main.py

Share Run

```python
def sum_of_squares_of_distinct_counts(nums):
    n = len(nums)
    result = 0

    for i in range(n):
        seen = set()
        for j in range(i, n):
            seen.add(nums[j])
            result += len(seen) ** 2

    return result
nums = [1, 2, 1]
print(sum_of_squares_of_distinct_counts(nums))

```

Output

Clear

```
15

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

main.py

Share Run

```python
def unique_elements(nums):
    return list(set(nums))

print(unique_elements([3, 7, 3, 5, 2, 5, 9, 2]))
print(unique_elements([-1, 2, -1, 3, 2, -2]))
print(unique_elements([1000000, 999999, 1000000]))

```

Output

Clear

```
[2, 3, 5, 7, 9]
[2, 3, -1, -2]
[1000000, 999999]

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

Python Online Compiler

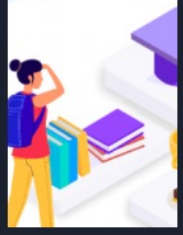
main.py

Share Run

```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

print(bubble_sort([64, 34, 25, 12, 22, 11, 90]))
print(bubble_sort([5, 1, 4, 2, 8]))
```

Output Clear

```
[11, 12, 22, 25, 34, 64, 90]
[1, 2, 4, 5, 8]

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

main.py

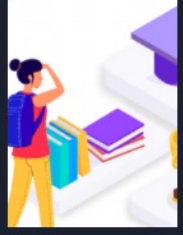
Share Run

```python
def count_valid_pairs(nums, k):
    n = len(nums)
    count = 0

    for i in range(n):
        for j in range(i + 1, n):
            if nums[i] == nums[j] and (i * j) % k == 0:
                count += 1

    return count
nums1 = [3, 1, 2, 2, 2, 1, 3]
k1 = 2
print(count_valid_pairs(nums1, k1))

nums2 = [1, 2, 3, 4]
k2 = 1
print(count_valid_pairs(nums2, k2))

```

Output

Clear

```
4
0

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

main.py

Share | Run

```python
def process_list(nums):
    if not nums:
        return "The list is empty."

    nums.sort()
    return nums[-1]
print(process_list([]))
print(process_list([5]))
print(process_list([3, 3, 3, 3, 3]))

```

Output | Clear

```
The list is empty.
5
3

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

main.py

Share Run

```python
def first_palindromic_string(words):
    for word in words:
        if word == word[::-1]:
            return word
    return ""

words1 = ["abc", "car", "ada", "racecar", "cool"]
words2 = ["notapalindrome", "racecar"]

print(first_palindromic_string(words1))
print(first_palindromic_string(words2))
```

Output Clear

```
ada
racecar

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

main.py

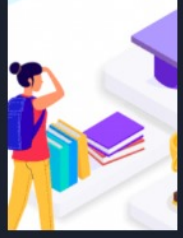
Share Run

```python
def binary_search(arr, key):
    arr.sort()
    left, right = 0, len(arr) - 1

    while left <= right:
        mid = left + (right - left) // 2
        if arr[mid] == key:
            return f"Element {key} is found at position {mid + 1}"
        elif arr[mid] < key:
            left = mid + 1
        else:
            right = mid - 1

    return f"Element {key} is not found"

arr1 = [3, 4, 6, -9, 10, 8, 9, 30]
key1 = 10
print(binary_search(arr1, key1))

arr2 = [3, 4, 6, -9, 10, 8, 9, 30]
key2 = 100
print(binary_search(arr2, key2))
```

Output

Clear

```
Element 10 is found at position 7
Element 100 is not found

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

main.py

Share Run

```python
def find_max(nums):
    return max(nums)
print(find_max([1, 2, 3, 4, 5]))
print(find_max([7, 7, 7, 7, 7]))
print(find_max([-10, 2, 3, -4, 5]))

```

Output Clear

```
5
7
5

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

main.py

Share Run

```python
        return nums

    mid = len(nums) // 2
    left = merge_sort(nums[:mid])
    right = merge_sort(nums[mid:])
    return merge(left, right)

def merge(left, right):
    sorted_array = []
    i = j = 0

    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            sorted_array.append(left[i])
            i += 1
        else:
            sorted_array.append(right[j])
            j += 1

    sorted_array.extend(left[i:])
    sorted_array.extend(right[j:])

    return sorted_array

nums1 = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
print(merge_sort(nums1))
```

Output

Clear

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

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

main.py

Share Run

```python
def count_indices(nums1, nums2):
    answer1 = sum(1 for num in nums1 if num in nums2)
    answer2 = sum(1 for num in nums2 if num in nums1)
    return [answer1, answer2]
nums1 = [2, 3, 2]
nums2 = [1, 2]
print(count_indices(nums1, nums2))

nums1 = [4, 3, 2, 3, 1]
nums2 = [2, 2, 5, 2, 3, 6]
print(count_indices(nums1, nums2))
```

Output Clear

```
[2, 1]
[3, 4]

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