C++ Problem Statements

1. Library Book Inventory

Problem Statement:
Create a program that manages a library's book inventory. The user can add, remove, or search for books based on title.

Sample Input:

```
1 (Add Book)
Enter book title: The Great Gatsby
2 (Search Book)
Enter book title: The Great Gatsby
```

Sample Output:

```
Book 'The Great Gatsby' added successfully.
Book 'The Great Gatsby' found in the inventory.
```

---

2. Online Voting System

Problem Statement:
Design a voting system where users can vote for candidates. Each user can only vote once.

Sample Input:

```
Enter your vote (1 for Alice, 2 for Bob): 1
Enter your vote (1 for Alice, 2 for Bob): 1
```

Sample Output:

```
Vote recorded for Alice.
You have already voted!
```

---

3. ATM Simulation

Problem Statement:
Create an ATM simulation that allows users to check balance, deposit, and withdraw money.

Sample Input:

```
1 (Check Balance)
2 (Deposit)
Enter amount: 500
3 (Withdraw)
Enter amount: 200
4 (Exit)
```

Sample Output:

```
Your balance is: 0
Deposit successful. New balance: 500
Withdrawal successful. New balance: 300
Thank you for using the ATM.
```

---

4. Student Grading System

Problem Statement:
Implement a grading system that takes student scores and assigns letter grades.

Sample Input:

```
Enter score: 85
```

Sample Output:

```
Grade: B
```

---

5. E-commerce Shopping Cart

Problem Statement:
Design a shopping cart system where users can add and remove items, and calculate total price.

Sample Input:

```
Add item price: 100
Add item price: 200
Remove item price: 100
Calculate total
```

Sample Output:

```
Total price: 200
```

---

6. Weather Forecast Application

Problem Statement:
Create a weather application that provides clothing recommendations based on weather conditions.

Sample Input:

```
Enter weather condition (1 for Sunny, 2 for Rainy): 2
```

Sample Output:

```
Recommendation: Wear a raincoat.
```

---

7. Bank Account Management

Problem Statement:
Build a bank account management system to create accounts and perform transactions.

Sample Input:

```
1 (Create Account)
Enter initial deposit: 1000
2 (Check Balance)
3 (Withdraw)
Enter amount: 500
4 (Exit)
```

Sample Output:

```
Account created with balance: 1000
Your balance is: 1000
Withdrawal successful. New balance: 500
Thank you for using the bank.
```

---

8. Traffic Light Simulation

Problem Statement:
Simulate a traffic light system that changes lights every few seconds.

Sample Input:
(Assume input is through time intervals)

Sample Output:

```
Red Light - Stop!
Green Light - Go!
Yellow Light - Caution!
```

---

9. Maze Solver

Problem Statement:
Write a program that checks if there is a path in a maze represented by a grid of `0`s and `1`s.

Sample Input:
Maze:

```
0 1 0
0 0 0
1 1 0
```

Sample Output:

```
Path exists.
```

---

10. Password Strength Checker

Problem Statement:
Create a password strength checker that evaluates user passwords based on criteria.

Sample Input:

```
Enter password: Pass123!
```

Sample Output:

```
Password strength: Strong.
```

---

11. Inventory Management System

Problem Statement:
Develop an inventory management system to track stock levels of products.

Sample Input:

```
Add product stock: Product A, Quantity: 10
Check stock of Product A.
Remove stock of Product A, Quantity: 5.
Check stock of Product A.
```

Sample Output:

```
Product A stock is available.
New quantity of Product A: 5.
```

---

12. Expense Tracker

Problem Statement:
Build an expense tracker to log daily expenses by category.

Sample Input:

```
Log expense - Category: Food, Amount: 50
Log expense - Category: Transport, Amount: 20
Show total expenses.
```

Sample Output:

```
Total expenses: $70.
```

---

13. Sudoku Validator

Problem Statement:
Implement a Sudoku validator that checks if a given board is valid.

Sample Input:
(Sudoku board as input)

```
5 . . | . . . | . . .
6 . . | . . . | . . .
. . . | . . . | . . .
---------------------
8 . . | . . . | . . .
4 . . | . . . | . . .
7 . . | . . . | . . .
---------------------
. . . | 2 . . | . .
. . . | . . . | 6 .
. . .
```

Sample Output:

```
Sudoku board is valid.
```

---

14. Recipe Manager

Problem Statement:
Design a recipe manager that allows users to add and search recipes by ingredients.

Sample Input:

```
Add recipe - Name: Pancakes, Ingredients: Flour, Eggs, Milk
Search by ingredient - Flour
```

Sample Output:

```
Recipe found: Pancakes includes Flour.
```

---

15. Game Score Tracker

Problem Statement:
Create a game score tracker where scores can be added or subtracted based on events.

Sample Input:

```
Add score for Player A: +10
Subtract score for Player B: -5
Show scores
```

Sample Output:

```
Scores -> Player A: +10, Player B: -5
```

---

16. Personal Finance Advisor

Problem Statement:
Develop a tool that analyzes income and expenses to provide savings recommendations.

Sample Input:

```
Enter income: $3000
Enter expenses (Food $500, Rent $1000)
Show savings recommendation
```

Sample Output:

```
You should save at least $1500 this month!
```

---

17. Quiz Application

Problem Statement:
Write a quiz application that presents questions and calculates scores.

Sample Input:

```
Question1 - What is the capital of France? (A) Paris (B) London
Your answer (A): A
Question2 - What is the capital of Germany? (A) Berlin (B) Madrid
Your answer (B): A
Calculate score
```

Sample Output:

```
Your score is: 1/2.
```

---

18. Fitness Tracker

Problem Statement:
Build an application that logs daily activities and calculates calories burned.

Sample Input:

```
Log activity - Type: Running, Duration(min):30, Calories/minute:10
Show total calories burned today.
```

Sample Output:

```
Total calories burned today:300.
```

---

19. Flight Reservation System

Problem Statement:
Create a flight reservation system where users can search and book flights based on destination and date.

Sample Input:

```
Search flights to New York on Dec-20-2024? Yes/No:
Book flight? Yes/No:
```

Sample Output:

```
Flight booked successfully to New York on Dec-20-2024.
```

---

20. Chatbot Simulator

Problem Statement:
Design a simple chatbot simulator that responds to user inputs with predefined answers.

Sample Input:

```
User input: Hi
User input: Tell me about the weather
User input: Bye
```

Sample Output:

```
Bot response: Hello! How can I help you today?
Bot response: The weather is sunny today!
Bot response: Goodbye! Have a great day!
```

Real-World Data

Library Inventory Data:

- Book Titles:
  - The Great Gatsby
  - To Kill a Mockingbird
  - Moby Dick

Voting Data:

- Candidates:
  - Alice
  - Bob

ATM Data:

- Initial Balance Example:
  - User starts with $1000

Student Grades Example:

- Score Ranges:
  - A = [90-100]
  - B = [80-89]
  - C = [70-79]
  - D = [60-69]
  - F = [0-59]

E-commerce Prices:

- Item Prices:
  - Item A = $100
  - Item B = $200

Weather Conditions:

- Conditions:
  - Sunny
  - Rainy

Bank Account Data:

- Initial Deposit Example:
  - $1000

Traffic Light Timings:

- Light Change Intervals:

    - Red = 30 seconds, Green = 30 seconds, Yellow = 5 seconds

Maze Example:

- Maze Layouts as Grids of `0`s and `1`s

Password Criteria:

- Minimum Length = 8 characters, Must include uppercase, lowercase, digits, symbols

Inventory Stock Levels:

- Product Stock Examples:
    - Product A = Quantity of `10`, Product B = Quantity of `5`.

Expense Categories:

- Categories Examples:
    - Food, Transport, Utilities

Sudoku Board Example:

- Valid Sudoku Board Layouts

Recipe Ingredients Example:

- Ingredients List Examples for Recipes

Game Scores Example:

- Player Scores Example

Personal Finance Data Example:

- Income and Expense Examples

Quiz Questions Example:

- Sample Questions with Correct Answers

Fitness Activity Examples:

- Activity Types with Duration and Calories Burned per Minute

Flight Reservation Example Data:

- Flight Destinations with Dates

Chatbot Responses Example Data:

- Predefined Responses for Common Queries
