

Java 1D Array

An array is a simple data structure used to store a collection of data in a contiguous block of memory. Each element in the collection is accessed using an *index*, and the elements are easy to find because they're stored sequentially in memory.

Because the collection of elements in an array is stored as a big block of data, we typically use arrays when we know exactly how many pieces of data we're going to have. For example, you might use an array to store a list of student ID numbers, or the names of state capitals. To create an array of integers named *myArray* that can hold four integer values, you would write the following code:

```
int[] myArray = new int[4];
```

This sets aside a block of memory that's capable of storing **4** integers. Each integer storage cell is assigned a unique *index* ranging from **0** to one less than the size of the array, and each cell initially contains a **0**. In the case of *myArray*, we can store integers at indices **0**, **1**, **2**, and **3**. Let's say we wanted the last cell to store the number **12**; to do this, we write:

```
myArray[3] = 12;
```

Similarly, we can print the contents of the last cell with the following code:

```
System.out.println(myArray[3]);
```

The code above prints the value stored at index **3** of *myArray*, which is **12** (the value we previously stored there). It's important to note that while Java initializes each cell of an array of integers with a **0**, not all languages do this.

Task

The code in your editor does the following:

1. Reads an integer from stdin and saves it to a variable, n , denoting some number of integers.
2. Reads n integers corresponding to a_0, a_1, \dots, a_{n-1} from stdin and saves each integer a_i to a variable, *val*.
3. Attempts to print each element of an array of integers named *a*.

Write the following code in the unlocked portion of your editor:

1. Create an array, *a*, capable of holding n integers.
2. Modify the code in the loop so that it saves each sequential value to its corresponding location in the array. For example, the first value must be stored in a_0 , the second value must be stored in a_1 , and so on.

Good luck!

Input Format

The first line contains a single integer, n , denoting the size of the array.

Each line i of the n subsequent lines contains a single integer denoting the value of element a_i .

Output Format

You are not responsible for printing any output to stdout. Locked code in the editor loops through array *a* and prints each sequential element on a new line.

Sample Input

```
5
10
20
30
40
50
```

Sample Output

```
10
20
30
40
50
```

Explanation

When we save each integer to its corresponding index in *a*, we get *a* = [10, 20, 30, 40, 50]. The locked code prints each array element on a new line from left to right.