Medium Level: Card Collection System Problem Statement 📝 Create a program to collect and store all the cards (e.g., playing cards) and assist users in finding all the cards of a given symbol using the Collection interface.

Key Concepts Used 🛠️ HashMap: To store cards with their symbols as keys.

Collection Interface: To manage and manipulate the card data.

User Interaction: Allow users to search for cards by symbol.

Code:

```
import java.util.*;

class CardCollection {
    private final Map<String, List<String>> cardMap;

    public CardCollection() {
        cardMap = new HashMap<>();
    }
    public void addCard(String symbol, String cardName) {
        cardMap.computeIfAbsent(symbol, k -> new
ArrayList<>()).add(cardName);
        System.out.println("Card added successfully!");
    }
    public void searchBySymbol(String symbol) {
        List<String> cards = cardMap.get(symbol);
        if (cards != null && !cards.isEmpty()) {
            System.out.println("Cards with symbol '" + symbol + "': " +
cards);
        } else {
            System.out.println("No cards found for symbol: " + symbol);
        }
    }
    public void displayAllCards() {
        if (cardMap.isEmpty()) {
            System.out.println("No cards in the collection.");
        } else {
            for (Map.Entry<String, List<String>> entry :
cardMap.entrySet()) {
                System.out.println("Symbol: " + entry.getKey() + " -> " +
entry.getValue());
            }
        }
    }
}
public class CardCollectionSystem {
    private static final Scanner scanner = new Scanner(System.in);
    private static final CardCollection cardCollection = new
CardCollection();

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nCard Collection System");
            System.out.println("1. Add Card");
            System.out.println("2. Search Cards by Symbol");
            System.out.println("3. Display All Cards");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
```

```
            scanner.nextLine(); // Consume newline
            switch (choice) {
                case 1 -> addCard();
                case 2 -> searchCard();
                case 3 -> cardCollection.displayAllCards();
                case 4 -> {
                    System.out.println("Exiting...");
                    return;
                }
                default -> System.out.println("Invalid choice! Please try
again.");
            }
        }
    }
    private static void addCard() {
        System.out.print("Enter Card Symbol (e.g., Hearts, Diamonds): ");
        String symbol = scanner.nextLine();
        System.out.print("Enter Card Name (e.g., Ace, King, Queen): ");
        String cardName = scanner.nextLine();
        cardCollection.addCard(symbol, cardName);
    }
    private static void searchCard() {
        System.out.print("Enter Card Symbol to search: ");
        String symbol = scanner.nextLine();
        cardCollection.searchBySymbol(symbol);
    }
}
```

Output: