# Problem 2 - Greenhouse



*Owning a* ***greenhouse*** *must be challenging. You have to deal with many tasks like following the steps necessary for the* ***crops*** *to* ***grow****.*

On the **first line**, you will receive the **crops**, separated by " **& "**.

On the **next lines, until you receive "Collect!"**, you will be receiving **commands** as follows**:**

* **"Plant {crop}":**
  + If the crop **is not** in the collection, **plant it** (**add it**)at the **beginning** of the sequence.
  + Otherwise, ignore the command.
* **"Transplant {crop}":**
  + If the **crop** is **in** the collection, **transplant it** (**move it**) to the **end** of the sequence.
  + Otherwise, ignore the command.
* **"Replace {cropIndex1} {cropIndex2}":**
  + If **both** crop **indexes** **are valid**, **swap** the **crops**' places.
  + Otherwise, ignore the command.
* **"Uproot {crop}"**:
  + If the **crop** is **in** the collection, **uproot** **it** (**remove** **it**).
  + Otherwise,ignorethe command.

In the **end**, upon receiving a command **"Collect!"**, **print** the final state of the **greenhouse** **collection** with each **crop**, **separated** by **" | "**.

## Input / Constraints

* On the **first line**, you will receive the **crops**, separated by " **& "**. There will always be **at least two different crops initially**.
* On the **following lines**, you will receive **commands** **until** the **command "Collect!"**.
* All the **commands** will be **valid**.

## Output

* Print **the crops in the final collection** on a single line, separated by **" | "** (if there is **more than one** **crop**).
* There will always be **at least one crop to print** at the **end**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| tomato & carrot & potato  Plant cucumber  Transplant carrot  Replace 1 3  Collect! | cucumber | carrot | potato | tomato |
| **Comments** | |
| The **initial state** of the sequence is **"tomato", "carrot", "potato"**.  The **first command** is **"Plant cucumber"**: **"cucumber"** is **not** **in the collection**, so it is **added at the beginning** of the sequence.  The **new state** is **"cucumber", "tomato", "carrot", "potato"**.  А **new command** is received **"Transplant carrot"**: **"carrot"** **exists**, so it is **moved to the end** of the sequence.  The **new state** is **"cucumber", "tomato", "potato", "carrot"**.  A **command** **"Replace 1 3"** is received: **both indexes 1 and 3 are valid**, so we **swap the crops** at these positions (index **1** has **"tomato"** and index **3** has **"carrot"**, now swap them).  **New state**: **"cucumber", "carrot", "potato", "tomato"**.  **Command**: **"Collect!"** comes: the **program ends**, and the **final collection is printed**.  The **output** is **"cucumber | carrot | potato | tomato"** | |
| **Input** | **Output** |
| mint & basil & pea & pepper  Plant bean  Replace 3 5  Replace 1 2  Uproot basil  Collect! | bean | mint | pea | pepper |
| tomato & bean & pea  Uproot tomato  Uproot bean  Collect! | pea |

## JS Examples

The input will be provided as an array of strings.

|  |  |
| --- | --- |
| **Input** | **Output** |
| (['tomato & carrot & potato', 'Plant cucumber', 'Transplant carrot', 'Replace 1 3', 'Collect!']) | cucumber | carrot | potato | tomato |
| **Comments** | |
| The **initial state** of the sequence is **"tomato", "carrot", "potato"**.  The **first command** is **"Plant cucumber"**: **"cucumber"** is **not** **in the collection**, so it is **added at the beginning** of the sequence.  The **new state** is **"cucumber", "tomato", "carrot", "potato"**.  А **new command** is received **"Transplant carrot"**: **"carrot"** **exists**, so it is **moved to the end** of the sequence.  The **new state** is **"cucumber", "tomato", "potato", "carrot"**.  A **command** **"Replace 1 3"** is received: **both indexes 1 and 3 are valid**, so we **swap the crops** at these positions (index **1** has **"tomato"** and index **3** has **"carrot"**, now swap them).  **New state**: **"cucumber", "carrot", "potato", "tomato"**.  **Command**: **"Collect!"** comes: the **program ends**, and the **final collection is printed**.  The **output** is **"cucumber | carrot | potato | tomato"** | |
| **Input** | **Output** |
| (['mint & basil & pea & pepper', 'Plant bean', 'Replace 3 5', 'Replace 1 2', 'Uproot basil', 'Collect!']) | bean | mint | pea | pepper |
| (['tomato & bean & pea', 'Uproot tomato', 'Uproot bean', 'Collect!']) | pea |