```
//Gaurav Rai 1706019

#include<bits/stdc++.h>
using namespace std;
bool lessthan(int Need[],int Available[],bool vis,int m)
{
    for(int i=0;i<m;i++)
    {
        if(Need[i]>Available[i])
        return false;
    }
    return true;
}
int main()
{
    int n=0,m=0;
    cout<<"Enter the number of Processes :";cin>>n;
    cout<<"Enter the number of Resources : ";cin>>m;
    int Available[m+1];
    cout<<"\n\tAvailable Number of Resources of : \n";
    for(int i=0;i<m;i++)
        {cout<<"R"<<i+1<<" : ";cin>>Available[i];}
    cout<<" Allocation Matrix\n";
    int Allocation[n+1][m+1];
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            cin>>Allocation[i][j];
    cout<<" Max Req. Matrix\n";
    int Max[n+1][m+1];
    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
            cin>>Max[i][j];
    cout<<"\tNeed Matrix\n";
    int Need[n+1][m+1];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<m;j++)
        {
            Need[i][j]=Max[i][j]-Allocation[i][j];cout<<Need[i][j]<<" ";
        }
            cout<<endl;
    }
    bool vis[n+1];
    memset(vis,false,sizeof(vis));
    vector<int> safeSeq;
    bool flag1=true;
    while(safeSeq.size()<n && flag1 )
    {
        flag1=false;
        for(int i=0;i<n ;i++)
```

```
        {
            if(vis[i]==false && lessthan(Need[i],Available,vis,m))
            {
                flag1=true;
                vis[i]=true;
                safeSeq.push_back(i);
                for(int j=0;j<m;j++)
                Available[j]+=Allocation[i][j];
            }
        }
    }
    if(safeSeq.size()!=n)
    {cout<<"Deadlock !!! \n No SafeSequence Exists\n";
    for(int i=0;i<safeSeq.size() ;i++)
        {cout<<"->"<<"P"<<safeSeq[i];}
        cout<<endl;}
    else
    {
        cout<<"Safe Sequence :";
        for(int i=0;i<n ;i++)
        {cout<<"->"<<"P"<<safeSeq[i];}
        cout<<endl;
    }
    return 0;
}
```