

**// DEADLOCK DETECTION**

```
#include<stdio.h>
int main()
{
    int n,p,i,x,y=0,j,k=0,u=0;
    printf("enter the no of resources");
    scanf("%d",&n);
    printf("enter no of processes");
    scanf("%d",&p);
    int a[p][n],m[p][n],r[p][n],w[n];
    int f[p],t[p];
    printf("enter instances for each resource");
    for(i=0;i<n;i++)
        scanf("%d",&w[i]);
    printf("enter the allocation values\n");
    for(i=0;i<p;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<n;i++)
    {
        x=0;
        for(j=0;j<p;j++)
        {
            x=x+a[j][i];
        }
        if(x>w[i])
        {
            y=3;
        }
        else
            w[i]=w[i]-x;
    }
    if(y==0)
    {

        printf("enter max matrix\n");
        for(i=0;i<p;i++)
        {
            for(j=0;j<n;j++)
            {
                scanf("%d",&m[i][j]);
                if(m[i][j]<a[i][j])
                    y=2;
            }
        }
        if(y==0)
        {
```

```

printf("allocation matrix\n");
for(i=0;i<p;i++)
{
    for(j=0;j<n;j++)
    {
        printf("  %d",a[i][j]);
    }
    printf("\n");
}
printf("\nmax matrix\n");
for(i=0;i<p;i++)
{
    for(j=0;j<n;j++)
    {
        printf("  %d",m[i][j]);
    }
    printf("\n");
}
for(i=0;i<p;i++)
{
    for(j=0;j<n;j++)
        r[i][j]=m[i][j]-a[i][j];
}
printf("\nrequest matrix\n");
for(i=0;i<p;i++)
{
    for(j=0;j<n;j++)
    {
        printf("  %d",r[i][j]);
    }
    printf("\n");
}
for(i=0;i<p;i++)
    f[i]=0;
for(i=0;i<p;i++)
{
    k=0;
    for(j=0;j<n;j++)
    {
        if(a[i][j]==0)
            k++;
    }
    if(k==n)
        f[i]=1;
}
for(i=0;i<p;i++)
{
    if(f[i]==0)
    {
        x=0;
        for(j=0;j<n;j++)
        {

```

```

        if (r[i][j] <= w[j])
            x++;
    }
    if (x == n)
    {
        for (j = 0; j < n; j++)
        {
            w[j] = w[j] + a[i][j];
        }
        f[i] = 1;
        t[u] = i;
        i = 0;
        k++;
        u++;
    }
}
if (k == p)
    break;
}
k = 0;
for (i = 0; i < p; i++)
    if (f[i] == 1)
        k++;
printf("k      %d\n", k);
if (k == p)
{
    printf("not a deadlock\n");
    for (i = 0; i < p; i++)
        printf("p%d->", i);
}
else
{
    printf("deadlock\n");
    for (i = 0; i < p; i++)
    {
        k = 0;
        for (j = 0; j < u; j++)
        {
            if (t[j] != i)
                k++;
            if (k == u)
            {
                printf("p%d is causing deadlock", i);
                x = 3;
                break;
            }
        }
    }
    if (x == 3)
        break;
}
}
}

```

Pritee Parwekar

```
        }
        else
            printf("invalid");
    }
    else
        printf("invalid input");
    return 0;
}
```

Pritee Parwekar