# Problem 2. Icarus

Icarus is the majestic phoenix who has been alive from the beginning of creation. Icarus travels through different planes. When Icarus travels through a plane, he damages Reality itself with his overwhelming, beyond godlike flames.

You will receive a **sequence** of **integers** – the **plane**. After that you will receive **1** **integer** – an **index** in that **sequence**, which is Icarus’s **starting position**. Icarus’s **INITIAL DAMAGE** is **1**.

You will then begin **receiving** **commands** in the following format: “{direction} {steps}”. The direction will be either “left” or “right”, and the **steps** will be an **integer**. Depending on the direction, Icarus must **step** through the sequence of **integers to the left** or **right**. Each time he **steps** on a **NEW** **position**, he **damages** it. In other words, he **SUBTRACTS** his **current damage** **from** the **integer** at **that** **position**. Walking left and right has its conditions though:

* If Icarus **passes beyond** the **start** of the **sequence** (index: -1) while going **left**, he must go at the **end** of the **sequence** (index: length – 1).
* If Icarus **passes beyond** the **end** of the **sequence** (index: length - 1) while going **right**, he must go at the **start** of the **sequence** (index: 0).

If **1** of the **2 cases** **stated above** happens, Icarus **increments** his **damage** by **1**.

The input ends when you receive the command “Supernova”. When that happens you must print what is **left** of the **sequence**.

### Input

* On the **first input line** you will get the **sequence** of **integers**, **separated** by **spaces**.
* On the **second input line** you will get Icarus’s **starting position**.
* On the **next several input lines** you will get the **commands**.

### Output

* As output you must print a **single line** containing the **remaining elements** of the **sequence**, **separated** by **spaces**.

### Constrains

* The **integers** in the **sequence** will be in **range [0, 1000]**.
* The **initial position** of Icarus will **always** be **valid** and **inside** the **sequence’s indexes**.
* The **direction** will always be either “left” or “right”.
* The **steps** will be in **range [0, 1000]**.
* There will be **NO invalid** input lines.
* Allowed working time / memory: **100ms / 16MB**.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 50 50 25 50 50  3  left 2  right 2  left 2  right 2  Supernova | 50 48 21 48 50 | Initial index: 3  Initial state:  50 50 25 50 50  Go left 2 steps:  50 50 24 50 50  50 49 24 50 50  Go right 2 steps:  50 49 23 50 50  50 49 23 49 50  Go left 2 steps:  50 49 22 49 50  50 48 22 49 50  Go right 2 steps:  50 48 21 49 50  50 48 21 48 50  Final state:  50 48 21 48 50 |
| 5 3 5 5 5  2  left 5  left 5  Supernova | 2 0 0 0 0 | Initial index: 2  Initial state:  5 3 5 5 5  Go left 5 steps:  5 2 5 5 5  4 2 5 5 5  4 2 5 5 3  4 2 5 3 3  4 2 3 3 3  Go left 5 steps:  4 0 3 3 3  2 0 3 3 3  2 0 3 3 0  2 0 3 0 0  2 0 0 0 0  Final state:  2 0 0 0 0 |