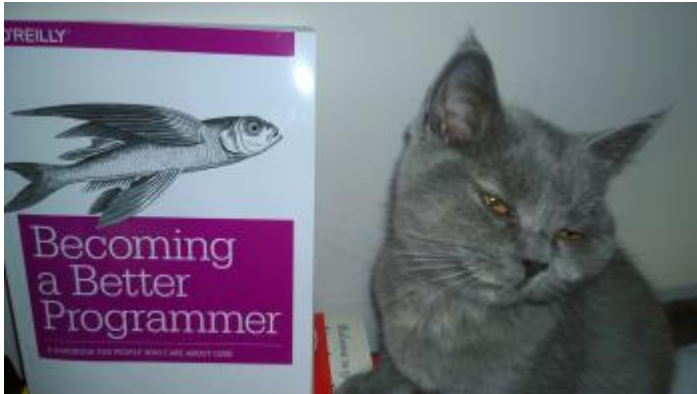# Problem 2 – Sum of Differences

Sid is a small and smart kitten.

She loves to eat programming books (and read them).

When Sid reads books she remembers some of the terms and then combines them and creates programming tasks.

In this case Sid combined "*Absolute difference*", "*Even jump*", "*Odd jump*" into a problem called "*Sum of Differences*". It is the problem you are reading right now. Solve the problem that this kitten created and disprove her allegation that you are a bad programmer. ☺

Absolute difference between two integer numbers **A** and **B** is the difference of the larger between **A** and **B** minus the smaller between **A** and **B**. Examples:

- Absolute difference between 5 and 1 is 4 (5 − 1 = 4).
- Absolute difference between -2 and -2 is 0 (-2 − -2 = 0).
- Absolute difference between 1 and 4 is 3 (4 − 1 = 3).

Even jump in a sequence of numbers is moving 2 positions right in the sequence.

Odd jump in a sequence of numbers is moving 1 position right in the sequence.

Implement the following algorithm on a zero-indexed sequence of numbers:

1. Start from the number with index 1 (second number).
2. Find the absolute difference between the current number and the previous one (with index minus one).
   - o If the absolute difference is even number, then make *even jump*.
   - o If the absolute difference is odd number, then make *odd jump*.
3. Continue with step 2, until you reach the end of the sequence.
4. **Sum all odd absolute differences** you calculate during the algorithm and output them.

Example

Let's follow the algorithm in the sequence "1 6 8 10 3 1 1".

We start from the second number (6) and find the absolute difference between 1 and 6 which is **5**.

5 is odd number so we do odd jump (move 1 position right) (to the number 8). The absolute difference between 8 and 6 is **2**.

2 is even number so we do even jump (move 2 positions right) (to the number 3). The absolute difference between 3 and 10 is **7**.

7 is odd number so we do odd jump (move 1 position right) (to the number 1). The absolute difference between 1 and 3 is **2**.

2 is even, but if we do even jump (move 2 positions right) we will reach out of the sequence, so our algorithm stops here.

$1^5 6^2 8^- 10^7 3^2 1^- 1$

We have found 4 absolute differences during the algorithm (**5**, **2**, **7** and **2**). Only two of them are odd (5 and 7) so we sum them and output the number **12**.

You are given sequences of numbers. **Implement the given algorithm and output the sum of all odd absolute differences found during the algorithm.**

Input

The input data should be read from the console.

The sequence will be given on the only input line. All numbers will be separated by a single space (' ').

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data should be printed on the console.

On the only output line write the sum of all odd absolute differences found during the algorithm.

Constraints

- There will be between 2 and 200 numbers in the input, inclusive.
- Each number will be between -2 000 000 000 and +2 000 000 000.
- Allowed working time for your program: 0.1 seconds.
- Allowed memory: 16 MB.

Examples

Example input	Example output	Explanation
1 6 8 10 3 1 1	12	$1\ {}^5 6\ {}^2 8\ {}^{-}10\ {}^7 3\ {}^2 1\ {}^{-}1$ **5**+**7**=**12**
-5 5 1 8 -4 1 2	8	$-5\ {}^{10} 5\ {}^{-}1\ {}^7 8\ {}^{12} -4\ {}^{-}1\ {}^1 2$ **7**+**1**=**8**

Mew.