# Breadth First Search

vector<int>parent, G[MAX];

void printPath(int u, int source\_node) {

if(u == source\_node) {

printf("%d", u);

return;

}

printPath(parent[u], source\_node);

printf(" %d", u);

}

int BFS(int source\_node, int finish\_node, int vertices) {

//contains the distance from source to end point

vector<int>dist(vertices+5, INF);

queue<int>Q;

Q.push(source\_node);

//for path printing

parent.resize(vertices+5, -1);

while(!Q.empty()) {

int u = Q.front();

Q.pop();

//remove this line if shortest path to all nodes are needed

if(u == finish\_node)

return dist[u];

for(int i = 0; i < G[u].size(); i++) {

int v = G[u][i];

if(dist[v] == INF) {

dist[v] = dist[u] + 1;

parent[v] = u;

Q.push(v);

}

}

}

//if not found, return -1

return -1;

}

main () {}