

Python Function Problems

1. Basic Function: Total Calculator

Write a function `calculate_total(values)` that takes a list of numerical values and returns the sum of these values. For example, `calculate_total([3.50, 4.75, 2.00])` should return `10.25`.

2. Basic Function: Weekly Average

Write a function `average_value(values)` that takes a list of values for a week and returns the average. For example, `average_value([72, 75, 78, 77, 74, 70, 68])` should return `73.0`.

3. Basic Function: Discount Adjuster

Write a function `apply_adjustment(amount, percentage)` that takes an initial amount and a percentage adjustment, and returns the adjusted amount. For example, `apply_adjustment(100, 15)` should return `85.0`.

4. Anonymous Function: Square Elements

Use a lambda function to create a list of squared values from a given list of integers. For example, given the list `[1, 2, 3, 4]`, your lambda function should produce `[1, 4, 9, 16]`.

5. Map Function: Convert Units

Write a function `convert_units(values)` that uses `map` and a lambda function to convert a list of values from one unit to another. For example, converting Fahrenheit to Celsius for `[32, 212]` should return `[0, 100]`.

6. Filter Function: Extract Even Values

Write a function `extract_even(values)` that uses `filter` and a lambda function to return a list of even values from a given list. For example, `extract_even([1, 2, 3, 4, 5])` should return `[2, 4]`.

7. Reduce Function: Compute Product

Write a function `compute_product(values)` that uses `reduce` to calculate the product of all elements in a list of values. For example, `compute_product([1, 2, 3, 4])` should return `24`.

8. Sorted Function: Rank Values

Write a function `rank_values(values)` that uses `sorted` to sort a list of values in descending order. For example, `rank_values([88, 92, 79, 95])` should return `[95, 92, 88, 79]`.

9. Any Function: Check Threshold

Write a function `check_threshold(values, threshold)` that uses `any` to check if any value in the list exceeds a given threshold. For example, `check_threshold([55, 80, 65], 60)` should return `True`.

10. All Function: Verify Threshold

Write a function `verify_threshold(values, threshold)` that uses `all` to check if all values in the list exceed a given threshold. For example, `verify_threshold([65, 70, 75], 60)` should return `True`.

11. Basic Function: Update Collection

Write a function `update_collection(collection, key, value)` that updates a collection (e.g., a dictionary) with the given key and value. If the key already exists, update its value. For example, `update_collection({'item1': 5}, 'item2', 3)` should return `{'item1': 5, 'item2': 3}`.

12. Lambda Function: Custom Sort Key

Use a lambda function with `sorted` to sort a list of tuples by a specific element in each tuple. For example, given `[(1, 3), (2, 2), (3, 1)]`, it should return `[(3, 1), (2, 2), (1, 3)]`.

13. Map Function: Convert Cases

Write a function `convert_cases(strings)` that uses `map` and a lambda function to convert a list of strings to uppercase. For example, `convert_cases(['alice', 'bob', 'charlie'])` should return `['ALICE', 'BOB', 'CHARLIE']`.

14. Filter Function: Long Word Filter

Write a function `filter_long_words(words, min_length)` that uses `filter` and a lambda function to return a list of words that are longer than a given minimum length. For example, `filter_long_words(['cat', 'elephant', 'dog'], 3)` should return `['elephant']`.

15. Reduce Function: Join Strings

Write a function `join_strings(strings)` that uses `reduce` to concatenate a list of strings into a single string. For example, `join_strings(['hello', 'world'])` should return `'helloworld'`.