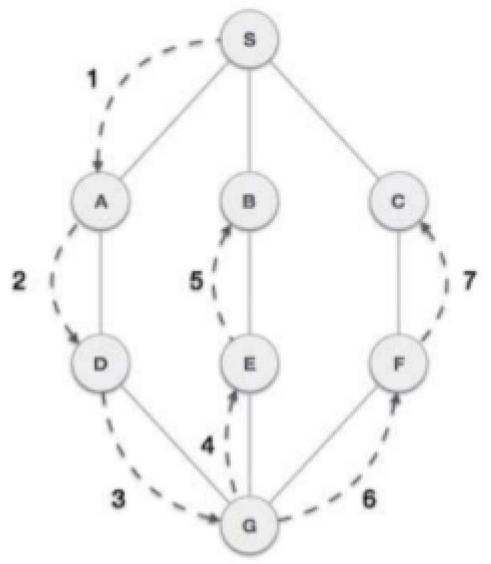
Program statement: Write programs in C to implement the following algorithms:
1) Depth First Search and

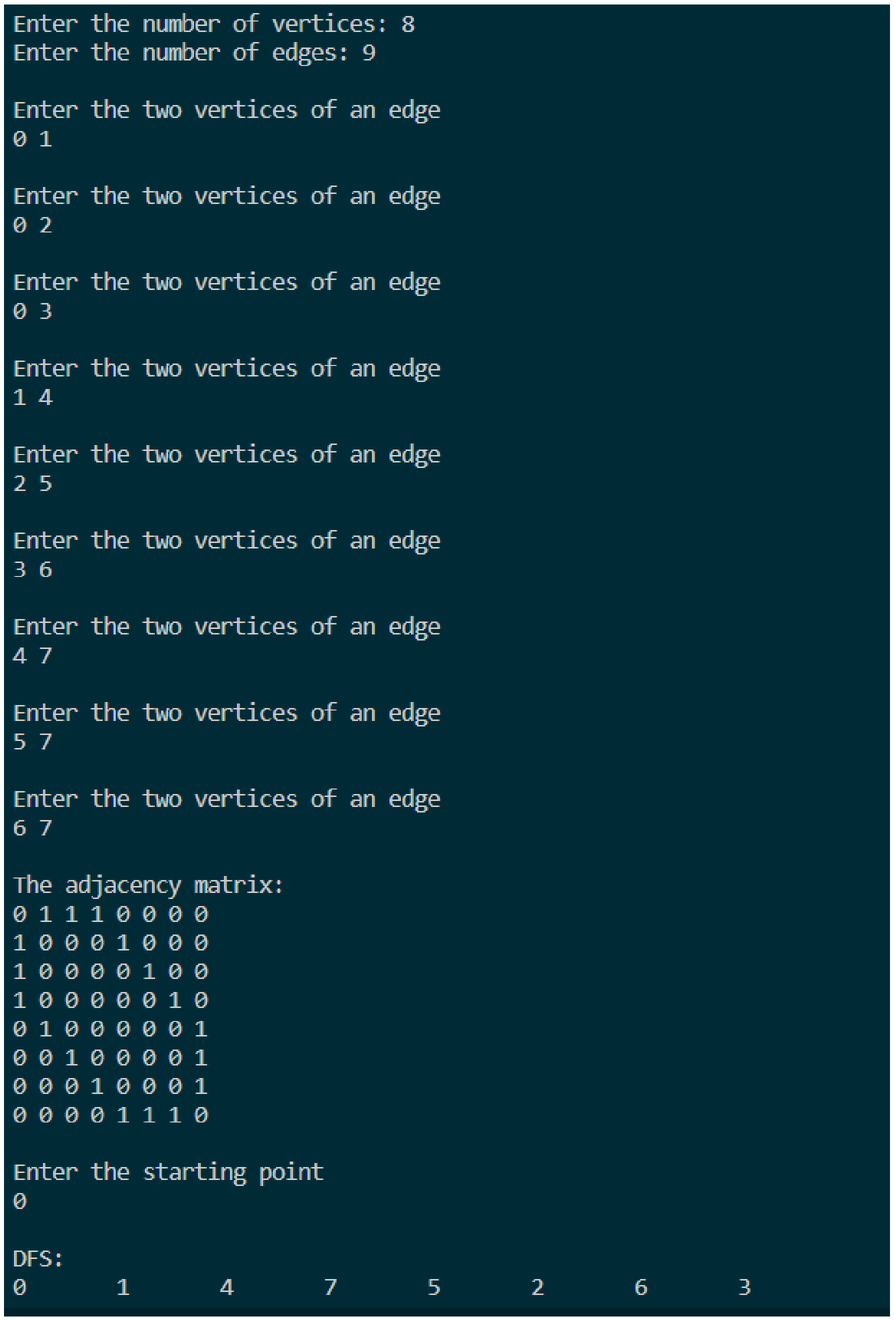
Source code:

```
#include<stdio.h>
#include<string.h>

void dfs(int v,int vt[v],int a[v][v],int s)
{
    int i;
    printf("%d\t", s);
    vt[s]=1;
    for( i=0; i<v; i++ )
    {
        if( a[s][i] == 1 && vt[i] == 0 )
        {
            dfs(v,vt,a,i);
        }
    }
}

int main()
{
    int v,e;
    printf("Enter the number of vertices: ");
    scanf("%d",&v);
    printf("Enter the number of edges: ");
    scanf("%d",&e);
    int a[v][v], vt[v];
    int i,j,p,start;
    for( i=0; i<v; i++ )
    {
        vt[i]=0;
        for( j=0; j<v; j++ )
        {
            a[i][j]=0;
        }
    }
    for( p=0; p<e; p++ )
    {
        printf("\nEnter the two vertices of an edge\n");
        scanf("%d%d", &i, &j);
        a[i][j]=1;
        a[j][i]=1;
    }
    printf("\nThe adjacency matrix:\n");
    for( i=0; i<v; i++)
    {
        for( j=0; j<v; j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    printf("\nEnter the starting point\n");
    scanf("%d", &start);
    printf("\nDFS:\n");
    dfs(v,vt,a,start);
}
```

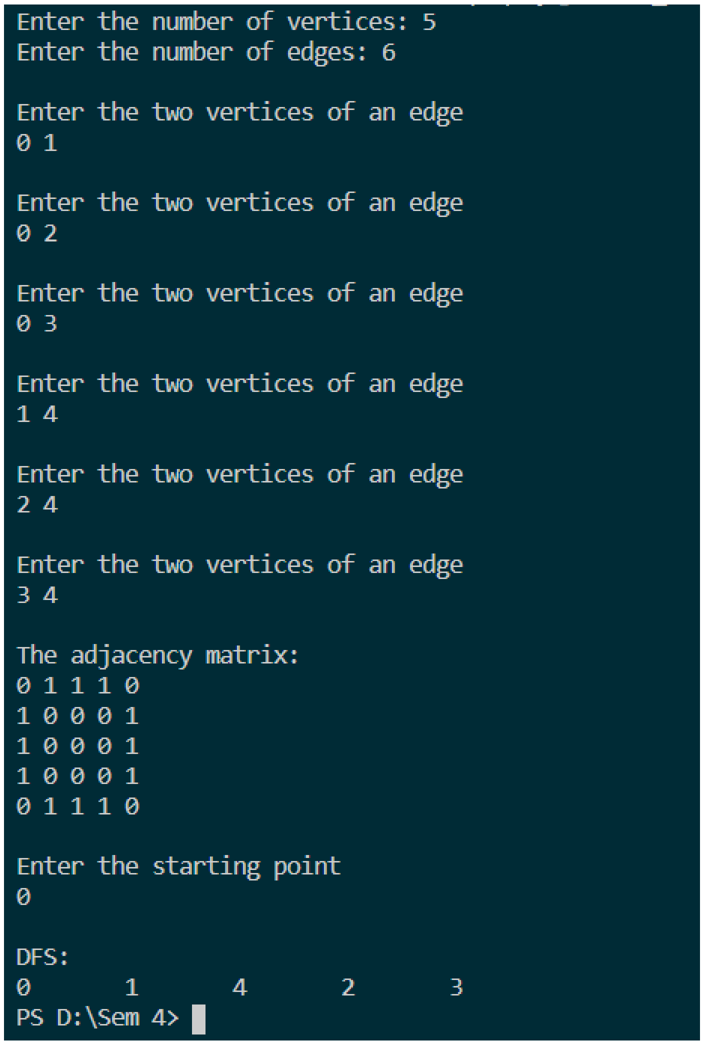
Output:



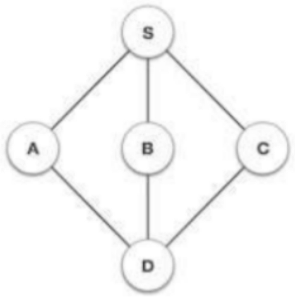
*Consider sABCDEFG as 01234567 respectively



```
Enter the number of vertices: 8
Enter the number of edges: 9

Enter the two vertices of an edge
0 1

Enter the two vertices of an edge
0 2

Enter the two vertices of an edge
0 3

Enter the two vertices of an edge
1 4

Enter the two vertices of an edge
2 5

Enter the two vertices of an edge
3 6

Enter the two vertices of an edge
4 7

Enter the two vertices of an edge
5 7

Enter the two vertices of an edge
6 7

The adjacency matrix:
0 1 1 1 0 0 0 0
1 0 0 0 1 0 0 0
1 0 0 0 0 1 0 0
1 0 0 0 0 0 1 0
0 1 0 0 0 0 0 1
0 0 1 0 0 0 0 1
0 0 0 1 0 0 0 1
0 0 0 0 1 1 1 0

Enter the starting point
0

DFS:
0       1       4       7       5       2       6       3
```



```
Enter the number of vertices: 5
Enter the number of edges: 6

Enter the two vertices of an edge
0 1

Enter the two vertices of an edge
0 2

Enter the two vertices of an edge
0 3

Enter the two vertices of an edge
1 4

Enter the two vertices of an edge
2 4

Enter the two vertices of an edge
3 4

The adjacency matrix:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0

Enter the starting point
0

DFS:
0       1       4       2       3
PS D:\Sem 4>
```

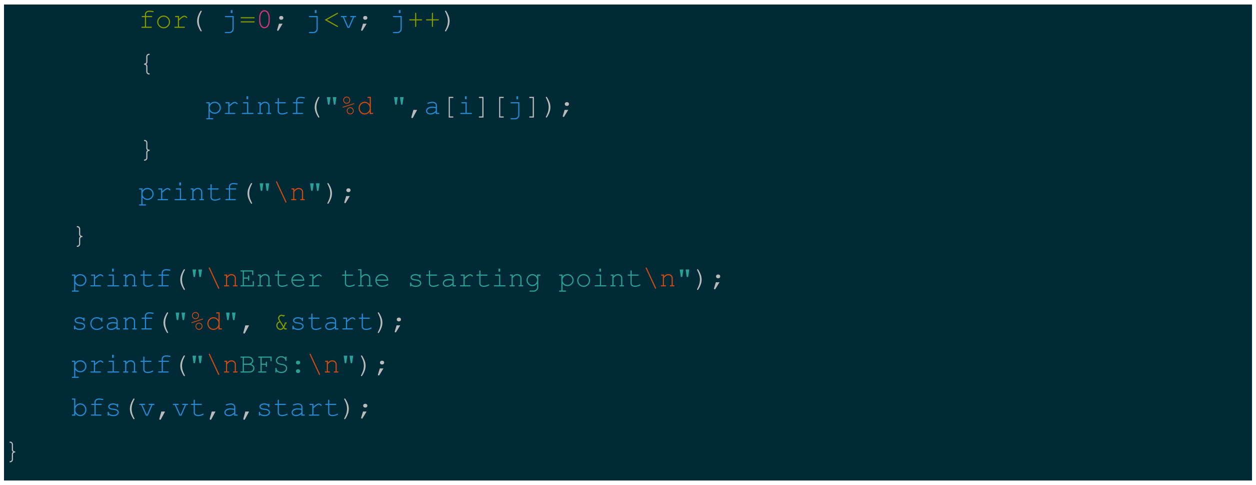


*Consider sABCD as 01234 respectively

Program statement: 2) Breadth First Search

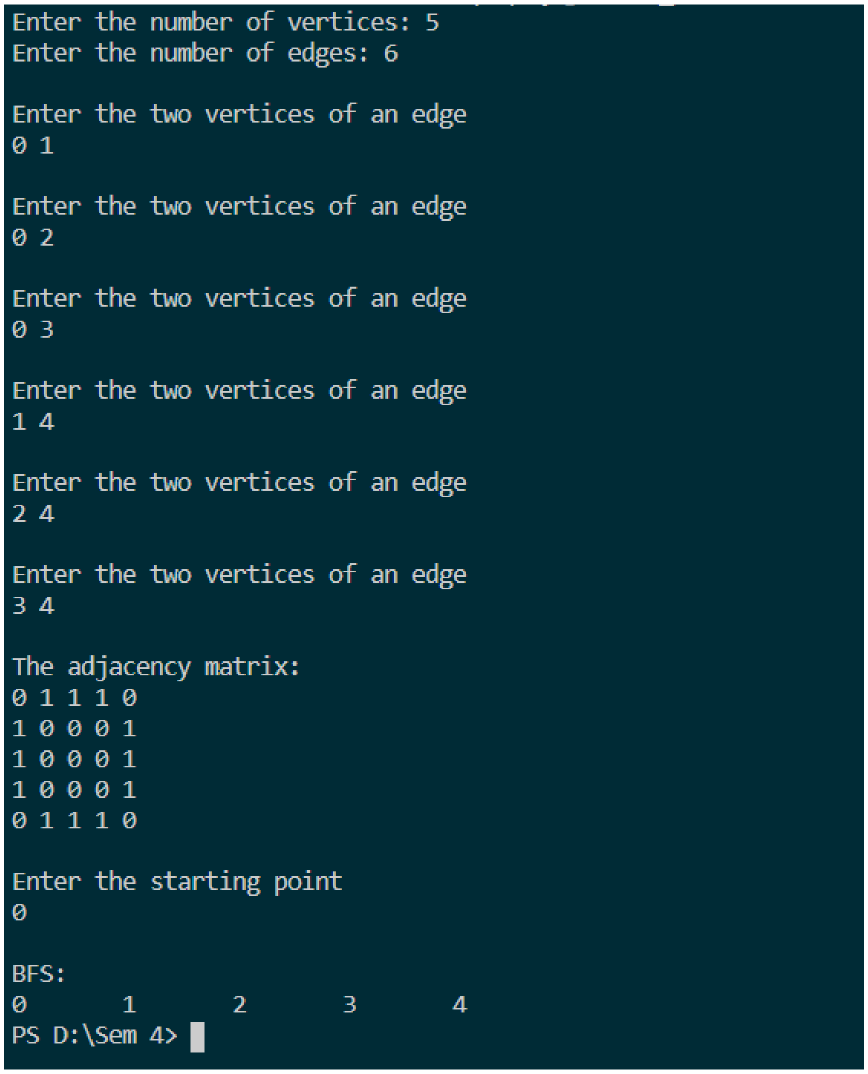
Source code:

```
#include<stdio.h>
#include<string.h>

void bfs(int v,int vt[v],int a[v][v],int s)
{
    int q[v], r=-1, f=-1, i;
    q[++r]=s; //EnQueue
    vt[s]=1;
    while(r!=f)
    {
        s=q[++f]; //DeQueue
        printf("%d\t",s);
        for( i=0; i<v; i++ )
        {
            if( a[s][i]==1 && vt[i]==0 ) //isEmpty and isVisited
            {
                q[++r]=i; //EnQueue
                vt[i]=1;
            }
        }
    }
}

int main()
{
    int v,e;
    printf("Enter the number of vertices: ");
    scanf("%d",&v);
    printf("Enter the number of edges: ");
    scanf("%d",&e);
    int a[v][v], vt[v];
    int i,j,p,start;
    for( i=0; i<v; i++ )
    {
        vt[i]=0;
        for( j=0; j<v; j++ )
        {
            a[i][j]=0;
        }
    }
    for( p=0; p<e; p++ )
    {
        printf("\nEnter the two vertices of an edge\n");
        scanf("%d%d", &i, &j);
        a[i][j]=1;
        a[j][i]=1;
    }
    printf("\nThe adjacency matrix:\n");
    for( i=0; i<v; i++)
    {
```

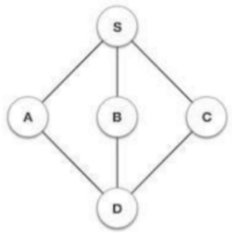


```c
        for( j=0; j<v; j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    printf("\nEnter the starting point\n");
    scanf("%d", &start);
    printf("\nBFS:\n");
    bfs(v,vt,a,start);
}
```

Output:

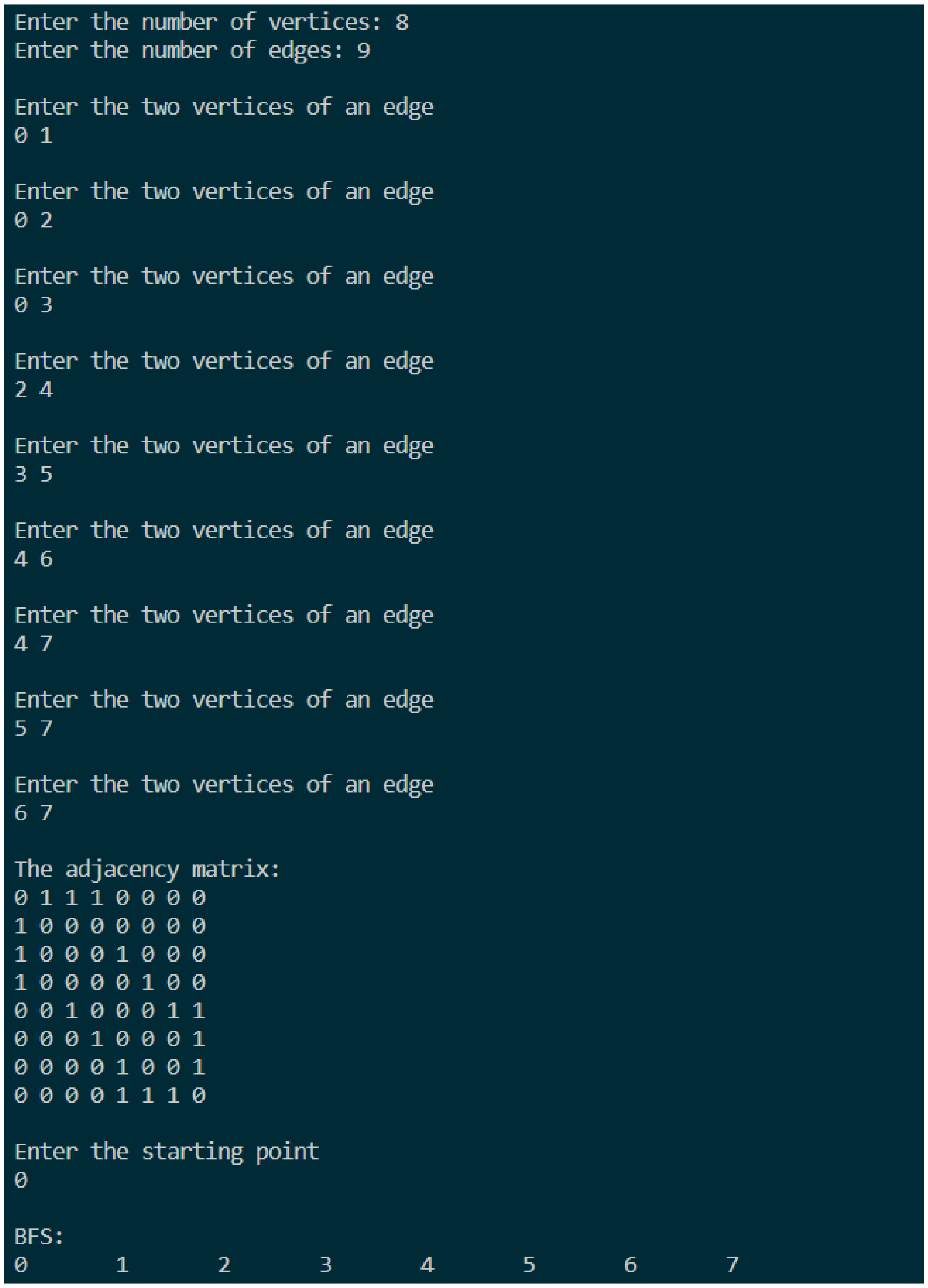


```
Enter the number of vertices: 5
Enter the number of edges: 6

Enter the two vertices of an edge
0 1

Enter the two vertices of an edge
0 2

Enter the two vertices of an edge
0 3

Enter the two vertices of an edge
1 4

Enter the two vertices of an edge
2 4

Enter the two vertices of an edge
3 4

The adjacency matrix:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
0 1 1 1 0

Enter the starting point
0

BFS:
0       1       2       3       4
PS D:\Sem 4> 
```

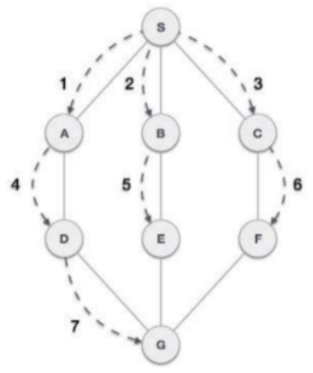


*Consider sABCD as 01234 respectively

Senjuti Maiti 19S-221



```
Enter the number of vertices: 8
Enter the number of edges: 9

Enter the two vertices of an edge
0 1

Enter the two vertices of an edge
0 2

Enter the two vertices of an edge
0 3

Enter the two vertices of an edge
2 4

Enter the two vertices of an edge
3 5

Enter the two vertices of an edge
4 6

Enter the two vertices of an edge
4 7

Enter the two vertices of an edge
5 7

Enter the two vertices of an edge
6 7

The adjacency matrix:
0 1 1 1 0 0 0 0
1 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0
1 0 0 0 0 1 0 0
0 0 1 0 0 0 1 1
0 0 0 1 0 0 0 1
0 0 0 0 1 0 0 1
0 0 0 0 1 1 1 0

Enter the starting point
0

BFS:
0       1       2       3       4       5       6       7
```



*Consider sABCDEFG as 01234567 respectively