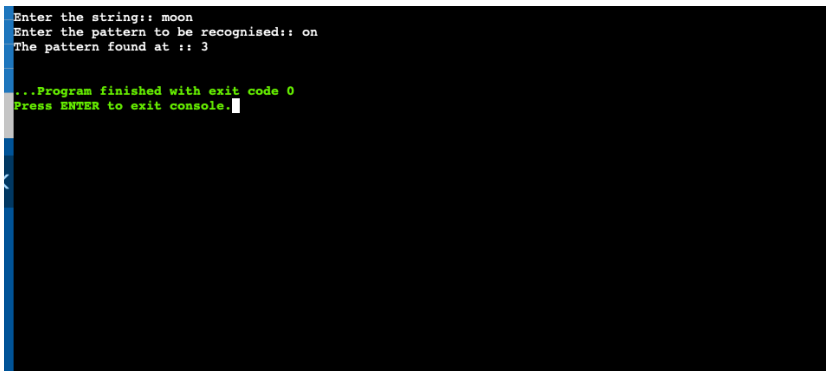
LAKSHMI S KUMAR
1BM19CS078
LAB-4

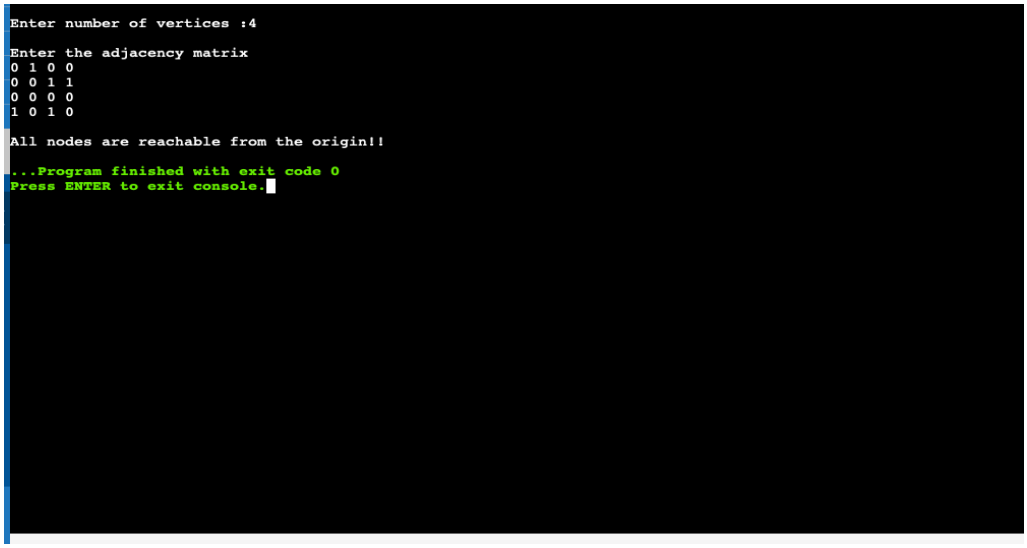
```
#include<stdio.h>
#include<string.h>
char str[50],pattern[25];
int n,m,i,j;
int main()
{
   int pos;
   printf("Enter the string:: ");
   scanf("%s",str);
   printf("Enter the pattern to be recognised:: ");
   scanf("%s",pattern);
   pos = pat_find();
   if(pos==-1)
       printf("Pattern not found!");
   else
       printf("The pattern found at :: %d\n",pos+1);
}
int pat_find()
{
   n = strlen(str);
   m = strlen(pattern);
   for(i=0;i<=n-m;i++)
   {
```

```
        j=0;
        while(j<m && str[i+j]==pattern[j])
            j++;
        if(j==m)
            return i;
    }
    return -1;

}
```




```
#include<stdio.h>

void main()
{
int a[20][20],n,i,j,st[20],tot=1,top=-1,r[20],flag;
printf("\nEnter number of vertices :");
scanf("%d",&n);
printf("\nEnter the adjacency matrix\n");
for(i=0;i<n;i++)
{
r[i]=0;
for(j=0;j<n;j++)
{
scanf("%d",&a[i][j]);
}
```

```
}
st[++top]=0;
r[0]=1;
while(top!=-1)
{
flag=0;
for(j=0;j<n;j++)
{
if(r[j]==0&&a[st[top]][j]==1)
{
st[++top]=j;
tot++;
r[j]=1;
flag=1;
break;
}
}
if(flag==0)
{
top=top-1;
}
}
if(tot==n)
{
printf("\nAll nodes are reachable from the origin!!");
}
else
{
printf("\nAll nodes are not reachable from the
origin!!");
}
}
```




```
#include<stdio.h
>
                    void dfs(int);
                    int a[10][10],vis[10],n,flag=0;
                    void main()
                    {
                      int i,j,src;
                      printf("Enter number of vertices:: \n");
                      scanf("%d",&n);
                      printf("Enter adjacency matrix:: \n");
                      for(i=1;i<=n;i++)
                      {
                        for(j=1;j<=n;j++)
                        {
                          scanf("%d", &a[i][j]);
                        }
                     }
                    for(i=1;i<=n;i++)
                    vis[i]=0;
                    printf("Enter source vertex\n");
                    scanf("%d",&src);
```

```
dfs(src);


for(i=1;i<=n;i++)
{
   if(vis[i] == 0)
   {
        printf("Graph not connected\n");
        flag=1;
        break;
   }
}
if(flag==0)
printf("Graph connected\n");
}
void dfs(int v)
{
int i;
vis[v]=1;


  for(i=1;i<=n;i++)
  {
       if(a[v][i]==1 && vis[i]==0)
      dfs(i);
  }
}
```

```
Enter number of vertices::
5
Enter adjacency matrix::
0 1 0 1 0
1 0 1 0 0
0 1 0 0 0
1 0 0 0 0
0 0 0 0 0
Enter source vertex
1
Graph not connected

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter number of vertices::
5
Enter adjacency matrix::
0 1 0 1 0
1 0 1 0 0
0 1 0 0 0
1 0 0 0 1
0 0 0 1 0
Enter source vertex
1
Graph connected

...Program finished with exit code 0
Press ENTER to exit console.
```