**DeadLock - Bankers**

**import** java.util.Scanner;

**public** **class** DeadLock {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of processes: ");

**int** numProcesses = sc.nextInt();

System.***out***.print("Enter the number of resource types: ");

**int** numResources = sc.nextInt();

**int**[][] max = **new** **int**[numProcesses][numResources];

**int**[][] allocation = **new** **int**[numProcesses][numResources];

**int**[][] need = **new** **int**[numProcesses][numResources];

**int**[] available = **new** **int**[numResources];

**boolean**[] finished = **new** **boolean**[numProcesses];

**int**[] safeSequence = **new** **int**[numProcesses];

System.***out***.println("Enter MAX matrix: ");

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

max[i][j] = sc.nextInt();

}

}

System.***out***.println("Enter Allocation matrix: ");

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

allocation[i][j] = sc.nextInt();

}

}

System.***out***.println("Enter Available matrix: ");

**for** (**int** i = 0; i < numResources; i++) {

available[i] = sc.nextInt();

}

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

need[i][j] = max[i][j] - allocation[i][j];

}

}

System.***out***.println("\nMAX matrix:");

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

System.***out***.print(max[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("\nAllocation matrix:");

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

System.***out***.print(allocation[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("\nNeed matrix:");

**for** (**int** i = 0; i < numProcesses; i++) {

**for** (**int** j = 0; j < numResources; j++) {

System.***out***.print(need[i][j] + " ");

}

System.***out***.println();

}

System.***out***.println("\nAvailable matrix:");

**for** (**int** i = 0; i < numResources; i++) {

System.***out***.print(available[i] + " ");

}

System.***out***.println();

**int** count = 0;

**while** (count < numProcesses) {

**boolean** found = **false**;

**for** (**int** p = 0; p < numProcesses; p++) {

**if** (!finished[p]) {

**int** j;

**for** (j = 0; j < numResources; j++) {

**if** (need[p][j] > available[j]) {

**break**;

}

}

**if** (j == numResources) {

**for** (**int** k = 0; k < numResources; k++) {

available[k] += allocation[p][k];

}

safeSequence[count++] = p;

finished[p] = **true**;

found = **true**;

}

}

}

**if** (!found) {

System.***out***.println("The system is in an unsafe state (deadlock).");

**return**;

}

}

System.***out***.print("Safe sequence: ");

**for** (**int** i = 0; i < numProcesses; i++) {

System.***out***.print("P" + safeSequence[i]);

**if** (i != numProcesses - 1) {

System.***out***.print(" -> ");

}

}

System.***out***.println();

}

}

OUTPUT:-

