# Problem 2. Pokemon Don’t Go

Ely likes to play Pokemon Go a lot. But Pokemon Go bankrupted … So the developers made Pokemon Don’t Go out of depression. And so Ely now plays Pokemon Don’t Go. In Pokemon Don’t Go, when you walk to a certain pokemon, those closer to you, naturally get further, and those further from you, get closer.

You will receive a **sequence** of **integers**, separated by **spaces** – the **distances** to the pokemons.  
Then you will begin **receiving integers**, which will **correspond** to **indexes** in **that** **sequence**.

When you **receive** an **index**, you must **remove** the **element** at **that index** from the **sequence** (as if you’ve captured the pokemon).

* You must **INCREASE** the **value** of **all elements** in the sequence which are **LESS** or **EQUAL** to the **removed element**, with the **value** of the **removed element**.
* You must **DECREASE** the **value** of **all elements** in the sequence which are **GREATER** than the **removed element**, with the **value** of the **removed element**.

If the **given index** is **LESS** than **0**, **remove** the **first element** of the **sequence**, and **COPY** the **last element** to its place.

If the **given index** is **GREATER** than the **last index** of the **sequence**, **remove** the **last element** from the sequence, and **COPY** the **first element** to its place.

The **increasing** and **decreasing** of elements should be done in these cases, **also**. The **element**, whose value you should use is the **REMOVED** element.

The program **ends** when the **sequence** has **no elements** (there are no pokemons left for Ely to catch).

### Input

* On the **first line** of input you will receive a **sequence** of **integers**, **separated** by **spaces**.
* On the **next several** lines you will receive **integers** – the **indexes**.

### Output

* When the program ends, you must print on the console, the **summed up** **value** of **all REMOVED elements**.

### Constrains

* The input data will consist **ONLY** of **valid integers** in the **range [-2.147.483.648, 2.147.483.647]**.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 4 5 3  1  1  0 | 14 | The **array** is {4, 5, 3}. The index is 1.  We **remove** 5, and we **increase all** **lower** than it and **decrease all higher** than it.  In this case there are **no higher** than 5.  The result is {9, 8}.  The **index** is 1. So we remove 8, and **decrease all higher** than it.  The result is {1}.  The **index** is 0. So we remove 1.  There are **no elements** **left**, so we print the **sum** of **all removed elements**.  5 + 8 + 1 = 14. |
| 5 10 6 3 5  2  4  1  1  3  0  0 | 51 | **Step 1**: {11, 4, 9, 11}  **Step 2**: {22, 15, 20, 22}  **Step 3**: {7, 5, 7}  **Step 4**: {2, 2}  **Step 5**: {4, 4}  **Step 6**: {8}  **Step 7**: {} **(empty).**  **Result** = 6 + 11 + 15 + 5 + 2 + 4 + 8 = 51. |

## C:\Users\inter\AppData\Local\Microsoft\Windows\INetCache\Content.Word\maxresdefault.jpg