Assignment no. 1

BFS-

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

const int MV = 100;

void inputGraph(int adj[MV][MV], int& vertices, int& edges)
{
   cout << "enter the number of vertices and edges:";
cin >> vertices >> edges;     cout << "enter the edges
(v1 v2):" << endl;
   for (int i = 0; i < edges; ++i)
   {
      int v1, v2;
cin >> v1 >> v2;
adj[v1][v2] = 1;
adj[v2][v1] = 1;
   }
}
void displayGraph(const int adj[MV][MV], int vertices)
{
   cout << "graph representation:" << endl;
for (int i = 1; i <= vertices; ++i)
   {
      for (int j = 1; j <= vertices; ++j)
      {
         cout << adj[i][j] << "";
      }
```

```
      cout << endl;
   }
}

void bfsUsingQueue(int startVertex, const int adj[MV][MV], bool visited[MV])
{
   queue<int> q;
   q.push(startVertex);
visited[startVertex] = true;

   while (!q.empty())
   {
      int currentVertex = q.front();
cout << currentVertex << "";
      q.pop();

      for (int i = 1; i <= MV; ++i)
      {
         if (adj[currentVertex][i] == 1 && !visited[i])
         {
            q.push(i);
visited[i] = true;
         }
      }
   }
}
int main()
{
   int   adj[MV][MV]   =   {0};
int       vertices,       edges;
inputGraph(adj,     vertices,
```

```
edges);     displayGraph(adj,
vertices);                  bool
visited[MV] = {false};


   int SN;
   cout<<"Enter the starting node:";
cin>>SN;     cout << "BFS traversal from
starting node:";     bfsUsingQueue(SN, adj,
visited);


   return 0; 1                                  Error! Bookmark not defined.
1                                                                          4
1                                                                          6
2                                                                          6
3                                                                          6
4                                                                          6


}
```

Output-

```
enter the number of vertices and edges:5
7 enter the edges (v1
v2):
2 3
graph representation:
01110
10110
11001
11001
00110
Enter the starting node:1
BFS traversal from starting node:12345
```

DFS-

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

const int MV = 100;

void inputGraph(int adj[MV][MV], int& vertices, int& edges)
{
   cout << "enter the number of vertices and edges:";
cin >> vertices >> edges;     cout << "enter the edges
(v1 v2):" << endl;
   for (int i = 0; i < edges; ++i)
   {
      int v1, v2;
cin >> v1 >> v2;
adj[v1][v2] = 1;
adj[v2][v1] = 1;
   }
}

void displayGraph(const int adj[MV][MV], int vertices)
{
   cout << "graph representation:" << endl;
   for (int i = 1; i <= vertices; ++i)
   {
      for (int j = 1; j <= vertices; ++j)
      {
         cout << adj[i][j] << "";
      }
      cout << endl;  }
```

```
}
void dfsUsingStack(int startVertex, const int adj[MV][MV], bool visited[MV])
{
   stack<int> s;
   s.push(startVertex);

   while (!s.empty())
   {
      int currentVertex = s.top();
      s.pop();

      if (!visited[currentVertex])
      {
         cout << currentVertex << "";
visited[currentVertex] = true;
      }

      for (int i = MV; i >= 1; --i)
      {
         if (adj[currentVertex][i] == 1 && !visited[i])
         {
            s.push(i);
         }
      }
   }
}
int main()
{
   int   adj[MV][MV]   =   {0};
int       vertices,       edges;
inputGraph(adj,     vertices,
```

```
edges);     displayGraph(adj,
vertices);


   bool visited[MV] = {false};
int SN;
   cout<<"Enter the starting node:";
cin>>SN;     cout << "DFS traversal from
starting node:";     dfsUsingStack(SN, adj,
visited);     return 0;
}
```

Output-

```
enter the number of vertices and edges:5
8 enter the edges (v1
v2):
1 2
1 3
2 3
2 4
3 5
2 5
3 4
4 5
graph representation:
01100
10111
11011
01101
01110
Enter the starting node:2
DFS traversal from starting node:21345
```