

## Deadlock

**Due Date:** 1<sup>th</sup> April 2024

Write a program that detects a deadlock and find the smallest set of processes whose termination ends the deadlock.

Input of the program are

1. Total number of distinct resources.
2. Number of instances of each resource type. This is a vector. So {3,5,2}, indicates resource type R0 has 3 instances, resource type R1 has 5 instances, and so on.
3. Allocation matrix. It contains the number of allocated resources of each process.  
For example, in the below allocation matrix, there are three distinct resources and 1 instance of resource type R0 is allocated to P2, 3 instances of resource type R1 is allocated to P2 and 1 instance of resource type R2 is allocated to P2.

Processes	Allocation Matrix		
	R0	R1	R2
P0	0	1	0
P1	1	0	2
P2	1	3	1
P3	0	0	1
P4	4	2	3

4. Request matrix. It contains the resources requested by each process.  
For example, in the below request matrix, there are three distinct resources and 3 instances of resource type R0 is requested by P2, 1 instance of resource type R1 is requested by P2 and 1 instance of resource type R2 is requested by P2.

Processes	Request Matrix		
	R0	R1	R2
P0	1	1	0
P1	1	2	3
P2	3	1	1
P3	0	0	1
P4	3	2	2

Output of the program

If there is no deadlock, the program simply outputs no deadlock. Otherwise it outputs that deadlock exists. As already mentioned, it then also outputs the smallest set of processes whose termination ends the deadlock. If more than one solutions exist, then it simply outputs any one solution.