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

Input

The first line of input will contain T ($\leq 4 * 10^5$) denoting the number of operations. Each of the next T 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, $1 \leq st \leq nd \leq 250000$ and $-10^5 \leq x \leq 10^5$. The meanings of these integers are explained by the code of Rip Van Winkle.

Output

For each line starting with the character ‘`S`’, print $S(st, nd)$ as defined in the code.

Sample Input

```
7
A 1 4
B 2 3
S 1 3
C 3 4 -2
S 2 4
B 1 3
S 2 4
```

Sample Output

```
9
0
3
```