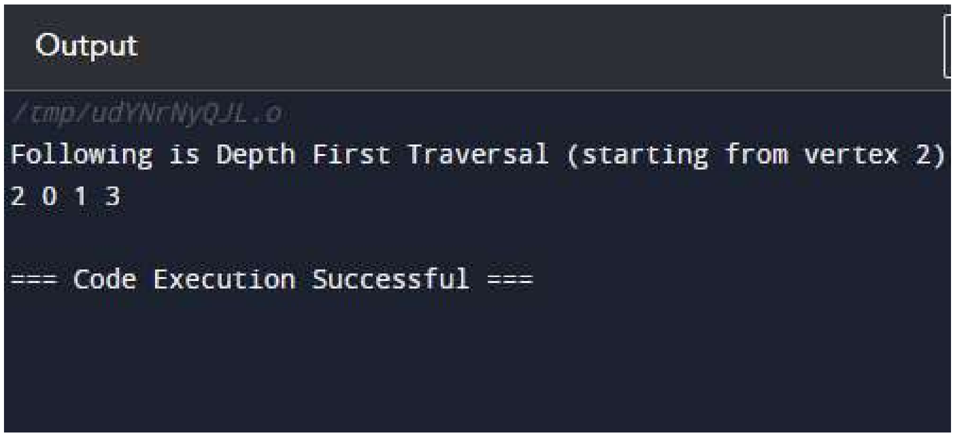
DFS TRAVERSAL IN A GRAPH

```
// C++ program to print DFS traversal from
// a given vertex in a given graph
#include <bits/stdc++.h>
using namespace std;

// Graph class represents a directed graph
// using adjacency list representation
class Graph {
public:
	map<int, bool> visited;
	map<int, list<int> > adj;

	// Function to add an edge to graph
	void addEdge(int v, int w);

	// DFS traversal of the vertices
	// reachable from v
	void DFS(int v);
};

void Graph::addEdge(int v, int w)
{
	// Add w to v’s list.
	adj[v].push_back(w);
}
```

```
void Graph::DFS(int v)
{
        // Mark the current node as visited and
        // print it
        visited[v] = true;
        cout << v << " ";

        // Recur for all the vertices adjacent
        // to this vertex
        list<int>::iterator i;
        for (i = adj[v].begin(); i != adj[v].end(); ++i)
                if (!visited[*i])
                        DFS(*i);
}

// Driver code
int main()
{
        Graph g;
        g.addEdge(1, 2);
        g.addEdge(1, 3);
        g.addEdge(2, 3);
        g.addEdge(2, 5);
        g.addEdge(3, 4);
        g.addEdge(4, 6);
```

```
    g.addEdge(5, 6);

        cout << "Following is Depth First Traversal"
                " (starting from vertex 0) \n";

    // Function call
    g.DFS(0);
    return 0;
}
```

OUTPUT:



Output

```
/tmp/udYNrNyQJL.o
Following is Depth First Traversal (starting from vertex 2)
2 0 1 3

=== Code Execution Successful ===
```