Name: M Maheeth Reddy Roll No.:1801CS31

CS206 Midsem Assignment

Program to count number of PATHS present in a graph given the starting vertex:

Code:

```
#include<bits/stdc++.h>
using namespace std;

class Graph {
  int V;
  list<int> *neighbours;
  int *count;
  void numPathsUtil(int u, int d, bool visited[], int &noOfPaths) {
    visited[u] = true;
    if (u == d) noOfPaths++;
    else {
      list<int>::iterator i;
      for (i = neighbours[u].begin(); i != neighbours[u].end(); ++i)
        if (!visited[*i]) numPathsUtil(*i, d, visited, noOfPaths);
    }
    visited[u] = false;
  }
public:
  Graph(int n) {
    V = n;
    neighbours = new list<int>[n];
  }
```

```
  void getEdge(int head, int tail) {
    neighbours[head].push_back(tail); neighbours[tail].push_back(head);
  }
  int numPaths(int start, int end) {
    bool *visited = new bool[V];
    for(int i = 0; i < V; i++) visited[i] = false;
    int noOfPaths = 0;
    numPathsUtil(start, end, visited, noOfPaths);
    return noOfPaths;
  }
};

int main() {
  int n,source;
  cout << "Give no. of vertices: ";
  cin >> n;
  Graph g(n);
  cout << "Give edges: (Hit -1 after giving all)\n";
  for(int i = 0; i < n*(n-1)/2; i++) {
    int x,y;
    cin >> x;
    if(x == -1) break;
    cin >> y;
    g.getEdge(x,y);
  }
  cout << "Give source vertex: ";
  cin >> source;
  int total = 0;
```

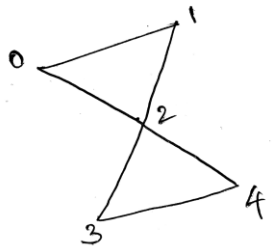
```
  for(int end = 0; end < n; end++) {
    if(end != source) {
      total += g.numPaths(source, end);
    }
  }
  cout << "Total paths from given source vertex is " << total << endl;
}
```

Input:

Give no. of vertices: 5

Give edges: (Hit -1 after giving all)

0 1
0 2
1 2
2 3
2 4
3 4
-1



Give source vertex: 1

Output:

Total no. of paths from given source vertex is 12