
LISTS

1. **List of Squares** Create a list of squares of numbers from 1 to 20.
2. **Second Largest Number** Find the second largest number in a list without using `sort()`.
3. **Remove Duplicates** Write a program to remove all duplicate values from a list while preserving order.
4. **Rotate List** Rotate a list to the right by `k` steps. *Example: [1, 2, 3, 4, 5] rotated by 2 → [4, 5, 1, 2, 3]*
5. **List Compression** From a list of numbers, create a new list with only the even numbers doubled.

TUPLES

6. **Swap Values** Write a function that accepts two tuples and swaps their contents.
7. **Unpack Tuples** Unpack a tuple with student details: `(name, age, branch, grade)` and print them in a sentence.
8. **Tuple to Dictionary** Convert a tuple of key-value pairs into a dictionary. *Example:* `(("a", 1), ("b", 2)) → {"a": 1, "b": 2}`

SETS

9. **Common Items** Find the common elements in two user-defined lists using sets.
10. **Unique Words in Sentence** Take a sentence from the user and print all unique words.
11. **Symmetric Difference** Given two sets of integers, print elements that are in one set or the other, but not both.
12. **Subset Checker** Check if one set is a subset of another.

DICTIONARIES

13. **Frequency Counter** Count the frequency of each character in a string using a dictionary.
14. **Student Grade Book** Ask for names and marks of 3 students. Then ask for a name and display their grade (`>=90: A`, `>=75: B`, else C).
15. **Merge Two Dictionaries** Merge two dictionaries. If the same key exists, sum the values.
16. **Inverted Dictionary** Invert a dictionary's keys and values. *Input:* `{"a": 1, "b": 2}` *→ Output:* `{1: "a", 2: "b"}`

17. **Group Words by Length** Input a list of words. Create a dictionary where the key is word length and the value is a list of words of that length.
