1. Create a program that takes a collection of numbers and returns the sum of the two lowest, positive numbers.  
   Show results for these inputs:

| **Input** |
| --- |
| [19, 5, 42, 2, 77] |
| [3453545353453, 10, 343445353, 3453445] |
| [2, 9, 6, -1] |
| [879, 953, 694, -847, 342, 221, -91, -723, 791, -587] |
| [3683, 2902, 3951, -475, 1617, -2385] |

1. Given a collection of numbers, create a program that removes duplicates and sorts in ascending order.  
   Show results for these inputs:

|  |
| --- |
| **Input** |
| [1, 2, 4, 3] |
| [1, 4, 4, 4, 4, 4, 3, 2, 1, 2] |
| [6, 7, 3, 2, 1] |

1. Consider that you are given a string that contains only parentheses, brackets, and curly braces: (), [], {}. Write a function that takes in this string, determines if the order of the brackets follows the correct scope order, and returns both (1) a Boolean indicating if the order was appropriate and (2) the number of successful pairs. If the order is broken or incomplete, return a value of zero (0) for the number of successful pairs. You may presume that *only* parentheses, brackets, and curly braces will be used in the string.  
   Here are some examples of both valid and invalid strings:

| **Category** | **Input** | **Output** |
| --- | --- | --- |
| Valid | () | True, 1 |
| Valid | ({[]}) | True, 3 |
| Valid | ((){}()) | True, 4 |
| Invalid | ((} | False, 0 |
| Invalid | ()( | False, 0 |