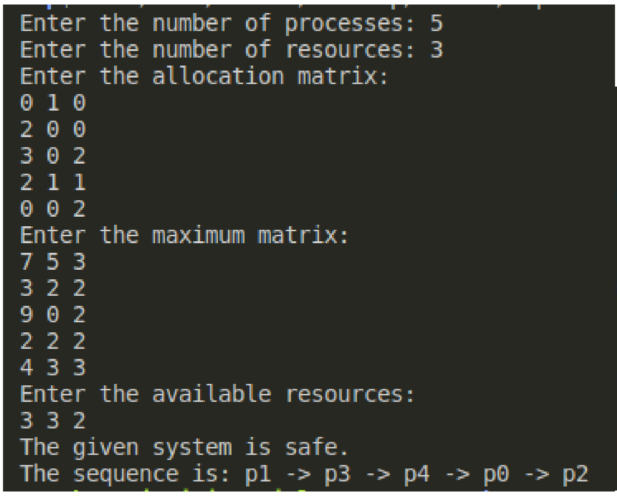
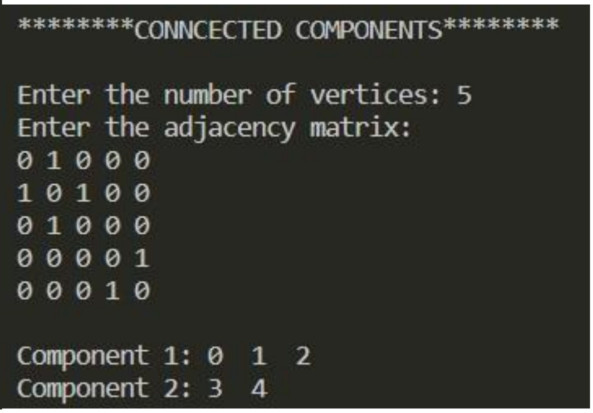


```
Enter the number of processes: 5
Enter the number of resources: 3
Enter the allocation matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the maximum matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter the available resources:
3 3 2
The given system is safe.
The sequence is: p1 -> p3 -> p4 -> p0 -> p2
```



```
********CONNCECTED COMPONENTS********

Enter the number of vertices: 5
Enter the adjacency matrix:
0 1 0 0 0
1 0 1 0 0
0 1 0 0 0
0 0 0 0 1
0 0 0 1 0

Component 1: 0  1  2
Component 2: 3  4
```

```
#include<stdio.h>

#include<stdlib.h>

#define MAX_VERTICES 30

int visited[MAX_VERTICES];

int Graph[MAX_VERTICES][MAX_VERTICES];

int n;

void InitializeGraph()

{

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

   {

      visited[i]=0;

      for(int j=0; j<n; j++)

      {

         Graph[i][j]=0;

      }

   }

}

void InputGraph()

{

   printf("Enter the adjacency matrix: \n");

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

   {

      for(int j=0; j<n; j++)
```

```
        {
            scanf("%d", &Graph[i][j]);
        }
    }
}
void DFS(int startVertext)
{
    printf("%d  ", startVertext);
    visited[startVertext]=1;
    for (int j = 0; j < n; j++)
    {
        if(Graph[startVertext][j]==1 && !visited[j])
        {
            DFS(j);
        }
    }
}
void main()
{
    printf("\nCONNCECTED COMPONENTS\n");
    printf("\nEnter the number of vertices: ");
    scanf("%d",&n);
    InitializeGraph();
    InputGraph();
    int componentCount = 1;
    for(int i=0; i<n; i++)
    {
        if(!visited[i])
        {
            printf("\nComponent %d: ",componentCount);
            componentCount++;
        }
    }
```