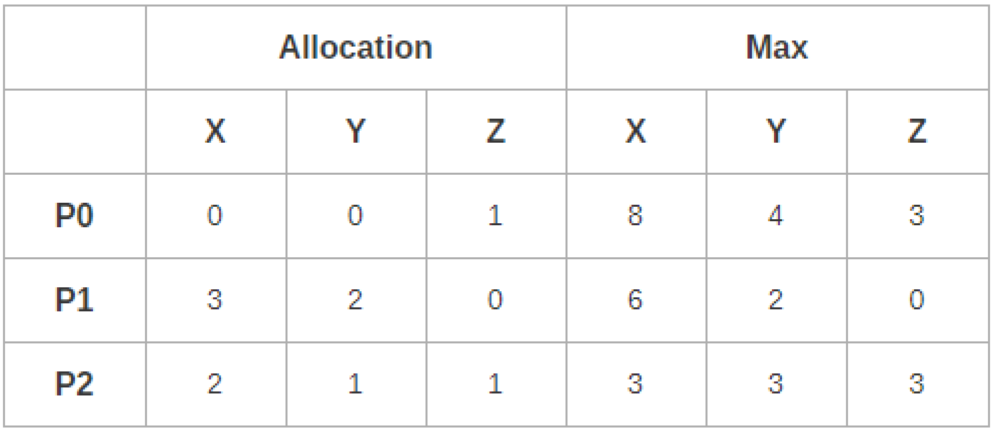
Lab Assignment 7

1. **WAP to implement Banker's Algorithm in C and find whether the processes are in safe state or not?**

The basic data structures used to implement this algorithm are given below.
Let n be the total number of processes and m be the total number of resource types in the system.
Available: A vector of length m. It shows number of available resources of each type. If Available[i] = k, then k instances of resource R_i are available.
Max: An n×m matrix that contain maximum demand of each process. If Max[i,j] = k, then process P_i can request maximum k instances of resource type R_j.
Allocation: An n×m matrix that contain number of resources of each type currently allocated to each process. If Allocation[i,j] = k, then P_i is currently allocated k instances of resource type R_j.
Need: An n×m matrix that shows the remaining resource need of each process. If Need[i,j] = k, then process P_i may need k more instances of resource type R_j to complete the task.

Example:

An operating system uses the banker's algorithm for deadlock avoidance when managing the allocation of three resource types X, Y and Z to three processes P0, P1 and P2. The table given below presents the current system state. Here, the Allocation matrix shows the current number of resources of each type allocated to each process and the Max matrix shows the maximum number of resources of each type required by each process during its execution.



	Allocation			Max		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in safe state. Consider the following independent requests for additional resources in the current state-

REQ1: P0 requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: P1 requests 2 units of X, 0 units of Y and 0 units of Z