# 1411 – Rip Van Winkle's Code

Rip Van Winkle was fed up with everything except programming. One day he found a problem which required to perform three types of update operations **(A, B, C)**, and one query operation **S** over an array **data[]**. Initially all elements of **data[]** are equal to 0. Though Rip Van Winkle is going to sleep for 20 years, and his code is also super slow, you need to perform the same update operations and output the result for the query operation **S** in an efficient way.

```
long long data[250001];
void A( int st, int nd ) {
    for( int i = st; i <= nd; i++ ) data[i] = data[i] + (i - st + 1);
}
void B( int st, int nd ) {
    for( int i = st; i <= nd; i++ ) data[i] = data[i] + (nd - i + 1);
}
void C( int st, int nd, int x ) {
    for( int i = st; i <= nd; i++ ) data[i] = x;
}
long long S( int st, int nd ) {
    long long res = 0;
    for( int i = st; i <= nd; i++ ) res += data[i];
    return res;
}
```

## Input

Input starts with an integer **T (≤ 5)**, denoting the number of test cases.

Each case starts with a line containing an integer **N (1 ≤ N ≤ $10^5$)** denoting the number of operations. Each of the next **N** lines starts with a character **('A', 'B', 'C' or 'S')**, which indicates the type of operation. Character **'A', 'B'** or **'S'** will be followed by two integers, **st** and **nd** in the same line. Character **'C'** is followed by three integers, **st, nd** and **x**. It's assumed that, **1 ≤ st ≤ nd ≤ 250000** and **$-10^5$ ≤ x ≤ $10^5$**. The meanings of these integers are explained in Winkle's code.

## Output

For each case, print the case number first. Then for each operation **'S'**, print **S( st, nd )** in a line.

| Sample Input | Output for Sample Input |
|---|---|
| 2<br>5<br>A 1 6<br>B 3 5<br>S 1 6<br>C 6 10 -2<br>S 1 6<br>1<br>S 1 10 | Case 1:<br>27<br>19<br>Case 2:<br>0 |

## Note

Dataset is huge, use faster I/O methods.