# Worms&Holes



The first line will give you **a sequence of integers representing worms**. Afterwards, you will be given another **sequence of integers representing holes**.

You have to start with the **last worm** and try to match it with the **first hole**.

* If their **values** are **equal**, the worm fits the hole and can get into it. After that, you should **remove both** of them from their sequences. Otherwise, you should **remove** the **hole** and **decrease** the **value** of the **worm** by **3**.
* If the **worm** value becomes **equal to or below** **0**, **remove it** from the sequence **before** trying to **match** it with the hole.

You need to **stop** **matching** when you have **no more worms or holes**.

### Input / Constraints

* On the **first line**, you will receive the integers, representing the **worm size**, **separated** by a **single space**.
* On the **second line,** you will receive the integers, representing the **hole size**, **separated** by a **single space**.
* All given numbers will be valid integers in the range **[1, 50]**.

### Output

* On the **first** line:
* If there are matches print the following:
  + "**Matches: {matchesCount}"**
* If there are no matches print the following:
* **"There are no matches."**
* On the **second** line print:
* If there are no worms left and **all of them fit a hole**:
  + "**Every worm found a suitable hole!**"
* If there are no worms left but only some of them fit a hole:
  + "**Worms left: none**"
* If there are worms left:
  + "**Worms left: {worm1}, {worm2}, (…),{wormn}**"
* On the **third** line print:
* If there are no holes left:
  + "**Holes left: none**"
* If there are holes left:
  + "**Holes left: {hole1}, {hole2}, (…),{holen}**"

### Examples

|  |  |  |
| --- | --- | --- |
| ****Input**** | ****Output**** | ****Comment**** |
| **9 5 8 13**  **13 8 5 6** | **Matches: 3**  **Worms left: 6**  **Holes left: none** | The first pair is the **first hole** with a value of 13 and the **last worm** with a value of 13, their **values are equal**, so the worm gets into the hole **and we remove values** from the **sequences**. Next, there are **two more matches** (8 = 8) and (5 = 5) you should **remove both of them**, too. But the value of the **next worm is 9** and the value of the **next hole is 6**, (9 > 6) so we **reduce** the **worm’s value** by 3 and **remove** the **hole**. |
| **17 20 25 25 30**  **9 8 7 21 5 4 3 2 1** | **Matches: 1**  **Worms left: 10, 25, 20, 17**  **Holes left: none** |  |
| **9 8 7 6**  **6 7 8 9** | **Matches: 4**  **Every worm found a suitable hole!**  **Holes left: none** |  |
| **10 10 10 10**  **5** | **There are no matches.**  **Worms left: 7, 10, 10, 10**  **Holes left: none** |  |