

# Data Race

## Problem ID: datarace

You have  $T$  threads that are writing to a circular data structure with  $N$  positions (each position storing a single integer), and all of the threads write 1 integer per second. Each thread  $i$  starts writing at time  $t_i$ , from position  $p_i$  in the data structure, and it writes  $v_i$  occurrences of an integer  $a_i$  to the circular data structure. The thread will write to each position sequentially (starting from  $p_i$ , and then writing to  $(p_i + 1)\%N$ ,  $(p_i + 2)\%N$ , and so on). If it reaches index  $N - 1$ , the next index will be 0 since the data structure is circular. If two or more threads try to write different integers at the same time to the same position, a race condition occurs, which means the value at that spot is indeterminate, because it could be one of many values depending on which thread succeeded the write operation. However, if the threads all write the value  $v$  to the same position at the same time, the value at that position will be  $v$  since there is no ambiguity.

For example, in the first sample case, the end result of the data structure is  $\{1, 0, 1, 1, 0\}$ . Even though threads 2 and 3 write to the same spots at the same time, this does not cause a race condition since they write the same integer. On the other hand, in the second sample case, the end result of the data structure is  $\{1, 0, ?, ?, 0\}$  where  $?$  represents an indeterminate character, since two threads write different characters at the same time to the same position.

The values that the threads can write are in the range  $[0, A)$ . At the end of all the thread writes, for each integer  $i \in [0, A)$ , there are  $k_i$  occurrences of  $i$ . Given that every position in the data structure was initialized to 0 before any threads started writing, print all  $k_i$  in order, and at the end, print the number of indeterminate characters.

### Input

The first line of the input contains three space-separated integers,  $T$  ( $1 \leq T \leq 50\,000$ ),  $N$  ( $1 \leq N \leq 10^9$ ),  $A$  ( $1 \leq A \leq 100\,000$ ). These represent the number of threads, the size of the circular data structure, and the number of integers that the threads can write to the data structure respectively.

Another  $T$  lines follow with four space-separated integers  $t_i$  ( $0 \leq t_i \leq 10^9$ ),  $p_i$  ( $0 \leq p_i < N$ ),  $v_i$  ( $1 \leq v_i \leq 10^9$ ),  $a_i$  ( $0 \leq a_i < A$ ). These represent the time that thread  $i$  starts writing, the position it starts writing from, the number of values it writes, and the number it writes to each position. Each thread writes 1 integer per second.

### Output

Output  $A + 1$  space-separated numbers  $k_0 k_1 \dots k_{A-1} k_?$  where  $k_i$  represents the amount of times the number  $i$  appears in the data structure after all threads finish writing for  $0 \leq i < A$ .  $k_?$  represents the amount of times an indeterminate character appears in the data structure after all threads finish writing.

#### Sample Input 1

```
3 5 2
0 0 1 1
1 2 2 1
1 2 2 1
```

#### Sample Output 1

```
2 3 0
```

#### Sample Input 2

```
3 5 2
0 0 1 1
1 2 2 0
1 2 2 1
```

#### Sample Output 2

```
2 1 2
```