



Arithmetic Progressions

by HackerRank

Problem

Submissions

Leaderboard

Discussions

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 $AP \Rightarrow a, a + d, a + 2d, a + 3d, \dots$. You are given n APs $\Rightarrow F(a_1, d_1), F(a_2, d_2), F(a_3, d_3), \dots, F(a_n, d_n)$. Let $G(a_1, a_2, \dots, a_n, d_1, d_2, \dots, d_n)$ 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, \dots, A_m , and terms of the second sequence be B_1, B_2, \dots, B_m . The sequence obtained by multiplying these two sequences is

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

If A_1, A_2, \dots, A_m are the terms of a sequence, then the terms of the first difference of this sequence are given by $A'_1, A'_2, \dots, A'_{m-1}$ calculated as $A_2 - A_1, A_3 - A_2, \dots, A_m - A_{m-1}$ 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 $\Rightarrow a^p, (a + d)^p, (a + 2d)^p, \dots$

Similarly, $G'(a_1, a_2, \dots, a_n, d_1, d_2, \dots, d_n, p_1, p_2, \dots, p_n)$ is defined as \Rightarrow product of $F'(a_1, d_1, p_1), F'(a_2, d_2, p_2), \dots, F'(a_n, d_n, p_n)$.

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:

- 0 i j \Rightarrow 0 indicates a query ($1 \leq i \leq j \leq n$). You are required to find the smallest k for which the k^{th} difference of $G'(a_i, a_{i+1}, \dots, a_j, d_i, d_{i+1}, \dots, d_j, p_i, p_{i+1}, \dots, p_j)$ is a constant. You should also output this constant value.
- 1 i j v \Rightarrow 1 indicates an update ($1 \leq i \leq j \leq n$). For all $i \leq k \leq j$, 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 ($1 \leq i \leq n$).

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

$$1 \leq n \leq 10^5$$

$$1 \leq a_i, d_i, p_i \leq 10^4$$

$$1 \leq q \leq 10^5$$

For updates of the form 1 i j v, $1 \leq v \leq 10^4$

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, ...**

The second sequence given in the input is => **5, 8, 11, 14, 17, ...**

For the first query operation, we have to consider the product of these two sequences:

=> **1 × 5, 3 × 8, 5 × 11, 7 × 14, 9 × 17, ...**

=> **5, 24, 55, 98, 153, ...**

First difference is => **19, 31, 43, 55, ...**

Second difference is => **12, 12, 12, ...** This is a constant and hence the output is 2 12.

After the update operation 1 1 1 1, the first sequence becomes => **1², 3², 5², 7², 9², ...**

i.e => **1, 9, 25, 49, 81, ...**

For the second query, we consider only the first sequence => **1, 9, 25, 49, 81, ...**

First difference is => **8, 16, 24, 32, ...**

Second difference is => **8, 8, 8, ...** This is a constant and hence the output is 2 8

[f](#) [t](#) [in](#)

Submissions: [595](#)



Max Score: 80

Difficulty: Advanced

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

Java 7



```

1 import java.io.*;
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11     }
12 }

```

Line: 1 Col: 1

 [Upload Code as File](#)

☐ Test against custom input

[Run Code](#)

[Submit Code](#)

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)