

Simulate the functioning of Lamport's Logical Clock in C

Code:

```
#include<stdio.h>

int max(int a,int b);
int main()
{
    int i,j,k,p1[20],p2[20],e1,e2,dep[20][20];
    printf("*** Lamport's Logical Clock ***\n");
    printf("Enter the events : ");
    scanf("%d %d",&e1,&e2);
    for(i=0;i<e1;i++)
        p1[i]=i+1;
    for(i=0;i<e2;i++)
        p2[i]=i+1;
    printf("Enter the Dependency matrix:\n");
    printf("\nEnter 1 if E1->E2 \nEnter -1, if E2->E1 \nElse Enter 0 \n\n");
    printf(" ");
    for(i=0;i<e2;i++)
        printf(" e2%d",i+1);
    for(i=0;i<e1;i++){
        printf("\ne1%d ",i+1);
        for(j=0;j<e2;j++){
            scanf("%d",&dep[i][j]);
        }
    }

    for(i=0;i<e1;i++){
        for(j=0;j<e2;j++){

            if(dep[i][j]==1){
                p2[j]=max(p2[j],p1[i]+1);
                for(k=j;k<e2;k++)
                    p2[k+1]=p2[k]+1;
            }

            if(dep[i][j]==-1){
                p1[i]=max(p1[i],p2[j]+1);
                for(k=i;k<e1;k++)
                    p2[k+1]=p1[k]+1;
            }
        }
    }

    printf("\nP1 : ");
    for(i=0;i<e1;i++){
        printf("%d ",p1[i]);
    }
    printf("\nP2 : ");
    for(j=0;j<e2;j++)
        printf("%d ",p2[j]);
    printf("\n");
    return 0 ;
}

int max(int a, int b)
{
    if (a>b)
```

```
    return a;
else
    return b;
}
```

Output:

```
(base) Shivs-Air:Distributed Computing championballer$ gcc Lamport.c -o lamport ]
(base) Shivs-Air:Distributed Computing championballer$ ./lamport ]
*** Lamport's Logical Clock ***
Enter the events : 2 4
Enter the Dependency matrix:

Enter 1 if E1->E2
Enter -1, if E2->E1
Else Enter 0

    e21 e22 e23 e24
e11 0   0   1  -1

e12 1   1   0   1

P1 : 5 2
P2 : 3 4 5 6
```