# Exercise: Arrays

Test your solutions in the **judge system**: <https://alpha.judge.softuni.org/contests/arrays-exercise/5085>

## Equal Arrays

Write a program that:

* Read **two arrays with the same length** from the console
* **Check whether they are identical or not**
  + If the arrays are identical, print: "**Arrays are identical.**"
  + If the arrays are NOT identical, print: "**Arrays are not identical.**"

**Note:** Arrays are identical if their elements are equal.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 10 20 30  10 20 30 | Arrays are identical. |
| 1 2 3 4 5  1 2 4 3 5 | Arrays are not identical. |
| 1  10 | Arrays are not identical. |

## Common Elements

Write a program that:

* Read **two integer arrays with the same length** from the console
* Prints common elements in two arrays, space-separated

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 20 30 40 50  30 10 15 50 | 30 50 |
| 7 8 9 10  1 2 3 7 | 7 |
| 1 2 3 4 5  5 4 3 2 1 | 1 2 3 4 5 |

## Condense Array to Number

Write a program that:

* Read **a sequence of integer numbers** from the console
* **Condense** them by **summing** adjacent couples of elements until a **single integer** is obtained

**Example:** If we have 3 elements {2, 10, 3}. We sum the first two and the second two elements and obtain {2+10, 10+3} = {12, 13}. Then we sum again all adjacent elements and obtain {12+13} = {25}.

### Example

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 2 10 3 | 25 | 2 10 3 🡪 2+10 10+3 🡪 12 13 🡪 12 + 13 🡪 25 |
| 5 0 4 1 2 | 35 | 5 0 4 1 2 🡪 5+0 0+4 4+1 1+2 🡪 5 4 5 3 🡪 5+4 4+5 5+3 🡪 9 9 8 🡪 9+9 9+8 🡪 18 17 🡪 18+17 🡪 35 |
| 1 | 1 | 1 is already condensed to number |

## Magic Sum

### Write a program that:

* Read **a sequence of integer numbers** from the first line of the console
* Read **an integer number (control number)** from the second line of the console
* Print all unique pairs in an array of integers whose **sum is equal to a control number**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 7 6 2 19 23  8 | 1 7  6 2 |
| 14 20 60 13 7 19 8  27 | 14 13  20 7  19 8 |

## Zig-Zag Arrays

Write a program that:

* Creates **two empty integer arrays**
* Reads an **integer number N** from the console
* Reads two **integer numbers** on the next N lines
* Form **two integer arrays** as shown below

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  1 5  9 10  31 81  41 20 | 1 10 31 20  5 9 81 41 |
| 2  80 23  31 19 | 80 19  23 31 |

## Array Rotation

Write a program that:

* Reads **sequence of integer numbers** from the first line of the console
* Reads **an integer number (rotations you have to perform)** from the second line of the console
* Each rotation is when the **first element goes at the end**
* Print the **resulting sequence**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 51 47 32 61 21  2 | 32 61 21 51 47 |
| 32 21 61 1  4 | 32 21 61 1 |
| 2 4 15 31  5 | 4 15 31 2 |

## Top Integers

Write a program that:

* Reads **sequence of integer numbers** from the first line of the console
* Find all the top integers in a sequence
* Top integer is an integer that is **bigger** than all the elements to its right

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 4 3 2 | 4 3 2 |
| 14 24 3 19 15 17 | 24 19 17 |
| 27 9 42 2 13 45 48 | 48 |