**Lab Assignment 7**

**Write a program to implement a Banker’s algorithm.**

**1. Objective**

* To implement the **Banker's Algorithm** for deadlock avoidance.
* To check if the system is in a **safe state**.
* To determine if a resource request can be granted safely.

**2. Theory**  
The **Banker's Algorithm** is a resource allocation and deadlock avoidance algorithm developed by Edsger Dijkstra. It tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, and then makes a "safe-state" check to test for possible deadlock conditions for all other pending activities before deciding whether allocation should be allowed to continue.

**Key Data Structures:**

* **Available**: Vector showing available instances of each resource type.
* **Max**: Maximum demand of each process.
* **Allocation**: Resources currently allocated to each process.
* **Need**: Remaining resources needed by each process (Need = Max - Allocation).

**3. Tools**

* **gcc** compiler
* **Linux Terminal**
* **C Programming Language**

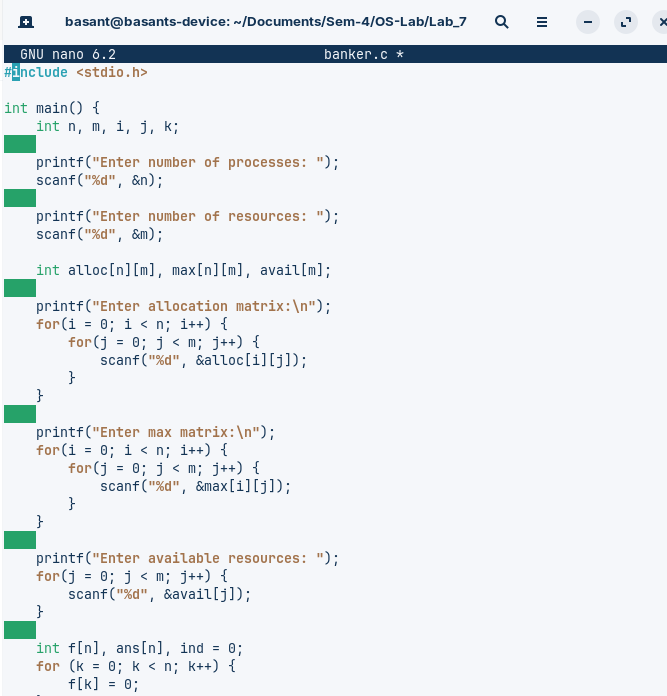
**4. Procedure**

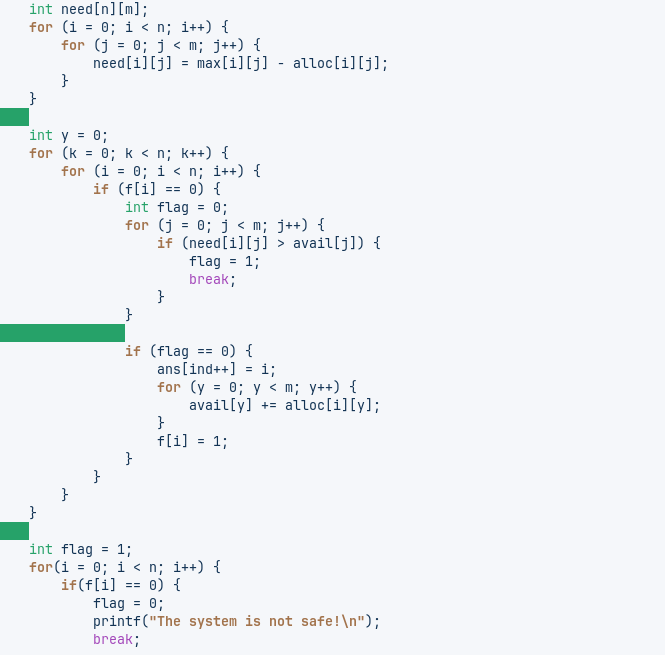
1. Input number of processes and resource types.
2. Input the **Available**, **Max**, and **Allocation** matrices.
3. Calculate the **Need** matrix.
4. Execute the **Safety Algorithm**:
   * Find a process whose **Need** <= **Available**.
   * Assume it finishes and releases its resources.
   * Repeat until all processes are finished (safe) or no such process exists (unsafe).
5. Display the result (safe/unsafe) and the safe sequence if applicable.

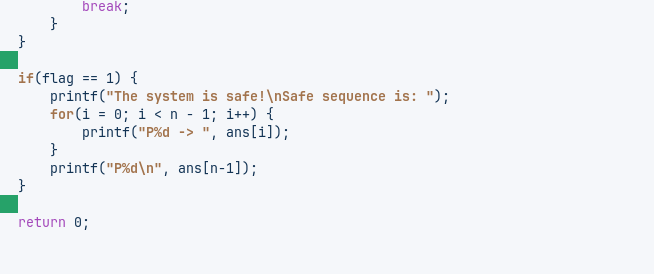
**5. Program**  
a. Create and open file:



b. Write the code.







c. Compile:



d. Run and test:



**6. Conclusion**  
The Banker's Algorithm was successfully implemented. The program correctly determines if a system is in a safe state, preventing deadlock by ensuring safe resource allocation. The safe sequence of process execution is identified when possible.