PROGRAM 8
Write a C program to simulate deadlock detection

```
#include<stdio.h>
void main()
{
   int n,m,i,j;
   printf("Enter the number of processes and number of types of resources:\n");
   scanf("%d %d",&n,&m);
   int max[n][m],need[n][m],all[n][m],ava[m],flag=1,finish[n],dead[n],c=0;
   printf("Enter the maximum number of each type of resource needed by each process:\n");
   for(i=0;i<n;i++)
   {
      for(j=0;j<m;j++)
      {
         scanf("%d",&max[i][j]);
      }
   }
   printf("Enter the allocated number of each type of resource needed by each process:\n");
   for(i=0;i<n;i++)
   {
      for(j=0;j<m;j++)
      {
         scanf("%d",&all[i][j]);
      }
   }
   printf("Enter the available number of each type of resource:\n");
   for(j=0;j<m;j++)
   {
      scanf("%d",&ava[j]);
   }
   for(i=0;i<n;i++)
   {
      for(j=0;j<m;j++)
      {
         need[i][j]=max[i][j]-all[i][j];
      }
   }
   for(i=0;i<n;i++)
   {
      finish[i]=0;
   }
   while(flag)
```

```
{
   flag=0;
   for(i=0;i<n;i++)
   {
      c=0;
      for(j=0;j<m;j++)
      {
         if(finish[i]==0 && need[i][j]<=ava[j])
         {
            c++;
            if(c==m)
            {
               for(j=0;j<m;j++)
               {
                  ava[j]+=all[i][j];
                  finish[i]=1;
                  flag=1;
               }
               if(finish[i]==1)
               {
                  i=n;
               }
            }
         }
      }
   }
}
j=0;
flag=0;
for(i=0;i<n;i++)
{
   if(finish[i]==0)
   {
      dead[j]=i;
      j++;
      flag=1;
   }
}
if(flag==1)
{
   printf("Deadlock has occured:\n");
   printf("The deadlock processes are:\n");
   for(i=0;i<n;i++)
   {
```

```
            printf("P%d ",dead[i]);
        }
    }
    else
    printf("No deadlock has occured!\n");
}
```

OUTPUT: