Given a list of strings g and an integer z, sort the strings by the zth 1-based character ignoring the characters' case.

In case of a tie, the first element that appears in g should go first.

**Example**

For g = ["Cow", "Chicken", "Deer", "Rabbit"] and z = 3,  
the output should be  
sortByZ(g, z) = ["Rabbit", "Deer", "Chicken", "Cow"].

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.string g**

A list of strings.

*Guaranteed constraints:*  
3 ≤ g.length ≤ 50,  
2 ≤ g[i].length < 40.

* **[input] integer z**

Sort the list based on the zth character.

*Guaranteed constraints:*  
 1 ≤ z ≤ length of the shortest string in g.

* **[output] array.string**

The sorted list.

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

string helloWorld(string name) {

Console.Write("This prints to the console when you Run Tests");

return "Hello, " + name;

}

<https://codefights.com/challenge/Z2ZgyaQtu96WDH6Pn/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp6

{

class Program

{

string[] SortByZ(string[] g, int z)

{

for (int i = 0; i < g.Length - 1; i++)

{

for (int j = i + 1; j < g.Length; j++)

{

string x = g[i].ToLower();

string y = g[j].ToLower();

if (z <= x.Length && z <= y.Length)

{

if (x[z - 1] <= y[z - 1])

{

string temp = g[i];

g[i] = g[j];

g[j] = temp;

}

}

}

}

Array.Reverse(g);

return g;

}

static void Main(string[] args)

{

}

}

}