NAME : JAGADEESH R
REGNO : 2021506314
ADS LAB 10 : Topological sorting

SOURCE CODE :

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class Graph {
	int V;
	list<int>* adj;
	void topologicalSortUtil(int v, bool visited[], stack<int>& Stack);
public:
	Graph(int V);
	void addEdge(int v, int w);
	void topologicalSort();
};
Graph::Graph(int V)
{
	this->V = V;
	adj = new list<int>[V];
}
void Graph::addEdge(int v, int w)
{
	adj[v].push_back(w);
}
void Graph::topologicalSortUtil(int v, bool visited[], stack<int>& Stack)
{
	visited[v] = true;
	list<int>::iterator i;
	for (i = adj[v].begin(); i != adj[v].end(); ++i)
		if (!visited[*i])
			topologicalSortUtil(*i, visited, Stack);
	Stack.push(v);
}
void Graph::topologicalSort()
{
	stack<int> Stack;
	bool* visited = new bool[V];
	for (int i = 0; i < V; i++)
		visited[i] = false;
	for (int i = 0; i < V; i++)
		if (visited[i] == false)
			topologicalSortUtil(i, visited, Stack);
	while (Stack.empty() == false) {
		cout<<Stack.top()<<" ";
```

```
                Stack.pop();
        }
}
int main()
{
   int nn,i,no,dr;
   cout<<"Enter the number of nodes in the graph : "<<endl;
        cin>>nn;
        Graph g(nn);

        for(i=0;i<nn;i++)
        {
                cout<<"Enter the values for the node with its directed node : "<<endl;
                cin>>no;
             cin>>dr;
           g.addEdge(no, dr);
        }
        cout<<"Topological Sort for the given graph is : "<<endl;
        g.topologicalSort();

        return 0;
}
```

OUTPUT :

```
Enter the number of nodes in the graph :
6
Enter the values for the node with its directed node :
5
2
Enter the values for the node with its directed node :
5
0
Enter the values for the node with its directed node :
4
0
Enter the values for the node with its directed node :
4
1
Enter the values for the node with its directed node :
2
3
Enter the values for the node with its directed node :
3
1
Topological Sort for the given graph is :
5 4 2 3 1 0
```