The task for this assignment was to implement a deadlock detection and correction algorithm for an OS.
It processes the “current state” of a system, detects if there is a deadlock, and acts correspondingly to the result of the detection.

Preliminary notation:

- Ri denotes a resource request vector for process i.
- Ai denotes a current resource allocation vector for process i.
- U denotes an unallocated (available) resource vector.

The input for my implementation consists of several lines. The first line of the input
‘n m’
contains two integers:
n - number of processes in the system (assume processes are numbered from 1 to n)
m - number of resources (assume resources are numbered from 1 to m).

This line is then followed by 2n + 1 lines, each containing m integer numbers:
R11, R12, ..., R1m
R21, R22, ..., R2m
. . .
Rn1, Rn2, ..., Rnm
A11, A12, ..., A1m
A21, A22, ..., A2m
. . .
An1, An2, ..., Anm
U1, U2, ..., Um

For example, the input
3 5
0 1 0 0 0
1 0 0 0 0
0 0 0 0 0
1 0 0 1 0
0 2 1 0 0
0 1 1 0 1
0 0 0 0 0
Is not deadlocked.

The input
3 5

0 1 0 0 0
0 0 0 0 1
0 1 0 0 0
1 0 0 0 0
0 2 0 0 0
0 1 1 1 0
0 0 1 1 0
Is deadlocked.

If the system is not deadlocked, my program will print the order of completion of processes. For the first example above, this will be:
3 1 2

Note: If more than one process can complete the execution at any time, they are printed in increasing order of process number.

If the system is deadlocked, my program will find a possible resolution for the deadlock: i.e. it finds the one or more processes that need to be terminated to resolve the deadlock and shows the order of completion of the remaining processes. The output, in this case, will contain three lines:

- The first line contains the process numbers for processes involved in the deadlock.
- The second line contains the process numbers for processes to be terminated.
- The third line contains the order of completion of the remaining processes.

For the second example above this will be:
1 2 3
3
1 2

The decision of which process(es) involved in a deadlock needs to be terminated is based on the following two rules:
1. Terminate the process with the largest total allocation of all resources.
2. If there are several deadlocked processes with the same amount of total resource allocation, terminate one with the smallest process number.

Note: There might be more than one deadlock cycle in the system. It is possible that more than one process needs to be terminated to resolve the deadlock.