**CODE :**

import java.util.Scanner;

public class BankersAlgorithm {

    private int need[][], allocate[][], max[][], available[][], np, nr;

    private void input() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter no. of processes and no. of resources: ");

        np = sc.nextInt();  // number of processes

        nr = sc.nextInt();  // number of resources

        need = new int[np][nr];  // initializing arrays

        max = new int[np][nr];

        allocate = new int[np][nr];

        available = new int[1][nr];

        System.out.println("Enter allocation matrix: ");

        for (int i = 0; i < np; i++)

            for (int j = 0; j < nr; j++)

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

        System.out.println("Enter max matrix: ");

        for (int i = 0; i < np; i++)

            for (int j = 0; j < nr; j++)

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

        System.out.println("Enter available matrix: ");

        for (int j = 0; j < nr; j++)

            available[0][j] = sc.nextInt();  // available matrix

        sc.close();

    }

    private int[][] calc\_need() {

        for (int i = 0; i < np; i++)

            for (int j = 0; j < nr; j++)  // calculating need matrix

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

        return need;

    }

    private boolean check(int i) {

        for (int j = 0; j < nr; j++)

            if (available[0][j] < need[i][j])

                return false;

        return true;

    }

    public void isSafe() {

        input();  // take user input

        calc\_need();  // calculate the need matrix

        boolean[] visited = new boolean[np];

        int[] safeSequence = new int[np];

        int index = 0;

        while (index < np) {  // until all processes allocated

            boolean flag = false;

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

                if (!visited[i] && check(i)) {  // trying to allocate

                    visited[i] = true;

                    safeSequence[index++] = i;

                    flag = true;

                    for (int j = 0; j < nr; j++)

                        available[0][j] = available[0][j] - need[i][j] + max[i][j];

                }

            }

            if (!flag) break;  // if no allocation

        }

        if (index == np) {  // if all processes are allocated

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

            System.out.println("Safe sequence is:");

            for (int i = 0; i < np; i++)

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

        } else {

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

        }

    }

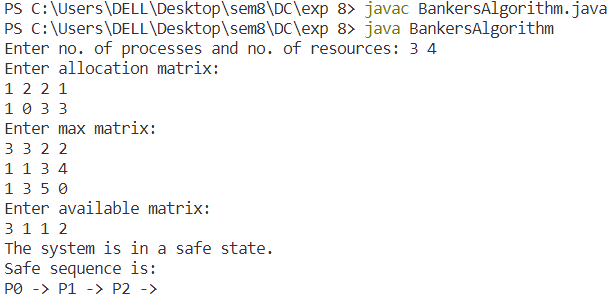
    public static void main(String[] args) {

        new BankersAlgorithm().isSafe();

    }

}

**OUTPUT :**

****

