Vaidehi M Godbole
Roll no 14

Experiment no 7

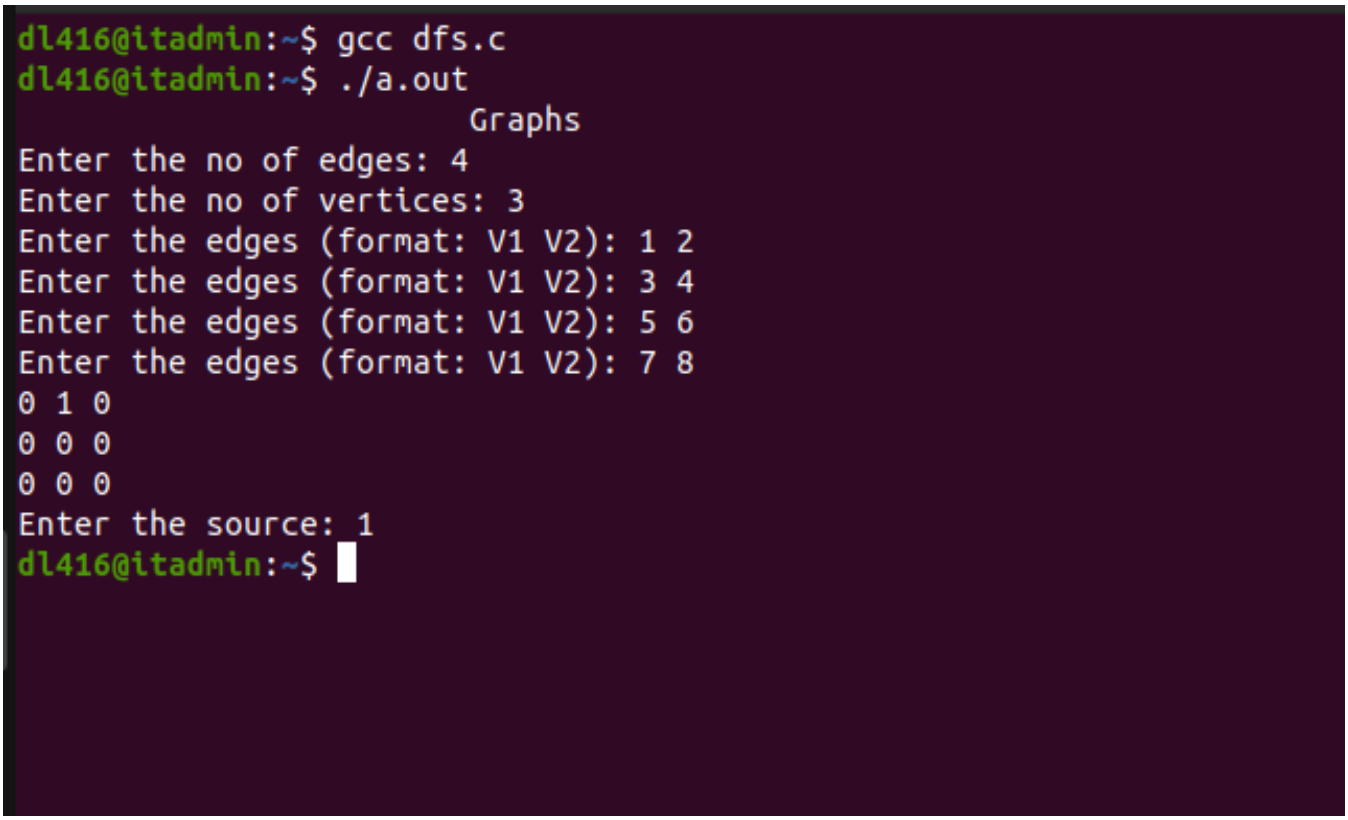
- DFS

  Code:

```
#include<stdio.h>
#include<stdlib.h>
int source,V,E,time,visited[20],G[20][20];
void DFS(int i)
{
int j;
visited[i]=1;
printf("%d->",i+1);
for(j=0;j<V;j++);
{
if(G[i][j]==1&&visited[j]==0)
DFS(j);
}
}
int main()
{
int i,j,v1, v2;
printf("\t\t\tGraphs\n");
printf("Enter the no of edges: ");
scanf("%d", &E);
printf("Enter the no of vertices: ");
scanf("%d", &V);
for(i=0;i<V;i++)
{
for(j=0;j<V;j++)
G[i][j]=0;
}
for(i=0;i<E;i++)
{
printf("Enter the edges (format: V1 V2): ");
scanf("%d %d", &v1,&v2);
G[v1-1][v2-1]=1;
}
for(i=0;i<V;i++)
```

```
{
for(j=0;j<V; j++)
{printf("%d ",G[i][j]);
}printf("\n");
}
printf("Enter the source: ");
scanf("%d", &source);
DFS(source-1);
return 0;
}
```

Output:

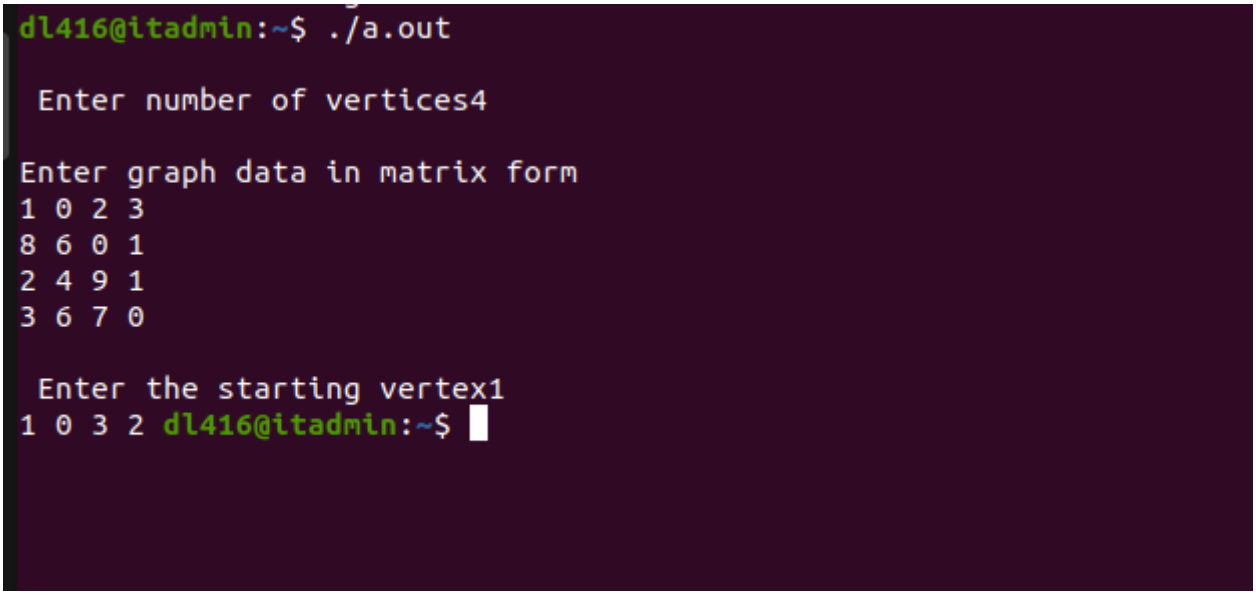


- BFS:

Code:

```
#include <stdio.h>
int a[20][20], q[20], visited[20], n, f = -1, r = -1;
void bfs(int v) {
int i;
for (i = 0; i < n; i++) {
if (a[v][i] != 0 && visited[i] == 0) {
r = r + 1;
q[r] = i;
visited[i] = 1;
printf("%d ", i);
}
}
f = f + 1;
if (f <= r)
bfs(q[f]);
}
int main() {
int v, i, j;
printf("\n Enter number of vertices");
scanf("%d", &n);
for (i = 0; i < n; i++) {
visited[i] = 0;
}
printf("\nEnter graph data in matrix form\n");
for (i = 0; i < n; i++) {
for (j = 0; j < n; j++) {
scanf("%d", &a[i][j]);
}
}
printf("\n Enter the starting vertex");
scanf("%d", &v);
f = r = 0;
q[r] = v;
visited[v] = 1;
printf("%d ", v);
bfs(v);
if (r != n - 1) {
printf("\nBFS not possible\n");
}
return 0;
}
```

Output:


dl416@itadmin:~$ ./a.out
Enter number of vertices4
Enter graph data in matrix form
1 0 2 3
8 6 0 1
2 4 9 1
3 6 7 0
Enter the starting vertex1
1 0 3 2 dl416@itadmin:~$