



main.c



Share



Run

Output

Clear

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int isPalindrome(char *str) {
5     int len = strlen(str);
6     for (int i = 0; i < len / 2; i++) {
7         if (str[i] != str[len - i - 1]) {
8             return 0; // Not a palindrome
9         }
10    }
11    return 1; // Palindrome
12 }
13
14 char* findFirstPalindromicString(char **words, int size) {
15     for (int i = 0; i < size; i++) {
16         if (isPalindrome(words[i])) {
17             return words[i];
18         }
19     }
20     return "";
21 }
22
23 int main() {
24     char *words[] = {"abc", "car", "ada", "racecar", "cool"};
25     int size = sizeof(words) / sizeof(words[0]);
26 }
```

/tmp/hXv6lNcuNo.o

Output: "ada"

=== Code Execution Successful ===



main.py



Share

Run

Output

Clear

```
1- def calculate_matching_indices(nums1, nums2):
2     set1 = set(nums1)
3     set2 = set(nums2)
4
5     answer1 = sum(1 for num in nums1 if num in set2)
6     answer2 = sum(1 for num in nums2 if num in set1)
7
8     return [answer1, answer2]
9
10 # Example 1
11 nums1 = [2, 3, 2]
12 nums2 = [1, 2]
13 print(calculate_matching_indices(nums1, nums2)) # Output- [2, 1]
14
15 # Example 2
16 nums1 = [4, 3, 2, 3, 1]
17 nums2 = [2, 2, 5, 2, 3, 6]
18 print(calculate_matching_indices(nums1, nums2)) # Output- [3, 4]
19
```

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

--- Code Execution Successful ---
```



main.py



Share

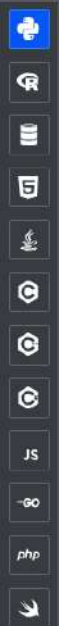
Run

Output

Clear

```
1- def sum_of_squares_distinct_counts(nums):
2-     result = 0
3-     for i in range(len(nums)):
4-         distinct_counts = set()
5-         for j in range(i, len(nums)):
6-             distinct_counts.add(nums[j])
7-             result += len(distinct_counts) ** 2
8-     return result
9-
10 # Example usage
11 nums1 = [1, 2, 1]
12 nums2 = [1, 1]
13 print(sum_of_squares_distinct_counts(nums1)) # Output: 15
14 print(sum_of_squares_distinct_counts(nums2)) # Output: 3
15
```

```
15
3
--- Code Execution Successful ---
```



main.py



Share

Run

Output

Clear

```
3 freq = {}
4
5 for i, num in enumerate(nums):
6     if num in freq:
7         count += freq[num]
8         freq[num] += 1
9     else:
10        freq[num] = 1
11
12    for j in range(i):
13        if (i * j) % k == 0 and nums[i] == nums[j]:
14            count += 1
15
16    return count
17
18 # Example 1
19 nums1 = [3, 1, 2, 2, 2, 1, 3]
20 k1 = 2
21 print(count_pairs(nums1, k1)) # Output: 4
22
23 # Example 2
24 nums2 = [1, 2, 3, 4]
25 k2 = 1
26 print(count_pairs(nums2, k2)) # Output: 0
27
```

```
9
0
--- Code Execution Successful ---
```



main.py



Share

Run

Output

Clear

```
1 # Test Case 1
2 list1 = [1, 2, 3, 4, 5]
3 max_element1 = max(list1)
4 print(max_element1) # Output: 5
5
6 # Test Case 2
7 list2 = [7, 7, 7, 7, 7]
8 max_element2 = max(list2)
9 print(max_element2) # Output: 7
10
11 # Test Case 3
12 list3 = [-10, 2, 3, -4, 5]
13 max_element3 = max(list3)
14 print(max_element3) # Output: 5
15
```

```
5
7
5

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



main.py



Share

Run

Output

Clear

```
1- def find_max_in_sorted_list(input_list):
2-     if not input_list:
3-         return None # Return None if the list is empty
4-
5-     sorted_list = sorted(input_list) # Sort the input list
6-     max_element = sorted_list[-1] # Get the maximum element from the sorted
       list
7-     return max_element
8-
9- # Test Cases
10 print(find_max_in_sorted_list([])) # Expected Output: None or appropriate
    message for empty list
11 print(find_max_in_sorted_list([5])) # Expected Output: 5
12 print(find_max_in_sorted_list([3, 3, 3, 3, 3])) # Expected Output: 3
13
```

None
5
3

=== Code Execution Successful ===



main.py



Share

Run

Output

Clear

```
1 def get_unique_elements(input_list):
2     unique_elements = list(set(input_list))
3     return unique_elements
4
5 # Test the function
6 input_list = [3, 7, 3, 5, 2, 5, 9, 2]
7 print(get_unique_elements(input_list))
8
9
```

```
[2, 3, 5, 7, 9]
=== Code Execution Successful ===
```



main.py



Share

Run

Output

Clear

```
1- def bubble_sort(arr):
2-     n = len(arr)
3-     for i in range(n):
4-         for j in range(0, n-i-1):
5-             if arr[j] > arr[j+1]:
6-                 arr[j], arr[j+1] = arr[j+1], arr[j]
7-     return arr
8-
9 arr = [64, 34, 25, 12, 22, 11, 90]
10 sorted_arr = bubble_sort(arr)
11 print("Sorted array:", sorted_arr)
12
```

Sorted array: [11, 12, 22, 25, 34, 64, 90]

=== Code Execution Successful ===



Upgrade now

Trading is risky. 100% apply

Programiz PRO >



main.py



Share

Run

Output

Clear

```
1- def findPaths(m, n, N, i, j):
2-     MOD = 10**9 + 7
3-     dp = [[[0 for _ in range(N + 1)] for _ in range(n)] for _ in range(m)]
4-
5-     for k in range(1, N + 1):
6-         for x in range(m):
7-             for y in range(n):
8-                 for dx, dy in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
9-                     nx, ny = x + dx, y + dy
10-                    if nx < 0 or nx >= m or ny < 0 or ny >= n:
11-                        dp[x][y][k] += 1
12-                    else:
13-                        dp[x][y][k] = (dp[x][y][k] + dp[nx][ny][k - 1]) % MOD
14-
15-     return dp[i][j][N]
16-
17- # Test cases
18- print(findPaths(2, 2, 2, 0, 0)) # Output: 6
19- print(findPaths(1, 3, 3, 0, 1)) # Output: 12
20-
```

```
6
12
--- Code Execution Successful ---
```

Activate Windows
Go to Settings to activate Windows.



main.py



Share

Run

Output

Clear

```
1- def rob(nums):
2-     def rob_range(start, end):
3-         rob, not_rob = 0, 0
4-         for i in range(start, end):
5-             rob, not_rob = not_rob + nums[i], max(rob, not_rob)
6-         return max(rob, not_rob)
7-
8-     if len(nums) == 1:
9-         return nums[0]
10-    return max(rob_range(0, len(nums) - 1), rob_range(1, len(nums)))
11-
12- # Example usage
13- nums1 = [2, 3, 2]
14- nums2 = [1, 2, 3, 1]
15- print("Maximum money robbed without alerting police (nums1):", rob(nums1))
16- print("Maximum money robbed without alerting police (nums2):", rob(nums2))
17-
```

Maximum money robbed without alerting police (nums1): 3
Maximum money robbed without alerting police (nums2): 4

--- Code Execution Successful ---

Activate Windows
Go to Settings to activate Windows.