with cost Q. If there is no such path with no more than 10 revisits, then R is equal to '`-1`'.

Note: The second test case uses the graph below and inquires about a path from '`x`' to '`z`' with *cost* = 65. This can only be achieved using the path '`xzxzxz`' which has 4 revisits ('`x`' is revisited twice, and so is '`z`').



Sample Input

```
2
6 9 2
a b 28
a d 25
b c 10
b d 8
b e 12
c e 20
d e 10
d f 45
e f 12
a f 100
a f 50
3 3 1
x y 7
z y 7
z x 13
x z 65
```

Sample Output

```
a f 100 1
a f 50 -1
x z 65 4
```