**Bankers Algorithm Project 3 Report**

**Design:**

* This implementation uses the Bankers Algorithm to simulate resource allocation for 5 processes
  + **Processes:** Each process has a maximum resource requirement and allocation of resources
  + **Available:** Tracks the current available resource of each type
  + **Max Matrix:** Represents the maximum number of resources each process might need
  + **Allocation Matrix:** Tracks the resources currently allocated to each process
  + **Need Matrix:** Using Max and Allocation to calculate and represent the remaining resource requirement for each process

**Functions:**

**printMatrix(int arr[][m], int rows):**

* This is used to print a matrix into a readable format. Iterates through each row and column and prints the values into a matrix format

**findNeed(int need[n][m], int maxm[n][m], int allot[n][m]):**

* This is used to compute the Need matrix. Basically the difference between the Max matrix and Allocation Matrix. Represents remaining resource requirement for each process

**safe(int process[], int available[], int maxm[][m], int allot[][m]):**

* This checks if the system is in a safe state by using the Banker’s Algorithm. We basically simulate the allocation of resources to the process in a safe sequence. We use the Need matrix, available resources, and allocation matrix to see if a safe sequence exists. If no safe sequence is found, then the system is in an unsafe state.

**request\_resources(int p, int request[], int available[], int maxm[][m], int allot[][m], int processes[]):**

* This handles resource requests by a process. First it checks if the requested resources are within the limits of the process’s maximum claim and if the resources are available. After ensuring that the request is valid, temporarily allocates the resources and checks if the system is still in a safe state. If it is still safe then resources are committed, if not then it is denied.

**Main:**

1. We first initialize the process and resource data

* The number of processes is 5 and the number of resource types is 3 and is predefined
* Matrices for the Max and Allocated resources are initialized
* Available resources is also included

1. We run three test cases:

* **First**: Checks if the system is initially in a safe state by running the Banker’s Algorithm on the matrices
* **Second**: Simulates a resource request by Process 1. Request is then checked and reevaluated for state
* **Third**: Simulates resource requests by Process 4. Request is rejected because it exceeds the available resources

**Video:**<https://youtu.be/2FMJ1A2EuTM>