import java.util.\*;

import java.io.\*;

public class Assignment06\_BankersAlgorithm {

public static void main(String[] args) throws Exception{

BufferedReader br = new BufferedReader(new FileReader("F:\\SEMESTER\\Fall- 21\\CSE321\\Lab\\Lab07\\input.txt"));

int row = Integer.parseInt(br.readLine());

//System.out.println("Number of process "+ row);

int column = Integer.parseInt(br.readLine());

//System.out.println("Number of resources "+ column);

String [] process = new String[row];

int[][] max = new int[row ][column];

int[][] allocation = new int[row ][column];

int[][] need = new int[row][column];

int[][] available = new int[row+1][column];

LinkedList<Integer> track = new LinkedList <Integer> ();

String s3 = br.readLine();

StringTokenizer st3 = new StringTokenizer(s3, " ");

int n = 0;

while(st3.hasMoreTokens()){

process[n]=st3.nextToken();

//System.out.println("process "+process[n]);

n++;

}

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

String s=br.readLine();

StringTokenizer st = new StringTokenizer(s," ");

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

max[i][j] = Integer.parseInt(st.nextToken());

//System.out.println("Max: "+max[i][j]);

}

}

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

String s=br.readLine();

StringTokenizer st = new StringTokenizer(s, " ");

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

allocation[i][j]=Integer.parseInt(st.nextToken());

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

}

}

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

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

System.out.println();

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

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

}

}

System.out.println();

String s = br.readLine();

StringTokenizer st = new StringTokenizer(s, " ");

int counter = 0;

while(st.hasMoreTokens()){

available[0][counter]=Integer.parseInt(st.nextToken());

counter++;

}

counter = 0;

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

i = i%row;

boolean flag=true;

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

if(need[i][j]<=available[counter][j]){

}

else{

flag = false;

break;

}

if(flag && j==(column-1) && !track.contains(i)){

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

available[counter+1][k]=available[counter][k]+allocation[i][k];

}

track.addLast(i);

counter++;

}

}

if(track.size()==row){

break;

}

}

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

for(int i=0; i<track.size(); i++){

System.out.print(process[track.get(i)]+" ");

}

System.out.println();

System.out.print("Change in available resource matrix : ");

for(int i=1; i<available.length; i++){

System.out.println();

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

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

}

}

System.out.println();

}

}