Genetics

Genetic engineering is fun. The scientists have gathered several DNAs and want to create something new out of those. Each DNAs can be represented as a sequence of bases A, G, T, C. Let DNA[a..b] be the subsequence of DNA starting from index a finishing in b inclusive, and DNA[a..] be the subsequence of DNA starting from index a to the end. Scientist can perform the following operations on the DNAs:

- cross operation – they take DNA1 and DNA2 and numbers k1 and k2. Then two new DNAs are created: DNA3 = DNA1[1..k1]+DNA2[k2+1..] and DNA4 = DNA2[1..k2]+DNA1[k1+1..].
- mutate operation – they take a DNA, number k and one of the bases. Then they replace the base in position k in DNA with that base.
- also they need to know certain characteristics of those DNAs. So they can perform count operation – they take DNA and numbers k1 and k2 (k1 <= k2). This operation should return the number of A, G, T, C bases in DNA[k1..k2].

The initial DNAs are numbered from 1 to n, where n is the amount of those DNAs. The new DNAs created in the cross operations are numbered with consecutive integers. You are to write a program to help scientists perform those operations.

Input

The first line of input file contains number n – the amount of initial DNAs. Each of the following n lines contains the description of each DNA. The next line contains number q – the amount of operations to perform. The next q lines contain the description of each operation in the following form:

- CROSS id1 id2 k1 k2
- MUTATE id k m
- COUNT id k1 k2

Constraints

1 <= n <= 20
1 <= q <= 30000
The length of each initial DNA does not exceed 30000. The length of any DNA formed in cross operation won't exceed 2000000000. Total amount of different DNAs won't exceed 10000. It is guaranteed that all operations are correct.

Output

For each count operation print four integers: the number of each of the bases in the given subsequence of the given DNA.

Example

Input:

```
2
CTCGC
TGCGG
5
MUTATE 1 2 A
COUNT 2 2 4
MUTATE 2 1 G
CROSS 2 1 1 5
COUNT 4 3 6
```

Output:

```
0 2 0 1
0 2 0 2
```