NAME : JAGADEESH R
REGNO : 2021506314
ADS LAB 10 : BFS (Breadth First Search) Graph

SOURCE CODE :

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
class Graph {
  int numVertices;
  list<int>* adjLists;
  bool* visited;
  public:
  Graph(int vertices);
  void addEdge(int src, int dest);
  void BFS(int startVertex);
};
Graph::Graph(int vertices) {
  numVertices = vertices;
  adjLists = new list<int>[vertices];
}
void Graph::addEdge(int src, int dest)
{
  adjLists[src].push_back(dest);
  adjLists[dest].push_back(src);
}
void Graph::BFS(int startVertex)
{
  visited = new bool[numVertices];
  for (int i = 0; i < numVertices; i++)
    visited[i] = false;
  list<int> queue;
  visited[startVertex] = true;
  queue.push_back(startVertex);
  list<int>::iterator i;
  while (!queue.empty()) {
    int currVertex = queue.front();
    cout << "Visited " << currVertex << "\n";
    queue.pop_front();
    for (i = adjLists[currVertex].begin(); i != adjLists[currVertex].end(); ++i)
    {
      int adjVertex = *i;
      if (!visited[adjVertex]) {
        visited[adjVertex] = true;
        queue.push_back(adjVertex);
      }
```

```
   }
  }
}
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);
         }
  g.BFS(5);
  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
Visited 5
Visited 2
Visited 0
Visited 3
Visited 4
Visited 1
```