```python
def process_list(lst):
    if not lst:
        return []
    elif len(lst) == 1:
        return lst
    elif len(set(lst)) == 1:
        return lst
    else:
        return sorted(lst)
print(process_list([]))
print(process_list([1]))
print(process_list([7, 7, 7, 7]))
print(process_list([-5, -1, -3, -2, -4]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[]
[1]
[7, 7, 7, 7]
[-5, -4, -3, -2, -1]
```

```python
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
print(selection_sort([5, 2, 9, 1, 5, 6]))
print(selection_sort([10, 8, 6, 4, 2]))
print(selection_sort([1, 2, 3, 4, 5]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[1, 2, 5, 5, 6, 9]
[2, 4, 6, 8, 10]
[1, 2, 3, 4, 5]
```

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        swapped = False
        for j in range(0, n-i-1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
        if not swapped:
            break
    return arr
print(bubble_sort([5, 1, 4, 2, 8]))
print(bubble_sort([1, 2, 3, 4, 5]))
print(bubble_sort([5, 4, 3, 2, 1]))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[1, 2, 4, 5, 8]
[1, 2, 3, 4, 5]
[1, 2, 3, 4, 5]
```

```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and arr[j] > key:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key
    return arr
print(insertion_sort([3, 1, 4, 1, 5, 9, 2, 6, 5, 3]))
print(insertion_sort([5, 5, 5, 5, 5]))
print(insertion_sort([2, 3, 1, 3, 2, 1, 1, 3]))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
[1, 1, 2, 3, 3, 4, 5, 5, 6, 9]
[5, 5, 5, 5, 5]
[1, 1, 1, 2, 2, 3, 3, 3]
```

```python
def findPeakElement(nums):
    left, right = 0, len(nums) - 1
    while left < right:
        mid = (left + right) // 2
        if nums[mid] < nums[mid + 1]:
            left = mid + 1
        else:
            right = mid
    return left
print(findPeakElement([1, 2, 3, 1]))
print(findPeakElement([1, 2, 1, 3, 5, 6, 4]))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
2
5

[Done] exited with code=0 in 0.137 seconds
```

```python
def strStr(haystack, needle):
    if not needle:
        return 0
    haystack_length = len(haystack)
    needle_length = len(needle)
    for i in range(haystack_length - needle_length + 1):
        if haystack[i:i + needle_length] == needle:
            return i
    return -1
print(strStr("sadbutsad", "sad"))
print(strStr("leetcode", "leeto"))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
0
-1

[Done] exited with code=0 in 0.129 seconds
```

```python
def find_substrings(words):
    substrings = set()
    for i in range(len(words)):
        for j in range(len(words)):
            if i != j and words[i] in words[j]:
                substrings.add(words[i])
                break
    return list(substrings)
print(find_substrings(["mass", "as", "hero", "superhero"]))
print(find_substrings(["leetcode", "et", "code"]))
print(find_substrings(["blue", "green", "bu"]))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python"
['as', 'hero']
['et', 'code']
[]
```

```python
def findKthPositive(arr, k):
    missing_count = 0
    current_number = 1
    index = 0
    while missing_count < k:
        if index < len(arr) and arr[index] == current_number:
            index += 1
        else:
            missing_count += 1
            if missing_count == k:
                return current_number
        current_number += 1
print(findKthPositive([2, 3, 4, 7, 11], 5))
print(findKthPositive([1, 2, 3, 4], 2))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[Running] python -u "c:\Users\hp\OneDrive\Desktop\tempCodeRunnerFile.python
9
6

[Done] exited with code=0 in 0.12 seconds
```