Assignment 08

Name: Deepak Singh **Roll No. 20bcs102**

Solution 1

Code:

```
#include <iostream>
#include <queue>
#include <vector>
using namespace std;
void addEdge(vector<vector<int>> &adyList) {
 int x, y;cout << "Enter the edge : " << endl;
 cout << "\tstart vertex and end index";cin >> x >> y;
 adyList[x].push_back(y);
}
void takeInput(vector<vector<int>> &adyList) {
 adyList.clear();int vertices, edges;
 cout << "Enter the number of vertices: ";cin >> vertices;
 for (int i = 0; i < vertices; i++) {
   vector<int> a;adyList.push_back(a);
 }
 cout << "Enter the number of edges: ";cin >> edges;
 for (int i = 0; i < edges; i++) {
   addEdge(adyList);
 }
 cout << "Successfully entered graph into Adjancy List " << endl;
}
void addVertex(vector<vector<int>> &adList) {
 vector<int> emptyVector;
 adList.push_back(emptyVector);
 int newEdges;
 cout << "Enter the number of new edges this new vertex makes: ";cin >>
newEdges;
```

```
 for (int i = 0; i < newEdges; i++) {
   int x, y;
   cout << "Enter the edge " << i + 1 << ": " << endl;
   cout << "\tstart vertex: ";cin >> x;
   cout << "\tEnd vertex: ";cin >> y;
   adList[x].push_back(y);
 }
}
void breadthFirstSearch(vector<vector<int>> &adyList, int start) {
 cout << "Breadth First Search traversal : ";
 bool status[adyList.size()];
 for (int i = 0; i < adyList.size(); i++) {
   status[i] = false;
 }
 queue<int> q;
 q.push(start);
 int element;
 while (!q.empty()) {
   element = q.front();
   if (status[element] == false) {
     status[element] = true;
     for (int i = 0; i < adyList[element].size(); i++) {
       q.push(adyList[element][i]);
     }
     cout << element << "\t";
   }
   q.pop();
 }
 cout << "\n";
}
void dfsTraversal(vector<vector<int>> &adyList, bool status[], int start) {
 status[start] = true;
```

```
 cout << start << "\t";
 vector<int>::iterator it;
 for (it = adyList[start].begin(); it != adyList[start].end(); ++it) {
   if (!status[*it]) {
     dfsTraversal(adyList, status, *it);
   }
 }
}
void depthFirstSearch(vector<vector<int>> &adyList, int start) {
 bool status[adyList.size()];
 for (int i = 0; i < adyList.size(); i++) {
   status[i] = false;
 }
 cout << "Depth First Search traversal : ";
 dfsTraversal(adyList, status, start);
 cout << "\n";
}
void selectOption() {
 int choice;
 vector<vector<int>> adyList;
 while (true) {
   cout << "\n Menu\n" << endl;
   cout << "\t1. Input New Graph" << endl;
   cout << "\t2. Add a vertex to the graph" << endl;
   cout << "\t3. Add an edge to the graph" << endl;
   cout << "\t4. Display Breadth First Traversal of Graph" << endl;
   cout << "\t5. Display Depth First Traversal of Graph" << endl;
   cout << "\t6. Exit" << endl;
   cout << "Choice: ";cin >> choice;
   if (choice == 1) {takeInput(adyList);
   } else if (choice == 2) {addVertex(adyList);
   } else if (choice == 3) {addEdge(adyList);
```

```
   } else if (choice == 4) {int start;
     cout << "Enter the start vertex: ";cin >>
start;breadthFirstSearch(adyList, start);
   } else if (choice == 5) {
     int start;
     cout << "Enter the start vertex: ";cin >>
start;depthFirstSearch(adyList, start);
   } else {exit(0);
   }
 }
}
int main() {
 selectOption();
 return 0;
}
```

Output:

```
 Menu

        1. Input New Graph
        2. Add a vertex to the graph
        3. Add an edge to the graph
        4. Display Breadth First Traversal of Graph
        5. Display Depth First Traversal of Graph
        6. Exit
Choice: 1
Enter the number of vertices: 6
Enter the number of edges: 5
Enter the edge :
        start vertex and end index4 6
Enter the edge :
        start vertex and end index1 4
Enter the edge :
        start vertex and end index1 5
Enter the edge :
        start vertex and end index5 3
Enter the edge :
        start vertex and end index6 2
Successfully entered graph into Adjancy List

 Menu

        1. Input New Graph
        2. Add a vertex to the graph
        3. Add an edge to the graph
        4. Display Breadth First Traversal of Graph
        5. Display Depth First Traversal of Graph
        6. Exit
Choice: 4
Enter the start vertex: 5
Breadth First Search traversal : 5      3

 Menu

        1. Input New Graph
        2. Add a vertex to the graph
        3. Add an edge to the graph
        4. Display Breadth First Traversal of Graph
        5. Display Depth First Traversal of Graph
        6. Exit
Choice: 5
Enter the start vertex: 6
Depth First Search traversal : 6        2
```

```
 Menu

        1. Input New Graph
        2. Add a vertex to the graph
        3. Add an edge to the graph
        4. Display Breadth First Traversal of Graph
        5. Display Depth First Traversal of Graph
        6. Exit
Choice: 2
Enter the number of new edges this new vertex makes: 2
Enter the edge 1:
        start vertex: 6
        End vertex: 2
Enter the edge 2:
        start vertex: 2
        End vertex: 8

 Menu

        1. Input New Graph
        2. Add a vertex to the graph
        3. Add an edge to the graph
        4. Display Breadth First Traversal of Graph
        5. Display Depth First Traversal of Graph
        6. Exit
Choice: 5
Enter the start vertex: 8
```