1. **Project introduction:**

   This project implements the Banker's Algorithm, The program reads the system state from an input file and determines whether the system is in a safe state. If the system is safe, it will print out a safe sequence of processes.

2. **How it works**

   It prevents deadlocks by ensuring that the system does not enter an unsafe state. The algorithm calculates the need for each process (Need = Max - Allocation) based on the currently available resources, each process's maximum needs, and allocated resources. Then, it attempts to find at least one safe sequence of process execution. This sequence ensures that each process can obtain all the resources it may need, complete its tasks, and release resources without causing the system to enter into a deadlock.

3. **input.txt**

```
3 3 2       // Available resources (A: 3, B: 3, C: 2)

7 5 3       // Maximum resources needed by P0
0 1 0       // Resources currently allocated to P0

3 2 2       // Maximum resources needed by P1
2 0 0       // Resources currently allocated to P1

9 0 2       // Maximum resources needed by P2
3 0 2       // Resources currently allocated to P2

2 2 2       // Maximum resources needed by P3
2 1 1       // Resources currently allocated to P3

4 3 3       // Maximum resources needed by P4
0 0 2       // Resources currently allocated to P4
```

4. **Answer the following question：Is the system in a safe state? If Yes, then what is the safe sequence?**