

D. Instant Messenger

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

User ainta decided to make a new instant messenger called "aintalk". With aintalk, each user can chat with other people. User ainta made the prototype of some functions to implement this thing.

1. `login(u)`: User u logs into aintalk and becomes online.
2. `logout(u)`: User u logouts and becomes offline.
3. `add_friend(u , v)`: User u and user v become friends. It means, u and v can talk with each other. The friendship is bidirectional.
4. `del_friend(u , v)`: Unfriend user u and user v . It means, u and v cannot talk with each other from then.
5. `count_online_friends(u)`: The function returns the number of friends of user u who are online at the moment.

Because the messenger is being tested by some users numbered from 1 to n , there is no register method. This means, at the beginning, some users may be online, and some users may have friends.

User ainta is going to make these functions, but before making the messenger public, he wants to know whether he is correct. Help ainta verify his code.

Input

The first line contains three space-separated integers n , m and q ($1 \leq n \leq 50000$; $1 \leq m \leq 150000$; $1 \leq q \leq 250000$) — the number of users, the number of pairs of friends, and the number of queries.

The second line contains an integer o ($1 \leq o \leq n$) — the number of online users at the beginning. The third line contains o space-separated integers x_1, x_2, \dots, x_o ($1 \leq x_i \leq n$) — the ids of the online users. It is guaranteed that these values are distinct.

Each of the next m lines contains two space-separated integers a_i and b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$) — the ids of two users who are friends at the beginning. It is guaranteed there are no multiple friendship given in the input. Note that the friendship is bidirectional.

Next q lines describe the q queries in the format:

- "O u " ($1 \leq u \leq n$): Call `online(u)`. It is guaranteed that user u was offline just before the function call.
- "F u " ($1 \leq u \leq n$): Call `offline(u)`. It is guaranteed that user u was online just before the function call.
- "A u v " ($1 \leq u, v \leq n$; $u \neq v$): Call `add_friend(u , v)`. It is guaranteed that these two users weren't friends just before the function call.
- "D u v " ($1 \leq u, v \leq n$; $u \neq v$): Call `del_friend(u , v)`. It is guaranteed that these two users were friends just before the function call.
- "C u " ($1 \leq u \leq n$): Call `count_online_friends(u)` and print the result in a single line.

Output

For each `count_online_friends(u)` query, print the required answer in a single line.

Examples

input
5 2 9 1 4

1 3
3 4
C 3
A 2 5
O 1
D 1 3
A 1 2
A 4 2
C 2
F 4
C 2

output

1
2
1