**Code:**

import java.util.\*;

public class bankers {

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

Scanner sc=new Scanner(System.in);

public void Input()

{

System.out.println("Enter number of processes:");

np=sc.nextInt();

System.out.println("Enter number of resources:");

nr=sc.nextInt();

need=new int[np][nr];

max=new int[np][nr];

allocation=new int[np][nr];

available=new int[1][nr];

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

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

{

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

{

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

}

}

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

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

{

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

{

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

}

}

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

for(int j=0;j<1;j++)

{

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

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

}

}

}

void need()

{

System.out.println("Need Matrix:");

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

{

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

{

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

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

}

System.out.println();

}

}

private boolean allocation(int row) {

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

if(available[0][i]<need[row][i]) {

return false;

}

}

return true;

}

public void Safe() {

int p = 0;

Input();

need();

boolean done[] = new boolean[np];

while ( p < np) {

boolean allocated = false;

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

if(!done[i] && allocation(i)) {

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

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

}

System.out.println("Process allocated: " + (i+1));

allocated = done[i] = true;

p++;

}

}

if(!allocated) {

break;

}

}

if(p == np) {

System.out.println("\nSafely Allocated.\n");

}

else {

System.out.println("Not allocated safely.");

}

}

public static void main(String args[]) {

bankers b=new bankers();

b.Safe();

}

}

**Output:**

