Can X be trusted ?

The National Secrets Exchange, processes 100s of thousands of online requests for secrets every minute. However recently it was noticed that a large number of the requests were by entities that could NOT be trusted. The exchange wants to subject every "Non-Trustable" entity request to further scrutiny.

For the first level of trust analysis, we use an IP Table based approach. We have four inputs.

We shall denote the IP Ranges using the format **a.b.c.d/e** - This corresponds to the IP Address range formed by the first **e** bits of the IP Address **a.b.c.d**, and the rest of the bits allowed to vary. Each of the **a,b,c,d** corresponds to 8bit values.

Example Ranges

1. 10.10.0.0/16 is the range from 10.10.0.0 to 10.10.255.255 since the first 16 bits are fixed, and the last 16 (32-16) bits can vary.
2. 172.0.0.0/8 is the range from 172.0.0.0 to 172.255.255.255 since the first 8 bits are fixed and the last 24 (32-8) bits can vary.

Operations

1. **TRUST a.b.c.d/e** - This denotes that the range denoted by the IP Range is trustable. The server should add the rule, and print 'OK'. If the current range exists, it should be overwritten
2. **NOTRUST a.b.c.d/e** - This denotes that the range denoted by the IP Range is not-trustable. The server should add the rule and print 'OK'. If the current range exists, it should be overwritten
3. **REMOVE a.b.c.d/e** - This should only remove the rule corresponding to the exact range, and not to any sub or super ranges. If the rule exists, remove it and print 'OK', if it does NOT exist, do not do anything and print 'NULL'.
4. **QUERY a.b.c.d** - We need to know whether to trust the following IP or not. **The trustability of an IP Address is defined by the tightest range encompassing this IP Address.**. It should reply either 'TRUST' or 'NOTRUST'
   For example if the given IP is within two ranges in the rule set, the one chosen should be the one with the higher **e** value. The input will be such that there will never be a QUERY that cannot be identified.

The format we use is the same as that used in IP Range system, except that in the standard system, it rarely has **e** value less than 8.

The input will always contain an entry for 0.0.0.0/0, which as can be imagined encompasses the entire set of IP Addresses,it may get reassigned as TRUST or NOTRUST but will never be removed. This ensures that every QUERY would have a valid answer.

Input
The first line of input would be *N*. The number of request that will be given. *N* requests follow, where each request will be in the above format.

Constraints

$1 < N < 25000$

Output
N lines of output each corresponding to a particular input.

Sample Input

```
16
TRUST 0.0.0.0/0
TRUST 10.0.0.0/8
QUERY 8.8.8.8
NOTRUST 8.0.0.0/6
QUERY 8.8.8.8
REMOVE 128.0.0.0/8
TRUST 172.16.0.0/12
NOTRUST 172.0.0.0/8
QUERY 172.8.3.122
QUERY 172.31.1.1
TRUST 172.0.0.0/8
QUERY 172.8.3.122
NOTRUST 172.16.0.0/12
QUERY 172.31.1.1
REMOVE 172.16.0.0/12
QUERY 172.31.1.1
```

Sample Output

```
OK
OK
TRUST
OK
NOTRUST
NULL
OK
OK
NOTRUST
TRUST
OK
TRUST
OK
NOTRUST
OK
TRUST
```

Explanation of tightest range

The width of a range is the number of IP addresses it encompasses
10.0.0.0/8 range encompasses addresses from 10.0.0.0 to 10.255.255.255 that is 24 bits or 2^{24}=16777216 IP Addresses.
The range 10.10.0.0/16 however is a smaller ranger that has IP addresses from 10.10.0.0 to 10.10.255.255, here the width is 16 bits or 2^{16}=65536 IP addresses
The tighter range is the one that encompasses lower number of IP addresses, here 10.10.0.0/16 is hence a tighter range than 10.0.0.0/8. The easiest way to check for it in the problem is to check for a **higher** e value