**🧩 namedtuple()**

1. **Employee Record:**  
   Create a namedtuple called Employee with fields id, name, and salary. Create three employee records and print the employee with the highest salary.
2. **Car Details:**  
   Define a namedtuple called Car with fields brand, model, price. Create 2–3 car objects and display them in a formatted output.

**🔄 deque**

1. **Customer Queue Simulation:**  
   Use a deque to simulate a customer queue at a billing counter. Add customers as they arrive and remove them as they are served.
2. **Palindrome Checker:**  
   Write a program using deque to check whether a given word is a palindrome (reads the same backward and forward).

**🪢 ChainMap**

1. **Configuration Settings:**  
   Create two dictionaries: default\_config and user\_config. Combine them using ChainMap so that user settings override the defaults.
2. **Multiple Department Budgets:**  
   You have department budgets as separate dictionaries (sales, hr, it). Use ChainMap to access them as a single structure and print total combined keys.

**🔢 Counter**

1. **Voting System:**  
   Given a list of votes = ['A', 'B', 'A', 'C', 'A', 'B'], use Counter to determine which candidate won.
2. **Character Frequency:**  
   Count the frequency of each letter in the word "mississippi" and print the two most common characters.

**📋 OrderedDict**

1. **Shopping List:**  
   Create an OrderedDict of grocery items in the order they are added. Print them and then reverse the order of insertion.
2. **Task Tracker:**  
   Maintain a sequence of daily tasks (like “Check email”, “Attend meeting”, “Write report”). Print them in order of creation and then reorder one task manually.

**🗂️ defaultdict**

1. **Group Cities by State:**  
   Given pairs of (state, city), group all cities under their respective states using defaultdict(list).
2. **Word Categorization:**  
   Categorize words based on their first letter using defaultdict — for example, grouping ['apple', 'banana', 'berry', 'cherry', 'carrot'].

**📞 UserDict**

1. **RestrictedDictionary:**  
   Create a custom dictionary using UserDict that does **not allow deletion** of keys. Raise an error if del is used.
2. **StudentGrades:**  
   Create a class StudentGrades based on UserDict that ensures all grades added are between 0 and 100, otherwise raises a ValueError.

**📝 UserList**

1. **PositiveNumberList:**  
   Create a UserList subclass that only allows positive numbers to be appended. If a negative number is added, raise an exception.
2. **TodoList Enhancement:**  
   Extend the TodoList example so it does not allow duplicate tasks (ignore if the task already exists).

**🔤 UserString**

1. **NoVowelsString:**  
   Create a UserString subclass that removes all vowels from the input text.
2. **TitleCaseString:**  
   Build a custom string class that automatically converts all words to Title Case (first letter capitalized) when created.