//Write C++ program to Implement DFS.

```
#include <iostream>

using namespace std;

int main() {

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

 // Variable declarations

   int cost[10][10] = {0}; // Adjacency matrix, initialized to 0


  int i, j, k, n, e, top = -1, v;

   int stk[10], visit[10] = {0}, visited[10] = {0};


  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;

      cost[j][i] = 1; // For undirected graph

   }


   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 << " ";
visited[v] = 1; // Mark the starting vertex as visited

k = 1;
while (k < n) {
   for (j = n; j >= 1; j--) {
      if (cost[v][j] != 0 && visited[j] != 1 && visit[j] != 1) {
         visit[j] = 1;
         stk[++top] = j; // Push vertex onto the stack
      }
   }

   if (top == -1) {
      break; // If stack is empty, traversal is complete
   }

   v = stk[top--]; // Pop vertex from the stack
   cout << v << " ";
   visit[v] = 0;  // Mark it as removed from the visit stack
   visited[v] = 1; // Mark it as visited
   k++;
}
```

```
    cout << "\n";
    return 0;
}
```

//Output

Enter the number of vertices in the Graph:

5

Enter the number of edges in the Graph: 4

Enter the start and end vertex of the edges:

1 2

1 3

2 4

3 5

Enter the initial vertex to start the DFS traversal with: 1

The DFS traversal on the given graph is:

1 2 4 3 5