

## Problem E

### Landscape Generator

Time limit: 4 seconds

Interactive Creative Players Collective (ICPC) is working on a new computer game for which they want to generate realistic landscapes. One of the ICPC engineers proposed an algorithm inspired by geological processes. The algorithm starts with a flat landscape and repeatedly modifies it by lifting or lowering continuous blocks, thus forming *horsts* (lifted blocks) and *grabens* (lowered blocks). The blocks to be lifted or lowered are selected at random. ICPC hopes to obtain realistic landscapes this way.

Your task is to interpret any sequence of such modifications and output the resulting landscape. The landscape is represented by a sequence of  $n$  integer height values, one for each integer point from 1 to  $n$  on the  $x$ -axis. Figure E.1 illustrates an example by connecting the height values with line segments.

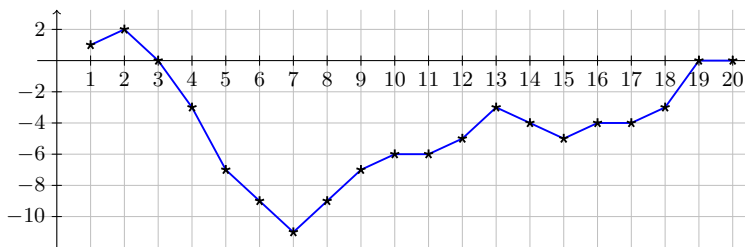


Figure E.1: Illustration of the landscape generated by Sample Input 1.

Initially the height is 0 at all  $n$  points. This flat shape is subjected to a sequence of modifications. Each modification applies one of the following four operations with two integer parameters  $x_1 \leq x_2$ :

- R: Raise – increase the height by 1 at all points between  $x_1$  and  $x_2$  inclusive.
- D: Depress – decrease the height by 1 at all points between  $x_1$  and  $x_2$  inclusive.
- H: Hill – add a new linearly shaped hill between  $x_1$  and  $x_2$ .
- V: Valley – add a new linearly shaped valley between  $x_1$  and  $x_2$ .

Adding a hill to the current landscape works as follows. The heights at points  $x_1$  and  $x_2$  are increased by 1. If  $x_2 - x_1 > 1$ , the heights at points  $x_1 + 1$  and  $x_2 - 1$  are increased by 2. If  $x_2 - x_1 > 3$ , the heights at points  $x_1 + 2$  and  $x_2 - 2$  are increased by 3, and so on. Figure E.2 shows an example. Adding a valley works in the same way except the heights are decreased instead. The maximal change of height happens in the middle between  $x_1$  and  $x_2$ . If  $x_2 - x_1$  is odd, there will be two neighboring points with maximal change, otherwise just one.

### Input

The first line of input contains two integers  $n$  and  $k$ , where  $n$  ( $1 \leq n \leq 200\,000$ ) is the number of points, and  $k$  ( $0 \leq k \leq 200\,000$ ) is the number of modifications. The  $n$  points along the  $x$ -axis are numbered from 1 to  $n$ . The next  $k$  lines describe the modifications. Each line contains one character  $c$  and two integers  $x_1$  and  $x_2$ , where  $c$  (one of R, D, H or V) designates the operation and  $x_1$  and  $x_2$  ( $1 \leq x_1 \leq x_2 \leq n$ ) specify its parameters.

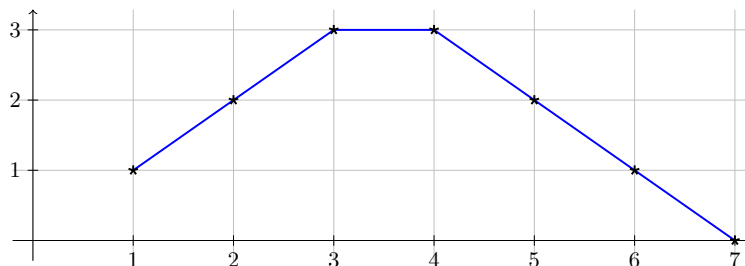


Figure E.2: Illustration of the landscape generated by Sample Input 2.

## Output

Output  $n$  lines, where the  $i^{\text{th}}$  line contains the height at point  $i$  after applying all modifications in the given order.

### Sample Input 1

```
20 13
H 12 13
D 5 18
R 13 14
R 8 16
H 2 3
V 10 19
V 3 13
R 8 13
V 3 10
D 5 18
V 11 12
R 1 6
R 14 19
```

### Sample Output 1

```
1
2
0
-3
-7
-9
-11
-9
-7
-6
-6
-5
-3
-4
-5
-4
-4
-3
0
0
```

### Sample Input 2

```
7 1
H 1 6
```

### Sample Output 2

```
1
2
3
3
2
1
0
```