

of every cell should be its index value. Output every cell of the array to the screen, each from the new line, starting from the cell with index 0. For example:

```
>4
0
1
2
3
```

▶

↶

📄

💡

💬

😊

```
3 namespace Arrays
4 {
5     class FillArray
6     {
7         public static void Main(string[] args)
8         {
9             int count = int.Parse(Console.ReadLine());
10            int[] Array = new int[count];
11            for(int i = 0; i < count; i++)
12            {
13                Array[i] = i;
14            }
15            for(int i = 0; i < count; i++)
16            {
17                Console.WriteLine(Array[i]);
18            }
19        }
20    }
21 }
```

0

1

2

3

4

Close

✓ Solved

```
>399.76
1.12
-2.333
399.76
```

▶

↶

📄

💡

💬

😊

```
5 class ArraysCopy
6 {
7     public static void Main(string[] args)
8     {
9         double[] myArray = GetNumbersFromConsole();
10
11         // Write your code here
12         double[] copy = new double[myArray.Length];
13         for (int i = 0; i < myArray.Length; i++)
14         {
15             copy[i] = myArray[i];
16         }
17         for (int i = 0; i < myArray.Length; i++)
18         {
19             Console.WriteLine(copy[i]);
20         }
21     }
22 }
```

1.12

-2.333

399.76

3

1.12

-2.333







399.76

Close

✓ Solved

values from the console into a `double` array. Find the maximum of the values in the array and print "Max is: {number}" to the console. For example:

```
>3
>1.12
>-2.333
>399.76
Max is: 399.76
```







```
11 // Write your code here
12 if (numbers.Length > 0)
13 {
14     double max = numbers[0];
15     for (int i = 1; i < numbers.Length; ++i)
16     {
17         if (numbers[i] > max)
18             max = numbers[i];
19     }
20     Console.WriteLine($"Max is: {max}");
21 }
22
23
24 static double[] GetNumbersFromConsole()
25 {
26     int count = int.Parse(Console.ReadLine());
```

Max is: 399.76

3
1.12
-2.333
399.76

First List

Create a new list of type `int` and name `myNumbers`. Add the following integers to the list using the add method (10, 20, 34, 50). Output the values of every cell in the list to the console, each on a new line.



```
1 using System;
2 using System.Collections.Generic;
3
4 namespace Lists
5 {
6     public class Lists
7     {
8         static void Main(string[] args)
9         {
10             List<int> myNumbers = new List<int>() {10, 20, 34, 50};
11             Console.WriteLine(myNumbers[0]);
12             Console.WriteLine(myNumbers[1]);
13             Console.WriteLine(myNumbers[2]);
14             Console.WriteLine(myNumbers[3]);
15         }
16     }
17 }
```

10
20
34
50







Console input

Close



Solved

Create a new list of type `string` named `countries` with the following strings: "England", "Sweden", "Uzbekistan", "London".
Call the remove method on the list to remove the "London" cell.
Output the values of every cell in the list to the console, each on a new line.



```
1 using System;
2 using System.Collections.Generic;
3
4 namespace Lists
5 {
6     public class ListsRemover
7     {
8         static void Main(string[] args)
9         {
10             List<string> countries = new List<string>() {"England", "Sweden", "Uzbekistan", "London"};
11             countries.Remove("London");
12             Console.WriteLine(countries[0]);
13             Console.WriteLine(countries[1]);
14             Console.WriteLine(countries[2]);
15         }
16     }
17 }
```

England
Sweden
Uzbekistan

Console input







Close



Solved

Removing a single item from a list using the cell index.

Create a new list of type `string` named `programmingLanguages`. Initialize it with the names of these programming languages: "C", "C#", "C++".
Call the `RemoveAt` method on the list to remove the "C" cell.
Output the values of every new cell in the list to the console, each on a new line.



```
1 using System;
2 using System.Collections.Generic;
3
4 namespace Lists
5 {
6     public class ListsRemoverAtIndex
7     {
8         static void Main(string[] args)
9         {
10             List<string> programmingLanguages = new List<string>() {"C", "C#", "C++"};
11             programmingLanguages.RemoveAt(0);
12             Console.WriteLine(programmingLanguages[0]);
13             Console.WriteLine(programmingLanguages[1]);
14         }
15     }
16 }
```

C#
C++

Console input







Close



Solved

Scopes.LoveAndHateEmphasizer

Make the program compile. It should output what you like in UPPERCASE and what you hate in lowercase letters.




```
7 static void Main(string[] args)
8 {
9     string whatILove = Console.ReadLine();
10    string whatIHate = Console.ReadLine();
11
12    string output = CombineString(whatILove, whatIHate);
13
14    Console.WriteLine(output);
15 }
16
17 static string CombineString(string love, string hate)
18 {
19     love = love.ToUpper();
20     hate = hate.ToLower();
21
22     return $"I love {love} but I hate {hate} ";
```

I love KITTENS but I hate kittens' poop.

kittens
kittens' poop

Close

 Solved

