Name: Atiq Israk Niloy	ID: 2016100000039	Section: 2

```
#include<bits/stdc++.h>
using namespace std;
int graph[100][100],cost[100],color_dfs[100],path[100], ans[100];
struct node{
    int u,w; //w & u are connected pairs
};
void dfsfun(int n,int s,int d){
    for(int i=1;i<=n;i++)
        path[i] = -1; //All paths are unvisited
    stack<node>S; //s is a stack which is node type, where we can store 2 values
    S.push({s,0}); //taking pair input in curly braces, pushing source in source stack and 
source cost is 0
    color_dfs[s]=1; //Making the color 1 as the path is visited
    cost[s]=0; //s to s cost is 0
    while(S.size()!=0) //run until the stack S is empty
    {
        int u = S.top().u, w = S.top().w; //taking top pair
        S.pop(); //popping node from the stack
        for(int i=1;i<=n;i++){
            if(graph[u][i] == 1 && color_dfs[i]==0) //if there is edge between u & i and if 
path's visited
            {
                S.push({i,w+1});
                color_dfs[i] = 1; //marking as visited
                cost[i] = w+1; //increase cost of child
                path[i] = u; //u is the parent of i
            }
        }
    }
    int count=0;
    cout << "Distance : " << cost[d] << endl;
    for(int j = d;j!=-1;j=path[j]) //destination is assigned in j. j!=-1 is source of parent.
path[j]=parent of j
    {
        ans[count++] = j; //if path exists,increment count
    }
    cout << "Path : ";
    for(int i=count-1;i>=0;i--){
        cout << ans[i] << " ";
    }
    cout<<endl;
}
int main(){
    int u, v, n, e, s, d;
    cout<<"Number of nodes:"; cin >> n;
    cout<<"Number of edges: "; cin>> e;
    cout<<"Connected nodes: "<<endl;
    for(int i=1; i<=e; i++){
    cin >> u >> v;
    graph[u][v] = graph[v][u] = 1; //u & v are bidirectional
 }
    cout<<"Source:   "; cin >> s;
    cout<<"Destination:   "; cin>> d;
    cout<<endl<<endl<<"Running DFS...."<<endl;
    dfsfun(n,s,d); //calling dfs function
}
```