#include<stdio.h>

#include<conio.h>

#include<iostream.h>

#include<stdlib.h>

#include<graphics.h>

#include<string.h>

#include<dos.h>

void main(){

int s[4][9],n,m=0;

int i,j,next=0,step=0;

int msg[10][4]={0},totmsg;

char op;

int pi,pj,ei,ej;

clrscr();

cout<<"\nProgram for Lamport Logical Clock";

cout<<"\nEnter Number Of Process ";

cin>>n;

for(i=0;i<n;i++){

cout<<"\nEnter number of STATES of process P"<<i<<" ";

cin>>s[i][8];

for(j=1;j<=s[i][8];j++){

s[i][j]=j;

}

}

do{

cout<<"\nEnter message transit";

cout<<"\nFROM ->\nEnter Process Number P";

cin>>msg[m][0];

cout<<"\nEnter Event Number e";

cin>>msg[m][1];

cout<<"\nTO ->\nEnter Process Number P";

cin>>msg[m][2];

cout<<"\nEnter Event Number e";

cin>>msg[m][3];

cout<<"\n\nPress 'y' to continue";

op=getch();

cout<<op;

m++;

totmsg=m;

}while(op=='y');

m=0;

for (i=0;i<totmsg;i++){

pi=msg[i][0];

ei=msg[i][1];

pj=msg[i][2];

ej=msg[i][3];

if(s[pj][ej]< (s[pi][ei]+1)){

s[pj][ej]=s[pi][ei]+1;

for (j=ej+1;j<=s[pj][8];j++){

s[pj][j]=s[pj][j-1]+1;

}

}

}

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TC\\BGI");

outtextxy(200,15,"Program For Lamport Logical Clock");

//drawing process and events

for(i=0;i<n;i++){

char\* p1;

itoa(i,p1,10);

outtextxy(5,100+next,"P");

outtextxy(13,100+next,p1);

line(100,100+next,600,100+next);

for(j=1;j<=s[i][8];j++){

char\* p2;

itoa(j,p2,10);

outtextxy(100+step,90+next,"e");

outtextxy(110+step,90+next,p2);

//timestamp

char\* p3;

itoa(s[i][j]-1,p3,10);

outtextxy(100+step,110+next,"t");

outtextxy(110+step,110+next,p3);

circle(105+step,100+next,5);

step+=50;

}

step=0;

next+=100;

}

delay(2000);

//drawing message transit

for(m=0;m<totmsg;m++){

setlinestyle(SOLID\_LINE,1,3);

setcolor(m+4);

line(msg[m][1]\*50+50,msg[m][0]\*100+100,msg[m][3]\*50+50,msg[m][2]\*100+100);

if (msg[m][2]>msg[m][0]){

line(msg[m][3]\*50+50,msg[m][2]\*100+100,msg[m][3]\*50+50,msg[m][2]\*100+90);

line(msg[m][3]\*50+50,msg[m][2]\*100+100,msg[m][3]\*50+40,msg[m][2]\*100+90);

}

else{

line(msg[m][3]\*50+50,msg[m][2]\*100+100,msg[m][3]\*50+50,msg[m][2]\*100+110);

line(msg[m][3]\*50+50,msg[m][2]\*100+100,msg[m][3]\*50+40,msg[m][2]\*100+110);

}

}

getch();

}