

1 .DFS

```
#include <iostream>

#include<vector>

using namespace std;

int main()

{ cout << "\n\nWelcome to Studytonight :-)\n\n\n";

  cout << " ===== Program to demonstrate the DFS Traversal on a Graph, in CPP ===== \n\n";

  //variable declaration

  int cost[10][10], i, j, k, n, e, top, v, stk[10], visit[10], visited[10];

  cout << "Enter the number of vertices in the Graph: ";

  cin >> n;

  cout << "\nEnter the number of edges in the Graph : ";

  cin >> e;

  cout << "\nEnter the start and end vertex of the edges: \n";

  for (k = 1; k <= e; k++)

  {   cin >> i >> j;

      cost[i][j] = 1;

  }   cout << "\nEnter the initial vertex to start the DFS traversal with: ";

  cin >> v;

  cout << "\nThe DFS traversal on the given graph is : \n";

  cout << v << " ";

  //As we start with the vertex v, marking it visited to avoid visiting again

  visited[v] = 1;

  k = 1;

  //The DFS Traversal Logic

  while (k < n)

  { for (j = n; j >= 1; j--)

      { if (cost[v][j] != 0 && visited[j] != 1 && visit[j] != 1)

          { visit[j] = 1;
```

```

//put all the vertices that are connected to the visited vertex into a stack
    stk[top] = j;

    top++;

}

} //output all the connected vertices one at a time
v = stk[--top];
cout << v << " ";

k++;

//as v is visited so it is not a valid candidate to visit in future so visit[v]=0 and visited[v]=1
    visit[v] = 0;

//to mark it visited
    visited[v] = 1;

} cout << "\n\n";

return 0;

}

```