//Bankers Algorithm for Dead Lock Prevention

//Sanket Sharma CS-D 2015bcs2035

#include<stdio.h>

void main()

{

int process,resource,i,j,instanc,k=0,count1=0,count2=0;

system("COLOR F0");

printf("\n\t Enter No. of Process:-\n");

printf("\t\t");

scanf("%d",&process);

printf("\n\tEnter No. of Resources:-\n");

printf("\t\t");

scanf("%d",&resource);

int avail[resource],max[process][resource],allot[process][resource],need[process][resource],completed[process];

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

completed[i]=0;

printf("\n\tEnter No. of Available Instances\n");

for(i=0;i<resource;i++)

{

printf("\t\t");

scanf("%d",&instanc);

avail[i]=instanc;

}

printf("\n\tEnter Maximum No. of instances of resources that a Process need:\n");

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

{

printf("\n\t For P[%d]",i);

for(j=0;j<resource;j++)

{

printf("\t");

scanf("%d",&instanc);

max[i][j]=instanc;

}

}

printf("\n\t Enter no. of instances already allocated to process of a resource:\n");

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

{

printf("\n\t For P[%d]\t",i);

for(j=0;j<resource;j++)

{

printf("\t\t");

scanf("%d",&instanc);

allot[i][j]=instanc;

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

}

}

printf("\n\t Safe Sequence is:- \t");

while(count1!=process)

{

count2=count1;

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

{

for(j=0;j<resource;j++)

{

if(need[i][j]<=avail[j])

{

k++;

}

}

if(k==resource && completed[i]==0 )

{

printf("P[%d]\t",i);

completed[i]=1;

for(j=0;j<resource;j++)

{

avail[j]=avail[j]+allot[i][j];

}

count1++; }

k=0; }

if(count1==count2)

{

printf("\t\t Stop ..After this.....Deadlock \n");

break;

}

}

}



