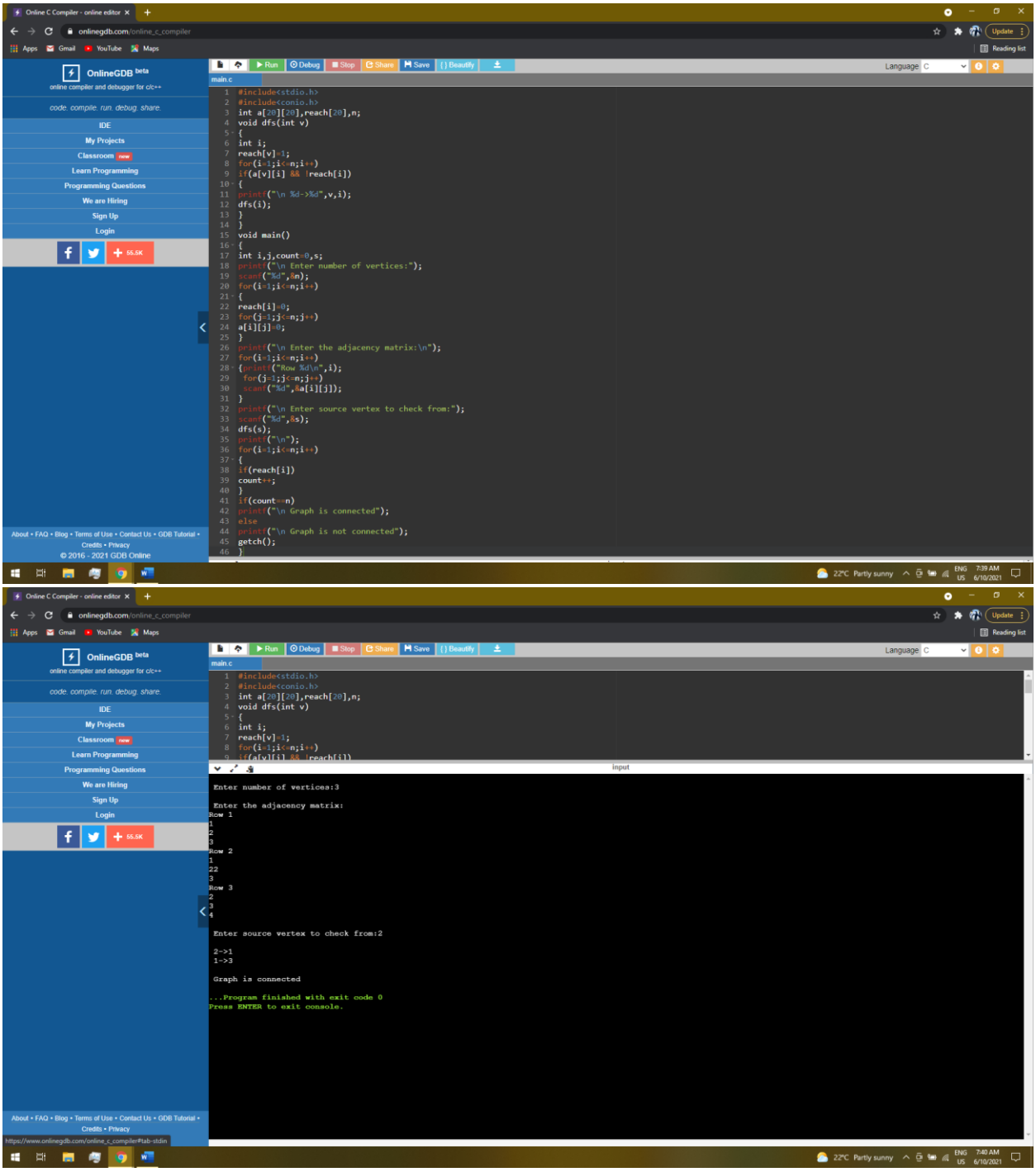
OUTPUT FOR DFS



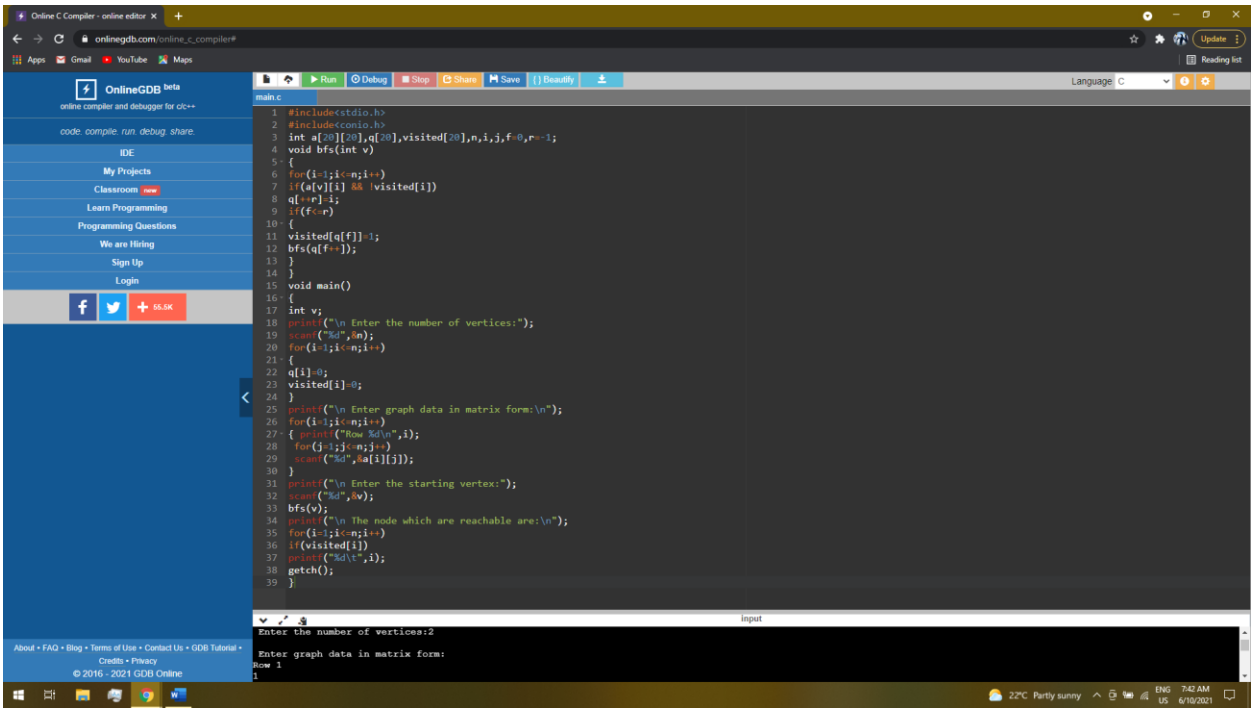
CODE FOR DFS

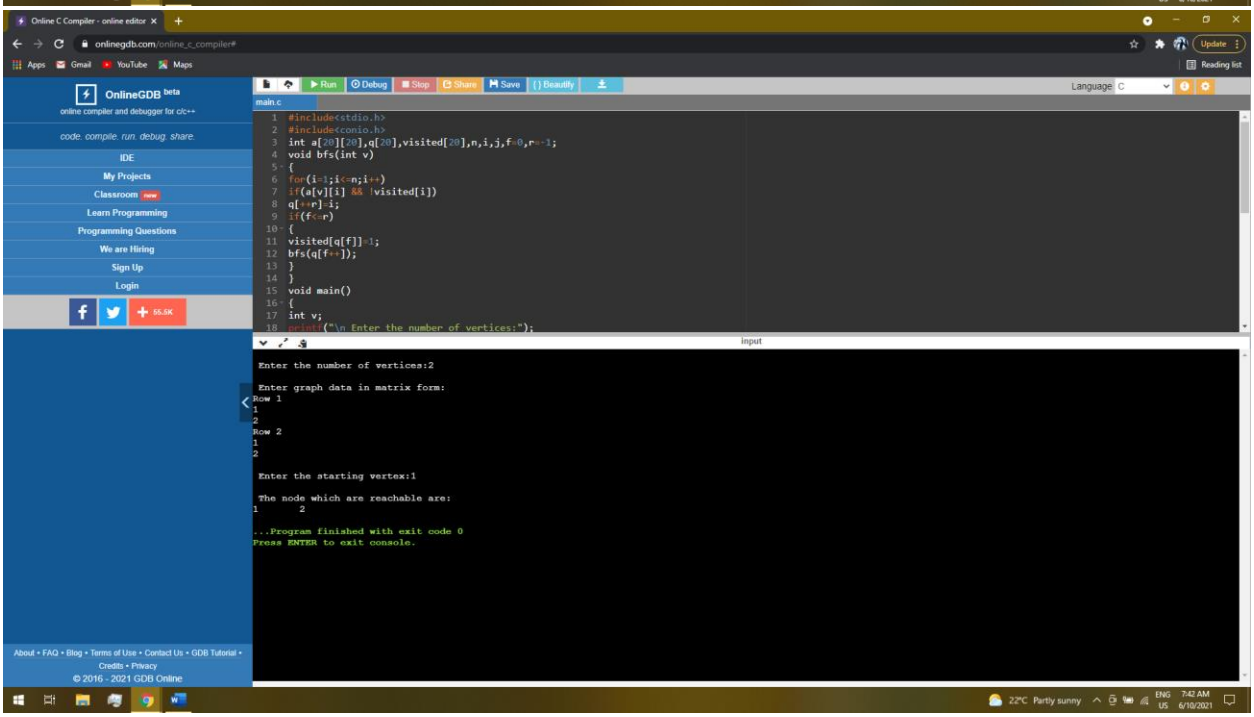
```
#include<stdio.h>

#include<conio.h>

int a[20][20],reach[20],n;
```

```
void dfs(int v)
{
int i;
reach[v]=1;
for(i=1;i<=n;i++)
if(a[v][i] && !reach[i])
{
printf("\n %d->%d",v,i);
dfs(i);
}
}
void main()
{
int i,j,count=0,s;
printf("\n Enter number of vertices:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
reach[i]=0;
for(j=1;j<=n;j++)
a[i][j]=0;}
printf("\n Enter the adjacency matrix:\n");
for(i=1;i<=n;i++)
{printf("Row %d\n",i);
 for(j=1;j<=n;j++)
 scanf("%d",&a[i][j]);}
printf("\n Enter source vertex to check from:");
scanf("%d",&s);
dfs(s);
```

```
printf("\n");
for(i=1;i<=n;i++){
if(reach[i])
count++;}
if(count==n)
printf("\n Graph is connected");
else
printf("\n Graph is not connected");
getch();}
```

OUTPUT FOR BFS





CODE FOR BFS

```
#include<stdio.h>

#include<conio.h>

int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
```

```
void bfs(int v)
{
for(i=1;i<=n;i++)
if(a[v][i] && !visited[i])
q[++r]=i;
if(f<=r)
{
visited[q[f]]=1;
bfs(q[f++]);
}
}
void main()
{
int v;
printf("\n Enter the number of vertices:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
q[i]=0;
visited[i]=0;
}
printf("\n Enter graph data in matrix form:\n");
for(i=1;i<=n;i++)
{ printf("Row %d\n",i);
 for(j=1;j<=n;j++)
 scanf("%d",&a[i][j]);
}
printf("\n Enter the starting vertex:");
scanf("%d",&v);
```

```
bfs(v);
printf("\n The node which are reachable are:\n");
for(i=1;i<=n;i++)
if(visited[i])
printf("%d\t",i);
getch();
}
```