Objective

Today, we're discussing a simple sorting algorithm called Bubble Sort. Check out the Tutorial tab for learning materials and an instructional video!

Consider the following version of Bubble Sort:

```
for (int i = 0; i < n; i++) {
    // Track number of elements swapped du
    int numberOfSwaps = 0;

    for (int j = 0; j < n - 1; j++) {
        // Swap adjacent elements if they
        if (a[j] > a[j + 1]) {
            swap(a[j], a[j + 1]);
            numberOfSwaps++;
        }
    }

    // If no elements were swapped during
    if (numberOfSwaps == 0) {
        break;
    }
}
```

Task

Given an array, a, of size n distinct elements, sort the array in ascending order using the Bubble Sort algorithm above. Once sorted, print the following 3 lines:

1. `Array is sorted in numSwaps swaps.`
   where $numSwaps$ is the number of swaps that took place.
2. `First Element: firstElement`
   where $firstElement$ is the first element in the sorted array.
3. `Last Element: lastElement`
   where $lastElement$ is the last element in the sorted array.

Hint: To complete this challenge, you will need to add a variable that keeps a running tally of all swaps that occur during execution.

Example

$a = [4, 3, 1, 2]$

```
original a: 4 3 1 2
round 1  a: 3 1 2 4 swaps this round: 3
round 2  a: 1 2 3 4 swaps this round: 2
round 3  a: 1 2 3 4 swaps this round: 0
```

In the first round, the 4 is swapped at each of the 3 comparisons, ending in the last position. In the second round, the 3 is swapped at 2 of the 3 comparisons. Finally, in the third round, no swaps are made so the iterations stop. The output is the following:

```
Array is sorted in 5 swaps.
First Element: 1
Last Element: 4
```

Input Format

Change Theme | Java 8

```
public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[] a = new int[n];
        for(int a_i=0; a_i < n; a_i++){
            a[a_i] = in.nextInt();
        }
        // Write Your Code Here
        int numberOfSwaps = 0;
        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n - 1 ; j++) {
                if (a[j] > a[j + 1]) {
                    int previousItem = a[j + 1];
                    a[j + 1] = a[j];
                    a[j] = previousItem;
                    numberOfSwaps++;
                }
            }
        }

        System.out.println("Array is sorted in " +
                           numberOfSwaps  + " swaps." +
                          "\nFirst Element: " + a[0] +
                          "\nLast Element: " + a[n - 1]);
    }
```

Line: 31 Col: 24

Upload Code as File
Test against custom input

Run Code | Submit Code

You have earned 30.00 points!

You are now 1 challenge away from the 4th star for your 30 days of code badge.

86% 21/22

30 Days of Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2

Compiler Message

Success

Input (stdin) Download

```
3
1 2 3
```

Expected Output Download

```
Array is sorted in 0 swaps.
First Element: 1
Last Element: 3
```

Objective

Today, we're discussing a simple sorting algorithm called Bubble Sort. Check out the Tutorial tab for learning materials and an instructional video!

Consider the following version of Bubble Sort:

```
for (int i = 0; i < n; i++) {
    // Track number of elements swapped du
    int numberOfSwaps = 0;

    for (int j = 0; j < n - 1; j++) {
        // Swap adjacent elements if they
        if (a[j] > a[j + 1]) {
            swap(a[j], a[j + 1]);
            numberOfSwaps++;
        }
    }

    // If no elements were swapped during
    if (numberOfSwaps == 0) {
        break;
    }
}
```

Task

Given an array, a, of size n distinct elements, sort the array in ascending order using the Bubble Sort algorithm above. Once sorted, print the following 3 lines:

1. `Array is sorted in numSwaps swaps.`
   where $numSwaps$ is the number of swaps that took place.
2. `First Element: firstElement`
   where $firstElement$ is the first element in the sorted array.
3. `Last Element: lastElement`
   where $lastElement$ is the last element in the sorted array.

Hint: To complete this challenge, you will need to add a variable that keeps a running tally of all swaps that occur during execution.

Example

$a = [4, 3, 1, 2]$

```
original a: 4 3 1 2
round 1  a: 3 1 2 4 swaps this round: 3
round 2  a: 1 2 3 4 swaps this round: 2
round 3  a: 1 2 3 4 swaps this round: 0
```

In the first round, the 4 is swapped at each of the 3 comparisons, ending in the last position. In the second round, the 3 is swapped at 2 of the 3 comparisons. Finally, in the third round, no swaps are made so the iterations stop. The output is the following:

```
Array is sorted in 5 swaps.
First Element: 1
Last Element: 4
```

Input Format

Change Theme Java 8

```
public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int[] a = new int[n];
        for(int a_i=0; a_i < n; a_i++){
            a[a_i] = in.nextInt();
        }
        // Write Your Code Here
        int numberOfSwaps = 0;
        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n - 1 ; j++) {
                if (a[j] > a[j + 1]) {
                    int previousItem = a[j + 1];
                    a[j + 1] = a[j];
                    a[j] = previousItem;
                    numberOfSwaps++;
                }
            }
        }

        System.out.println("Array is sorted in " +
                           numberOfSwaps  + " swaps." +
                          "\nFirst Element: " + a[0] +
                          "\nLast Element: " + a[n - 1]);
    }
```

Line: 31 Col: 24

Upload Code as File
Test against custom input

Run Code Submit Code

You have earned 30.00 points!

You are now 1 challenge away from the 4th star for your 30 days of code badge.

86% 21/22

30 Days of Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2

Compiler Message

Success

Input (stdin) Download

```
3
1 2 3
```

Expected Output Download

```
Array is sorted in 0 swaps.
First Element: 1
Last Element: 3
```