

After winning a gruesome battle against the Romans in a far-away land, Asterix and his dearest friend Obelix are now returning home. However Obelix is not with Asterix now. He has left Asterix in order to deliver menhir to one of his international buyers (as you probably know, recently he has extended his trade to international markets). But he has promised to join Asterix on his way home and Asterix has promised to host a feast for Obelix (you know how fat he is!) in the city they meet. Obelix may meet Asterix in any city on his way home including the starting and the destination city.

Now Asterix is sitting with a map and trying to figure out the cheapest route home. The map shows the cities and the cost (in sestertii) of going from one city to another if there is a road connecting them directly. For each city in the map Asterix has also calculated the cost (in sestertii) of hosting a feast for Obelix in that city. There will be only one feast and for safety Asterix has decided to set aside enough sestertii to host a feast in the costliest city on the route.



Since Asterix does not have a computer, he seeks your help to find out the cheapest route home.

Input

The input may contain multiple test cases.

The first line of each test case contains three integers $C (\leq 80)$, $R (\leq 1000)$ and $Q (\leq 6320)$ where C indicates the number of cities (cities are numbered using distinct integers ranging from 1 to C), R represents the number of roads and Q is the number of queries.

The next line contains C integers where the i -th integer f_i is the cost (in sestertii) of hosting a feast in city i .

Each of the next R lines contains three integers: c_1 , $c_2 (\neq c_1)$ and d indicating that the cost of going from city c_1 to c_2 (or from c_2 to c_1) is d sestertii.

Each of the next Q lines contains two integers c_1 and c_2 ($c_1 \neq c_2$) asking for the cost (in sestertii) of the cheapest route from city c_1 to city c_2 .

The input will terminate with three zeros form C , S and Q .

Output

For each test case in the input first output the test case number (starting from 1) as shown in the sample output. Then for each query in the input print a line giving the minimum cost (in sestertii) of going from the first to the second city in the query. If there exists no path between them just print '-1'.

Print a blank line between two consecutive test cases.

Sample Input

```
7 8 5
2 3 5 15 4 4 6
1 2 20
1 4 20
1 5 50
2 3 10
3 4 10
3 5 10
4 5 15
6 7 10
1 5
1 6
5 1
3 1
6 7
4 4 2
2 1 8 3
1 2 7
1 3 5
2 4 8
3 4 6
1 4
2 3
0 0 0
```

Sample Output

```
Case #1
45
-1
45
35
16

Case #2
18
20
```