import java.util.\*;

public class bankers{

public static void main(String[] args) {

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

System.***out***.println("Enter the number of Processes:");

int process = sc.nextInt();

System.***out***.println("Enter the number of Resources:");

int resources = sc.nextInt();

int[][] max = new int[process][resources];

int[][] allocation = new int[process][resources];

int[][] need = new int[process][resources];

int[] available = new int[resources];

System.***out***.println("Enter the Max Matrix:");

for (int i = 0; i < process; i++) {

for (int j = 0; j < resources; j++) {

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

}

}

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

for (int i = 0; i < process; i++) {

for (int j = 0; j < resources; j++) {

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

}

}

for (int i = 0; i < process; i++) {

for (int j = 0; j < resources; j++) {

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

}

}

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

for (int j = 0; j < resources; j++) {

available[j] = sc.nextInt();

}

if (*isSafe*(process, resources, allocation, need, available)) {

System.***out***.println("System is in a safe state.");

} else {

System.***out***.println("System is in a deadlock state.");

}

sc.close();

}

private static boolean isSafe(int process, int resources, int[][] allocation, int[][] need, int[] available) {

boolean[] finish = new boolean[process];

int[] safeSequence = new int[process];

int count = 0;

while (count < process) {

boolean found = false;

for (int p = 0; p < process; p++) {

if (!finish[p]) {

int j;

for (j = 0; j < resources; j++) {

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

break;

}

}

if (j == resources) {

for (int k = 0; k < resources; k++) {

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

}

safeSequence[count++] = p;

finish[p] = true;

found = true;

}

}

}

if (!found) {

return false;

}

}

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

for (int i = 0; i < process; i++) {

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

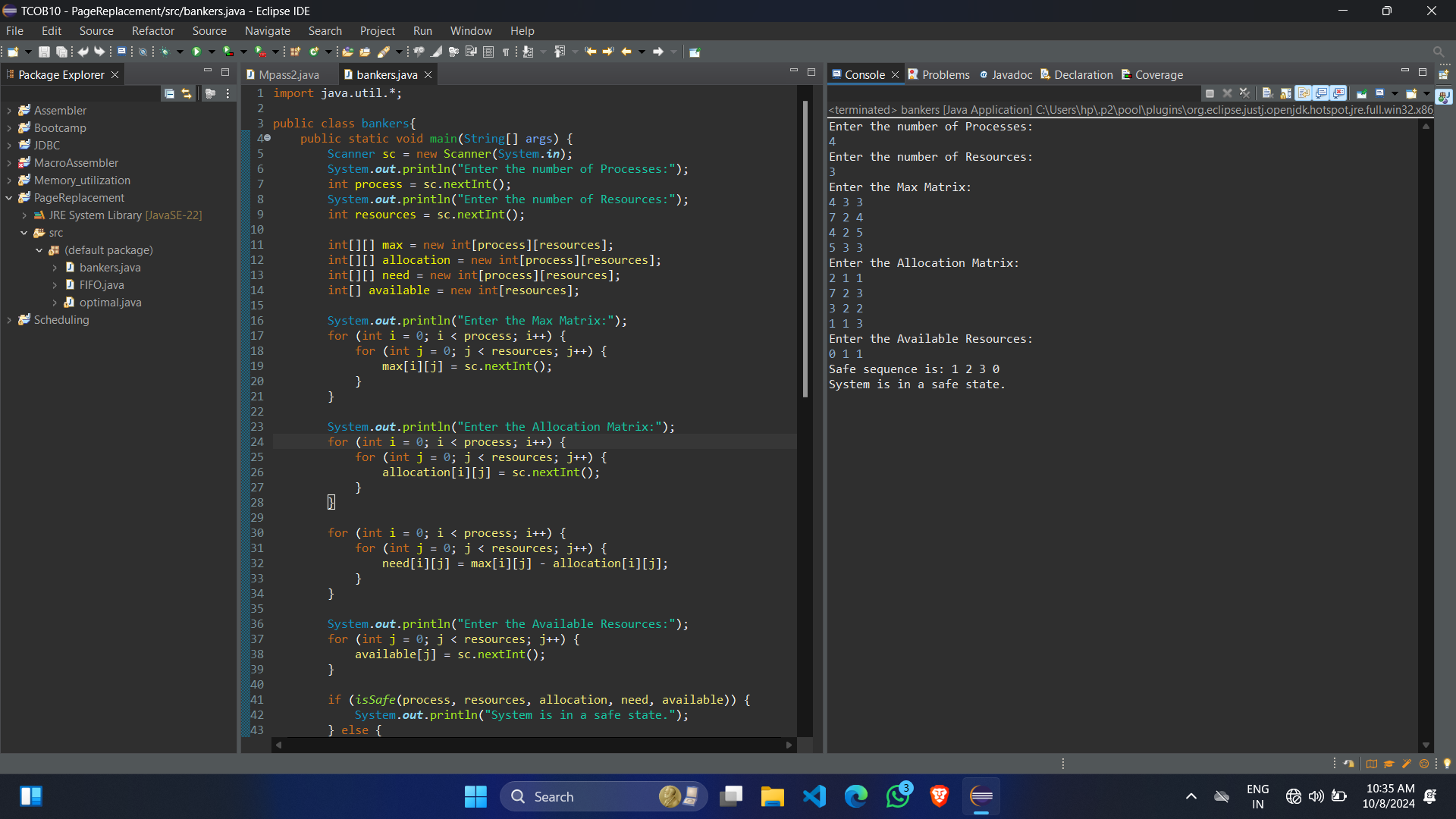
}

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

return true;

}

}

**OUTPUT**:-