# Arithmetic Progressions

Let $F(a, d)$ denote an arithmetic progression (AP) with first term $a$ and common difference $d$, i.e. $F(a, d)$ denotes an infinite $AP => a, a + d, a + 2d, a + 3d, \ldots$. You are given $n$ APs => $F(a_1, d_1), F(a_2, d_2), F(a_3, d_3), \ldots F(a_n, d_n)$. Let $G(a_1, a_2, \cdots a_n, d_1, d_2, \cdots d_n)$ denote the sequence obtained by multiplying these APs.

Multiplication of two sequences is defined as follows. Let the terms of the first sequence be $A_1, A_2, \cdots A_m$, and terms of the second sequence be $B_1, B_2, \cdots B_m$. The sequence obtained by multiplying these two sequences is

$$A_1 \times B_1, A_2 \times B_2, \cdots A_m \times B_m$$

If $A_1, A_2, \cdots A_m$ are the terms of a sequence, then the terms of the first difference of this sequence are given by $A'_1, A'_2, \cdots, A'_{m-1}$ calculated as $A_2 - A_1, A_3 - A_2, \cdots A_m - A_{(m-1)}$ respectively. Similarly, the second difference is given by $A'_2 - A'_1, A'_3 - A'_2, A'_{m-1} - A'_{m-2}$, and so on.

We say that the $k^{th}$ difference of a sequence is a constant if all the terms of the $k^{th}$ difference are equal.

Let $F'(a, d, p)$ be a sequence defined as => $a^p, (a + d)^p, (a + 2d)^p, \cdots$
Similarly, $G'(a_1, a_2, \cdots a_n, d_1, d_2, \cdots d_n, p_1, p_2, \cdots p_n)$ is defined as => product of $F'(a_1, d_1, p_1), F'(a_2, d_2, p_2), \cdots, F'(a_n, d_n, p_n)$.

**Task:**
Can you find the smallest $k$ for which the $k^{th}$ difference of the sequence $G'$ is a constant? You are also required to find this constant value.

You will be given many operations. Each operation is of one of the two forms:

1) `0 i j` => 0 indicates a query $(1 \le i \le j \le n)$. You are required to find the smallest $k$ for which the $k^{th}$ difference of $G'(a_i, a_{i+1}, \ldots a_j, d_i, d_{i+1}, \cdots d_j, p_i, p_{i+1}, \cdots p_j)$ is a constant. You should also output this constant value.

2) `1 i j v` => 1 indicates an update $(1 \le i \le j \le n)$. For all $i \le k \le j$, we update $p_k = p_k + v$.

**Input Format**
The first line of input contains a single integer $n$, denoting the number of APs.
Each of the next $n$ lines consists of three integers $a_i, d_i, p_i$ $(1 \le i \le n)$.
The next line consists of a single integer $q$, denoting the number of operations. Each of the next $q$ lines consist of one of the two operations mentioned above.

**Output Format**
For each query, output a single line containing two space-separated integers $K$ and $V$. $K$ is the smallest value for which the $K^{th}$ difference of the required sequence is a constant. $V$ is the value of this constant. Since $V$ might be large, output the value of $V$ modulo 1000003.

Note: $K$ will always be such that it fits into a signed 64-bit integer. All indices for query and update are 1-based. Do not take modulo 1000003 for $K$.

**Constraints**
$1 \le n \le 10^5$
$1 \le a_i, d_i, p_i \le 10^4$
$1 \le q \le 10^5$
For updates of the form `1 i j v`, $1 \le v \le 10^4$

### Sample Input

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

### Sample Output

```
2 12
2 8
```

### Explanation

The first sequence given in the input is => $1, 3, 5, 7, 9, \ldots$
The second sequence given in the input is => $5, 8, 11, 14, 17, \ldots$

For the first query operation, we have to consider the product of these two sequences:
=> $1 \times 5, 3 \times 8, 5 \times 11, 7 \times 14, 9 \times 17, \ldots$
=> $5, 24, 55, 98, 153, \ldots$
First difference is => $19, 31, 43, 55, \ldots$
Second difference is => $12, 12, 12, \ldots$ This is a constant and hence the output is `2 12`.

After the update operation `1 1 1 1`, the first sequence becomes => $1^2, 3^2, 5^2, 7^2, 9^2, \ldots$
i.e => $1, 9, 25, 49, 81, \ldots$
For the second query, we consider only the first sequence => $1, 9, 25, 49, 81, \ldots$
First difference is => $8, 16, 24, 32, \ldots$
Second difference is => $8, 8, 8, \ldots$ This is a constant and hence the output is `2 8`