**Santa’s Gifts**

You will be given an **array of integers**, which represent the **house numbers** you should visit. The **commands** will lead you to them. If they lead you to **non-existing** places, **don’t move**.

* **Forward {numberOfSteps}**
* **Back {numberOfSteps}**
  + When you receive the “**Forward**” or “**Back**” command, you **move the given number of times** in this **direction** and **remove** the house in **this position** from your list. Also, when you receive the next command, you **continue from this position**.
* **Gift {index} {houseNumber}**
  + Enter a **new house number**, which the dwarves have left out on purpose, **at** **the** **given position** and move to its position.
* **Swap {indexOfFirst} {indexOfSecond}**
  + Santa wants to rearrange his path and **swap the order of two houses**. You will receive the **numbers of the houses**, that need to be switched and he doesn’t need to move to fulfill this command.

## Input

* On the first line you will receive the number of commands – **integer in the range [1-50]**
* On the second line you will receive the array of integers, that represent the houses, split by a single space – valid **integers in the range [1 – 500]**
* On the next n lines, you will receive the commands in the **following format**:
  + **Forward {steps}**
  + **Back {steps}**
  + **Gift {index} {value}**
  + **Swap {value1} {value2}**

## Output

* Print the **last position** and the **remaining houses** in the following format:

“Position {position}”

“{houseNumber}, {houseNumber}………, {houseNumber}”

## Constraints

* The house numbers will be valid integers in the range [1 - 1000]
* The number of commands will be a valid integer in the range [1 - 50]
* The commands will be given in the exact format as they are written above
* There will always be at least one valid command

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| [  '5',  '255 500 54 78 98 24 30 47 69 58',  'Forward 1', 0->1  'Swap 54 47', 1  'Gift 1 20', 1  'Back 1', 0  'Forward 3' 3  ] | Position: 3  20, 47, 78, 24, 30, 54, 69, 58 | First, we receive the “Forward” command, the sleigh will start from the beginning – index 0. He has to move 1 step, so he will move to index 1 and delete the house number, which is stored there – 500. What is left of the list:  255 54 78 98 24 30 47 69 58  and Santa’s position is 1.  The next command is “Swap”. After it, the list looks like this:  255 47 78 98 24 30 54 69 58 and Santa’s position doesn’t change.  The “Gift” command has to insert at index 1 the house with number 20:  255 20 47 78 98 24 30 54 69 58 and move Santa to current index – 1.  The “Back” command has to move Santa back 1 step from his current position. He is at 1 position, so he has to move back to position 0, and remove the house number, which it stores:  20 47 78 98 24 30 54 69 58  The last “Forward” command will move him three steps forward from his current position, which is 0, so he goes to – 3 and removes the house:  20 47 78 24 30 54 69 58 |
| `6`  50 40 25 63 78 54 66 77 24 87  Forward 4  Back 3  Forward 3  Gift 2 88  Swap 50 87  Forward 1 | Position: 3  87, 25, 88, 54, 77, 24, 50 |  |

function solve(input) {

    let numOfCommands = input.shift();

    let housesArr = input.shift().split(` `).map(Number);

    let currentHouseIndex = 0;

    for (let i = 0; i < input.length; i++) {

        let [command, num1, num2] = input[i].split(` `);

        num1 = Number(num1);

        num2 = Number(num2);

        if (command === `Forward` && currentHouseIndex + num1 <= housesArr.length) {

            currentHouseIndex += num1;

            housesArr.splice(currentHouseIndex, 1);

        } else if (command === `Back` && currentHouseIndex - num1 >= 0) {

            currentHouseIndex -= num1;

            housesArr.splice(currentHouseIndex, 1);

        } else if (command === `Gift`) {

            housesArr.splice(num1, 0, num2);

            currentHouseIndex = num1;

        } else if (command === `Swap`) {

            a = housesArr.indexOf(num1);

            b = housesArr.indexOf(num2);

            housesArr.splice(b, 1, num1);

            housesArr.splice(a, 1, num2);

        }

    }

    console.log(`Position ${currentHouseIndex}`);

    console.log(housesArr.join(` `))

}

solve([

    '5',

    '255 500 54 78 98 24 30 47 69 58',

    'Forward 1',

    'Swap 54 47',

    'Gift 1 20',

    'Back 1',

    'Forward 3'

])