Corporative Network

A very big corporation is developing its corporate network. At the beginning, each of the **N** enterprises of the corporation, numbered from 1 to **N**, organized its own computing and telecommunication center. Soon, for amelioration of the services, the corporation started to collect some enterprises in clusters, each of them served by a single computing and telecommunication center as follows. The corporation chose one of the existing centers **I** (serving the cluster **A**) and one of the enterprises **J** in some other cluster **B** (not necessarily the center) and linked them with a telecommunication line. The length of the line between the enterprises **I** and **J** is |**I** – **J**|(mod 1000). In such a way two old clusters are joined to form a new cluster, served by the center of the old cluster **B**. Unfortunately after each join the sum of the lengths of the lines linking an enterprise to its serving center could be changed and the end users would like to know what is the new length.

Write a program to keep trace of the changes in the organization of the network that is able at each moment to answer the questions of the users.

Input

The first line of the input file will contains only the number **T** of the test cases (1 <= **T** <= 5). Each test will start with the number **N** of enterprises (5<=**N**<=20000). Then some number of lines (no more than 200000) will follow with one of the commands:

E I – asking the length of the path from the enterprise **I** to its serving center at the moment; **I I J** – informing that the serving center **I** is linked to the enterprise **J**. The test case finishes with a line containing the word **O**. There are fewer **I** commands than **N** commands.

Output

The output should contain as many lines as the number of **E** commands in all test cases. Each line must contain a single number – the requested sum of lengths of lines connecting the corresponding enterprise with its serving center.

Example

Input:
```
1
4
E 3
I 3 1
E 3
I 1 2
E 3
I 2 4
E 3
O
```

Output:
```
0
2
3
```