

# Dora and C++

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Dora has just learned the programming language C++!

However, she has completely misunderstood the meaning of C++. She considers it as two kinds of adding operations on the array  $c$  with  $n$  elements. Dora has two integers  $a$  and  $b$ . In one operation, she can choose one of the following things to do.

- Choose an integer  $i$  such that  $1 \leq i \leq n$ , and increase  $c_i$  by  $a$ .
- Choose an integer  $i$  such that  $1 \leq i \leq n$ , and increase  $c_i$  by  $b$ .

Note that  $a$  and  $b$  are **constants**, and they can be the same.

Let's define a *range* of array  $d$  as  $\max(d_i) - \min(d_i)$ . For instance, the range of the array  $[1, 2, 3, 4]$  is  $4 - 1 = 3$ , the range of the array  $[5, 2, 8, 2, 2, 1]$  is  $8 - 1 = 7$ , and the range of the array  $[3, 3, 3]$  is  $3 - 3 = 0$ .

After any number of operations (possibly, 0), Dora calculates the range of the new array. You need to help Dora minimize this value, but since Dora loves exploring all by herself, you only need to tell her the minimized value.

## Input

Each test consists of multiple test cases. The first line contains a single integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases. The description of test cases follows.

The first line of each test case contains three integers  $n$ ,  $a$ , and  $b$  ( $1 \leq n \leq 10^5$ ,  $1 \leq a, b \leq 10^9$ ) — the length of the array  $c$  and the constant values, respectively.

The second line of each test case contains  $n$  integers  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 10^9$ ) — the initial elements of the array  $c$ .

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^5$ .

## Output

For each test case, output a single integer — the minimum possible range of the array after any number of operations.

## Example

| standard input                                 | standard output    |
|--|--------------------|
| 10   | 3                  |
| 4 5 5  | 0                  |
| 1 3 4 4  | 3                  |
| 4 2 3  | 2                  |
| 1 3 4 6  | 3                  |
| 4 7 7  | 5                  |
| 1 1 2 6  | 1                  |
| 3 15 9   | 0                  |
| 1 9 5  | 17                 |
| 3 18 12  | 205359241          |
| 1 4 5  |                    |
| 7 27 36  |                    |
| 33 13 23 12 35 24 41                           |                    |
| 10 6 9   |                    |
| 15 5 6 9 8 2 12 15 3 8                         |                    |
| 2 1 1000000000                                 |                    |
| 1 1000000000                                   |                    |
| 6 336718728 709848696                          |                    |
| 552806726 474775724 15129785 371139304         | 178408298 13106071 |
| 6 335734893 671469786                          |                    |
| 138885253 70095920 456876775 9345665 214704906 | 375508929          |

## Note

In the first test case, we can increase  $c_1 = 1$  by  $a = 5$ . The array  $c$  will become  $[6, 3, 4, 4]$ , and the range is 3. Note that there is more than one way to reach the answer.

In the second test case, we can increase  $c_1 = 1$  by  $a = 2$  and then increase  $c_1 = 3$  by  $b = 3$ . Also, we can increase  $c_2 = 3$  by  $b = 3$  and increase  $c_3 = 4$  by  $a = 2$ . The array  $c$  will become  $[6, 6, 6, 6]$ , and the range is 0.