```c
#include<stdio.h>
main()
 {
   int p,r,instances,i=0,j=0,k=0,count1=0,count2=0;
   printf("\n Enter No. of Process : ");                        //collecting no: of processes
   scanf("%d",&p);
   printf("\nEnter No. of Resources : ");               //collecting no: of resources
   scanf("%d",&r);
   int available[r],max[p][r],allotted[p][r],need[p][r],completed[p];
   for(i=0;i<p;i++)
   completed[i]=0;                             //Setting Flag for uncompleted Process
   printf("\nEnter No. of Available Instances -\n");
   for(i=0;i<r;i++)
    {
     printf(" R[%d] : ",i);
     scanf("%d",&instances);
     available[i]=instances;                        // Storing Available instances
    }
   printf("\nEnter Maximum (MAX table) No. of instances of resources that a Process need -\n");
    for(i=0;i<p;i++)
     {
        printf(" P[%d] : ",i);
        for(j=0;j<r;j++)
         {
            scanf("%d",&instances);
            max[i][j]=instances;
         }
     }
    printf("\nEnter no. of instances already allocated to process of a resource -\n");
     for(i=0;i<p;i++)
     {
        printf(" P[%d] : ",i);
        for(j=0;j<r;j++)
         {
            scanf("%d",&instances);
            allotted[i][j]=instances;
            need[i][j]=max[i][j]-allotted[i][j];       //calculating Need of each process
         }
    }
    printf("\n Possible Sequence is : ");
    while(count1!=p)
    {
    count2=count1;
    for(i=0;i<p;i++)
     {
       for(j=0;j<r;j++)
        {
            if(need[i][j]<=available[j])
              {
                 k++;                              //flag for resources allocation
              }
        }
        if(k==r && completed[i]==0 )            //check if all resources fulfilled for process
         {
           printf("P[%d]\t",i);
           completed[i]=1;                      //mark process as completed
           for(j=0;j<r;j++)
             {
               available[j]=available[j]+allotted[i][j];        //return resources
             }
             count1++;                          //flag for resource availability
         }
         k=0;
       }
         if(count1==count2)                     //resources no more available
         {
         printf("...and then Deadlock occurs");
         break;
       }
 }
printf("\n\n");
}
```