

## Problem 3 - Memory game

Problem for exam preparation for the [Programming Fundamentals Course @SoftUni](#).  
Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/Practice/Index/2517#1>.

Write a program that recreates the **Memory game**.

On the first line, you will **receive a sequence of elements**. Each element in the sequence **will have a twin**. Until the player receives **"end"** from the console, you will receive **strings with two integers** separated by a space, representing **the indexes** of elements in the sequence.

If the player **tries to cheat** and enters **two equal indexes** or indexes which are **out of bounds of the sequence**, you should **add** two matching elements at the middle of the sequence in the following format:

**"-{number of moves until now}a"**

Then print this message on the console:

**"Invalid input! Adding additional elements to the board"**

### Input

- On the **first** line, you will receive a **sequence of elements**
- On the **following** lines, you will receive **integers** until the command **"end"**

### Output

- Every time the player hit **two matching elements**, you should **remove** them from the sequence and **print** on the console the following message:  
**"Congrats! You have found matching elements - \${element}!"**
- If the player hit **two different elements**, you should **print** on the console the following message:  
**"Try again!"**
- If the player hit **all matching elements** before he receives **"end"** from the console, you should **print** on the console the following message:  
**"You have won in {number of moves until now} turns!"**
- If the player receives **"end"** before he hits all matching elements, you should **print** on the console the following message:  
**"Sorry you lose :(**  
**{the current sequence's state}"**

### Constraints

- All elements in the sequence will always have a matching element.

### Examples

| Input               | Output  |
|---------------------|---|
| 1 1 2 2 3 3 4 4 5 5 | Congrats! You have found matching elements - 1! |

| 1 0<br>-1 0<br>1 0<br>1 0<br>1 0<br>end  | Invalid input! Adding additional elements to the board<br>Congrats! You have found matching elements - 2!<br>Congrats! You have found matching elements - 3!<br>Congrats! You have found matching elements - -2a!<br>Sorry you lose :(<br>4 4 5 5 |
|--|---|
| Comment  |   |
| 1)<br>1 0<br>1 1 2 2 3 3 4 4 5 5 → 1 = 1, equal elements, so remove them. Moves: 1<br>2)<br>-1 0<br>-1 is invalid index so we add additional elements<br>2 2 3 3 -2a -2a 4 4 5 5, Moves: 2<br>3)<br>1 0<br>2 2 3 3 -2a -2a 4 4 5 5 → 2 = 2, equal elements, so remove them. Moves: 3<br>4)<br>1 0<br>3 3 -2a -2a 4 4 5 5 → 3 = 3, equal elements, so remove them. Moves: 4<br>5)<br>1 0<br>-2a -2a 4 4 5 5 → -2a = -2a, equal elements, so remove them. Moves: 5<br>6)<br>You receive the end command.<br>There are still elements in the sequence, so the player loses the game.<br>Final state - 4 4 5 5 |   |
| a 2 4 a 2 4<br>0 3<br>0 2<br>0 1<br>0 1<br>end   | Congrats! You have found matching elements - a!<br>Congrats! You have found matching elements - 2!<br>Congrats! You have found matching elements - 4!<br>You have won in 3 turns!   |
| a 2 4 a 2 4<br>4 0<br>0 2<br>0 1<br>0 1<br>end   | Try again!<br>Try again!<br>Try again!<br>Try again!<br>Sorry you lose :(<br>a 2 4 a 2 4  |

## JS Examples

| Input  | Output   |
|--|--|
| [<br>"1 1 2 2 3 3 4 4 5 5",<br>"1 0",<br>"-1 0",<br>"1 0",<br>"1 0",<br>"1 0",<br>"end"] | Congrats! You have found matching elements - 1!<br>Invalid input! Adding additional elements to the board<br>Congrats! You have found matching elements - 2!<br>Congrats! You have found matching elements - 3!<br>Congrats! You have found matching elements - -1a!<br>Sorry you lose :(<br>4 4 5 5 |

| ]   |   |
|---|---|
| Comment   |   |
| <p>1)<br/> 1 0<br/> 1 1 2 2 3 3 4 4 5 5 → 1 = 1, equal elements, so remove them. Moves: 1<br/> 2)<br/> -1 0<br/> -1 is invalid index so we add additional elements<br/> 2 2 3 3 -2a -2a 4 4 5 5, Moves: 2<br/> 3)<br/> 1 0<br/> 2 2 3 3 -2a -2a 4 4 5 5 → 2 = 2, equal elements, so remove them. Moves: 3<br/> 4)<br/> 1 0<br/> 3 3 -2a -2a 4 4 5 5 → 3 = 3, equal elements, so remove them. Moves: 4<br/> 5)<br/> 1 0<br/> -2a -2a 4 4 5 5 → -2a = -2a, equal elements, so remove them. Moves: 5<br/> 6)<br/> You receive the end command.<br/> There are still elements in the sequence, so the player loses the game.<br/> Final state - 4 4 5 5</p> |   |
| [<br>"a 2 4 a 2 4",<br>"0 3",<br>"0 2",<br>"0 1",<br>"0 1",<br>"end"<br>]   | Congrats! You have found matching elements - a!<br>Congrats! You have found matching elements - 2!<br>Congrats! You have found matching elements - 4!<br>You have won in 3 turns! |
| [<br>"a 2 4 a 2 4",<br>"4 0",<br>"0 2",<br>"0 1",<br>"0 1",<br>"end"<br>]   | Try again!<br>Try again!<br>Try again!<br>Try again!<br>Sorry you lose :(<br>a 2 4 a 2 4  |