

Implementation of a Simulator for Banker's Algorithm

Problem Description:

You are to make a simple simulator to implement the Banker's algorithm. Given a set of data, which represent the current status of the system, as time passes, when each process requests resources your simulator should determine whether the request can be granted or delayed. To have the system be in a safe state your simulator is using the Banker's algorithm.

Each process requests a set resources and OS checks whether the request can be granted immediately. If the system remains safe after granting the request, OS allows the process to use the requested set of resource. If not, the request is delayed and saved in a waiting queue. When system status changes (ie, some processes release their resources) OS checks in FIFO order whether the delayed requests can be granted. Note that when 2 or more requests have been delayed (ie, are waiting in the waiting queue) and a running process releases a lot of resources multiple delayed requests can be granted one by one in FIFO order.

If the request asked by process P_i is bigger than $NEED[i]$, it is unallowable request. If such requests arrive, OS ignores them. In other words, such requests are totally ignored and not saved in the waiting queue.

Input:

Input file name is 'banker.inp'. In the first line, an integer n which represents the number of processes and another integer m which represents the number of resource types are given. Neither n nor m is greater than 50. The indices for n processes are from 0 to $n-1$ and 0 to $m-1$ for m resource types.

In the next line, m integers are given, each of which represents the number of instances for each resource type in the system. Next line is just a blank line to increase the readability of the input. After that, the MAX matrix of size $n \times m$, which represents the maximum requests for each resource type by every process, is given along n lines, m values in each line. The next line is also another blank line. After that, the ALLOCATION matrix of size $n \times m$, which represents the current usage for each resource type by every process, is given along n lines, m values in each line. The next line is also another blank line.

In the following lines, there are commands issued by processes. Each line consists of a command and its parameters, which represents which process requests or releases how many instances for each resource type.

There commands are used:

- 1) When process P_i requests ($m_0 m_1 \dots m_{m-1}$):
request i $m_0 m_1 \dots m_{m-1}$
- 2) When process P_i releases ($m_0 m_1 \dots m_{m-1}$):
release i $m_0 m_1 \dots m_{m-1}$
- 3) To stop the simulator:
quit

Output:

Output file name is 'banker.out'. For each command in the input file, your program prints the available resources in the system for each resource type after processing (either granting or delaying) the command. Each value is

identified by one or more blanks.

Sample:

Input	Output
4 5	2 4 4 1 3
10 10 6 10 7	2 3 4 0 3
	2 3 4 0 3
2 1 1 3 3	3 4 5 2 5
3 3 2 2 1	3 4 3 1 5
3 1 3 5 0	
1 5 0 4 1	
1 1 0 0 2	
3 1 0 1 0	
2 1 1 2 0	
1 3 0 3 1	
request 0 1 0 1 3 1	
request 3 0 1 0 1 0	
request 2 0 0 2 3 0	
release 0 1 1 1 2 2	
release 3 0 0 0 2 0	
quit	