Lab Test
Q3

```
#include <iostream>
#include <list>
using namespace std;

class DFSGraph
{
int V;
list<int> *adjList;
void DFS_util(int v, bool visited[]);
public:

DFSGraph(int V)
   {
 this->V = V;
 adjList = new list<int>[V];
   }

void addEdge(int v, int w){
adjList[v].push_back(w);
   }

void DFS();
};
void DFSGraph::DFS_util(int v, bool visited[])
{

visited[v] = true;
cout << v << " ";
```

```
list<int>::iterator i;
for(i = adjList[v].begin(); i != adjList[v].end(); ++i)
if(!visited[*i])
DFS_util(*i, visited);
}

void DFSGraph::DFS()
{

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)
DFS_util(i, visited);
}

int main()
{

DFSGraph gdfs(5);
gdfs.addEdge(0, 1);
gdfs.addEdge(0, 2);
gdfs.addEdge(0, 3);
gdfs.addEdge(1, 2);
gdfs.addEdge(2, 4);
```

```
gdfs.addEdge(3, 3);
gdfs.addEdge(4, 4);

cout << "Depth-first traversal for the given graph:"<<endl;
gdfs.DFS();

return 0;
}
```