Fall 2021 CIS 3362 Homework #7: Group Diffie-Hellman Key Exchange
Check WebCourses for the due date

Directions: Submit your solution, GroupDH.java or GroupDH.py via WebCourses by the due date posted on WebCourses. No late assignments accepted. In the header comment of your file, please include your name. Remember, do standard in, standard out!!!

In this assignment you will simulate adding and deleting users to a company and the management of their group keys. To simplify grading, your program will read in a list of commands (either adding a user or deleting a user), which will always specify the newly chosen secret key for the appropriate individual, which will lead to a deterministic set of group keys.

Input Format

The first line of the input file will contain two integers, *p* and *g*, separated by spaces. Both integers will be less than 100 digits long (so use BigInteger in Java!!!) and will represent a prime number and a generator for that prime number, respectively.

The following line will contain a single integer, *n (n < 1000)*, representing the number of modifications to the tree structure in the Group Diffie-Hellman protocol or queries for particular keys.

The first of these lines will have a unique format because it will add two users at once. Its format is as follows:

USER1 secretkey1 USER2 secretkey2 KeyID

Both users will be uppercase strings of length 20 or less and both secretkeys will be positive integers less than *p*. These two users will be added to the group first and share a single secret key. KeyID will be a string of 1 to 20 uppercase characters labeling that particular shared key.

The following n-1 lines will have one of the three following formats:

ADD USER1 secretkey1 USER2 secretkey2 KeyID
DEL USER1 secretkey2
QUERY KeyID

The first format specifies adding a user. USER1 represents the sponsor for the new user while USER2 represents the new user to be added. Both users must pick new secret keys, which are secretkey1 and secretkey2, respectively. KeyID will be a string of 1 to 20 uppercase characters labeling that particular shared key.

The second format describes deleting USER1. In delete cases where the sibling node of the deleted node is ALSO a leaf node, the sibling user must choose a new secretkey. This is secretkey2. If the sibling node is NOT a leaf node, no new key needs to be chosen. Instead, the internal sibling node's parent will get deleted and replaced with this sibling node. (To see an example, consider the picture in the handout. Imagine deleting user M_3. All we would need to do

is delete node K_3 and put K_7 in its place. Then we would have to propagate up K_7 to its ancestors.) A key is still provided for the ease of the file format. If the delete fits into this second case, just ignore the key provided.

The third format makes a query for a particular shared key with the id KeyID. For each of these lines, your program should output on a single line the value of the specified key at that point in time.

You are guaranteed that all queries are for valid keys in the tree, that all add commands list a valid user for USER1 and a new user for USER2. Each different user will have a unique identifying string as will each key. After the first operation, the tree will always have at least two users. (Thus, the second request in the file must either be a query or an add and no deletes can be made until there are at least three users.)

Output Format

For each query, output a single integer on a line by itself satisfying the given query.

Sample Input

```
29 3
21
ALICE 7 BOB 5 K0
QUERY K0
ADD ALICE 3 CAROL 13 K1
QUERY K0
QUERY K1
ADD CAROL 6 DAVID 20 K2
QUERY K0
QUERY K1
QUERY K2
ADD CAROL 2 EARL 17 K3
QUERY K0
QUERY K1
QUERY K2
QUERY K3
DEL CAROL 21
QUERY K0
QUERY K1
QUERY K2
DEL ALICE 23
QUERY K0
QUERY K2
```

Sample Output

```
12
18
15
3
17
7
20
20
24
4
8
27
1
11
1
```

Deliverables

A single source file, GroupDH.java or GroupDH.py that solves the given problem. Make sure your solution reads from standard in and outputs to standard out. ***If you don't do this, I'll take off 20 points no matter what.***