CS - 310 Communicating Distributed Processes

Lab Assignment - 2

Implementation of banker's algorithm: Simulate the execution of p processes in an operating system where there are r resources with R, V, C, A representing the states of the execution. R, V, C, A are the standard notations for resources, available, claim, allocation. Implement this with the help of the Deadlock avoidance strategy - "BANKER's ALGORITHM".

- The simulation program should initialize (say based on user input) the resource vector R and the claim matrix C.
- Processes need to query for resources randomly (upper limited by their claim row in C)
- The banker algorithm must check whether it can allow the requested resources or put the process in blocked state.
- You have to update timeline of all processes each second as shown below, with '|' if resources will be given and '-' if resources will not be given. Show updated timeline of all processes every second.
- As and when a process gets all the claimed resources then it must get terminated with its claim row in C getting reset after returning the resources to available resource vector V.
- Once a process gets terminated, it should not produce any further requests for resources.

Run the simulation for a given time or till the given processes terminate to completion. After initialization of Allocation matrix, first check whether process are already in deadlock or not and then proceed accordingly. Also if unsafe state is reached then it should report that it is unsafe. The output should contain the time-line of each process conveying the time of their run or suspension etc or should report unsafe.

Input Example:

```
// p = number of process  r = number of resources
3 5
// R = resource Vector
2 3 1 2 4
// C= Claim Matrix
0 2 2 3 0 // p1
3 3 1 2 3 // p2
5 2 2 3 1 // p3
```

Sample output style: (This is sample output style, this output is not for given example)

Time when process is not running : ‘-’
Time when process is running : ‘|’