main.py

Share Run Output

```
1 empty_list = []
2 print(empty_list)
3
```

```
[]

=== Code Execution Successful ===
```

main.py

Share Run

```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
input_arr = [5, 2, 9, 1, 5, 6]
sorted_arr = selection_sort(input_arr)
print(sorted_arr)

```

Output

```
[1, 2, 5, 5, 6, 9]

=== Code Execution Successful ===
```

main.py

Share

Run

Output

Share code

```python
def bubble_sort_early_termination(arr):
    n = len(arr)
    for i in range(n):
        already_sorted = True
        for j in range(n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                already_sorted = False
        if already_sorted:
            break
    return arr
my_list = [64, 34, 25, 12, 22, 11, 90]
sorted_list = bubble_sort_early_termination(my_list)
print(sorted_list)

```

```
[11, 12, 22, 25, 34, 64, 90]

=== Code Execution Successful ===
```

main.py

Share Run

Output

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
input_list = [64, 25, 12, 22, 11]
sorted_list = bubble_sort(input_list)
print(sorted_list)

```

```
[11, 12, 22, 25, 64]

=== Code Execution Successful ===
```

main.py

Share Run

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

```

Output

```
[1, 1, 2, 3, 3, 4, 5, 5, 6, 9]

=== Code Execution Successful ===
```

main.py

Share Run

Output

```python
def find_kth_missing(arr, k):
    missing_set = set(range(1, arr[-1] + k + 1)) - set(arr)
    return sorted(missing_set)[k - 1]
arr = [2, 3, 4, 7, 11]
k = 5
result = find_kth_missing(arr, k)
print(result)  # Output: 9

```

```
9

=== Code Execution Successful ===
```

main.py

Share Run

Output

```python
def find_peak_element(nums):
    left, right = 0, len(nums) - 1

    while left < right:
        mid = left + (right - left) // 2

        if nums[mid] < nums[mid + 1]:
            left = mid + 1
        else:
            right = mid

    return left

```

```
=== Code Execution Successful ===
```

main.py

Share Run

```python
def find_substrings(words):
    result = []
    for i in range(len(words)):
        for j in range(len(words)):
            if i != j and words[i] in words[j]:
                result.append(words[i])
                break
    return list(set(result))
words = ["mass", "as", "hero", "superhero"]
print(find_substrings(words))

```

Output

```
['hero', 'as']

=== Code Execution Successful ===
```