Classes’ distribution

In a given school, the classes of different grades are **placed in order** in N floors. A single floor is completely dedicated to a single grade. A lucky class is the class whose grade is in the same floor number. Note that the offices of employees and teachers are given –ve numbers.

That is... you are given an array of size N, containing the grade number in each floor (numbers are **distinct**). Design an **efficient algorithm** to check if there exists such a lucky class?

If exists, return its grade (NOTE: If more than one lucky class exists, return the grade of 1st one)

else, return -1.

Function to Implement

```
        public static int GetLuckyClass(int N, int[] array)
```

`ClassesDistribution.cs includes this method.`

Examples:

Input: Array = [-5,-2,-1, 2, 4, 6, 7, 9]

Output: 4 (since grade 4 is located in 4th floor)

Floor 0: employees →false
Floor 1: teachers → false
Floor 2: employees → false
Floor 3: grade 2 → false
Floor 4: grade 4→ true
Floor 5: grade 6 → false
Floor 6: grade 7 → false
Floor 7: grade 9 → false

Input: Array = [-4, -1, 1, 3, 4, 5, 7, 8]

Output: 3 (since grade 3 is the FIRST LUCKY CLASS located in 3rd floor)

Floor 0: employees →false
Floor 1: teachers → false
Floor 2: grade 1 → false
Floor 3: grade 3 → true
Floor 4: grade 4→ true
Floor 5: grade 5 → true
Floor 6: grade 7 → false
Floor 7: grade 8 → false
Input: Array =[-3, 6, 7, 9, 10]
Output: -1

C# Help

Creating 1D array

```
int [] array1D = new int [size]
```

Creating 2D array

```
int [,] array2D = new int [size1, size2]
```

Getting the size of 1D array

```
int size = array1D.GetLength(0);
```

Getting the size of 2D array

```
int size1 = array2D.GetLength(0);

int size2 = array2D.GetLength(1);
```

Sorting single array

Sort the given array "items" in ascending order

```
Array.Sort(items);
```

Sorting parallel arrays

Sort the first array "master" and re-order the 2nd array "slave" according to this sorting

```
Array.Sort(master, slave);
```