

# LAB TASK # 12

## MOHAMMAD BASIL ALI KHAN

### 20K-0477

#### Task#01

- Code

```
Task#01.cpp x
Task#01.cpp > printGraph(vector<int> [], int)
1  #include<iostream>
2  #include<vector>
3
4  using namespace std;
5
6  void addEdge(vector<int> adj[], int u, int v)
7  {
8      adj[u].push_back(v);
9      adj[v].push_back(u);
10 }
11
12
13 void printGraph(vector<int> adj[], int V)
14 {
15     for (int v = 0; v < V; ++v)
16     {
17         cout << endl << "Adjacency list of vertex: " << v << endl << "head";
18         for(auto x : adj[v])
19         {
20             cout << "->" << x << endl;
21         }
22     }
23 }
24
25 int main()
26 {
27     int V = 4;
28     vector<int> *adj;
29     adj = new vector<int>[V];
30     addEdge(adj, 0, 1);
31     addEdge(adj, 1, 2);
32     addEdge(adj, 2, 3);
33     printGraph(adj, V);
34     return 0;
35 }
36
```

- Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12" ; if ($?) { g++ Task#01.cpp -o Task#01 }

Adjacency list of vertex: 0
head->1

Adjacency list of vertex: 1
head->0
->2

Adjacency list of vertex: 2
head->1
->3

Adjacency list of vertex: 3
head->2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

## Task#02:

- Code

```
Task#02.cpp X
Task#02.cpp > main()
1  #include <iostream>
2
3  using namespace std;
4
5  class Node{
6  public:
7      int data;
8      Node* next;
9
10     Node()
11     {
12         data = 0;
13         next = NULL;
14     }
15     Node(int val)
16     {
17         data = val;
18         next = NULL;
19     }
20 };
21
22 class LinkedList{
23 public:
24     Node* head;
25
26     LinkedList()
27     {
28         head = NULL;
29     }
30
31     void insert(int val)
32     {
33         Node *n = new Node(val);
```

```
Task#02.cpp X
Task#02.cpp > main()
31     void insert(int val)
32     {
33         Node *n = new Node(val);
34         if(head == NULL)
35         {
36             head = n;
37             return;
38         }
39         Node* temp = head;
40         while(temp->next != NULL)
41         {
42             temp = temp->next;
43         }
44         temp->next = n;
45     }
46
47     bool search(int val)
48     {
49         if(head == NULL)
50         {
51             return false;
52         }
53         Node* temp = head;
54         while(temp != NULL)
55         {
56             if(temp->data == val)
57             {
58                 return true;
59             }
60             temp = temp->next;
61         }
62         return false;
```

```
Task#02.cpp X
Task#02.cpp > main()
65 void PrintEdges()
66 {
67     if(head == NULL)
68     {
69         cout << " -> NULL";
70         return;
71     }
72     Node* temp = head;
73     while(temp != NULL){
74         cout << " -> " << temp->data;
75         temp = temp->next;
76     }
77 }
78 };
79
80 class Graph{
81 public:
82     int size;
83     LinkedList *adj;
84
85     Graph(int n){
86         size = n;
87         adj = new LinkedList[n];
88     }
89
90     void addEdge(int u, int v)
91     {
92         if(!adj[u].search(v))
93         {
94             adj[u].insert(v);
95         }
96         if(!adj[v].search(u))
97         {
98             adj[v].insert(u);
99         }
100     }
101
102     void PrintGraph(){
103         for(int i=0; i<size; i++)
104         {
105             cout << endl << "Adjacency list of vertex " << i << endl << "Head ";
106             adj[i].PrintEdges();
107         }
108     }
109 };
110
111 int main()
112 {
113     Graph G(4);
114     G.addEdge(0,1);
115     G.addEdge(1,2);
116     G.addEdge(2,3);
117     G.PrintGraph();
118 }
```

```
Task#02.cpp X
Task#02.cpp > main()
87     adj = new LinkedList[n];
88 }
89
90 void addEdge(int u, int v)
91 {
92     if(!adj[u].search(v))
93     {
94         adj[u].insert(v);
95     }
96     if(!adj[v].search(u))
97     {
98         adj[v].insert(u);
99     }
100 }
101
102 void PrintGraph(){
103     for(int i=0; i<size; i++)
104     {
105         cout << endl << "Adjacency list of vertex " << i << endl << "Head ";
106         adj[i].PrintEdges();
107     }
108 }
109 };
110
111 int main()
112 {
113     Graph G(4);
114     G.addEdge(0,1);
115     G.addEdge(1,2);
116     G.addEdge(2,3);
117     G.PrintGraph();
118 }
```

- Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#02.cpp -o Task#02 } ; if ($?) { .\Task#02 }

Adjacency list of vertex 0
Head -> 1
Adjacency list of vertex 1
Head -> 0 -> 2
Adjacency list of vertex 2
Head -> 1 -> 3
Adjacency list of vertex 3
Head -> 2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

## Task#03:

- Code

```
Task#03.cpp X
Task#03.cpp > main()
1  #include<iostream>
2  #include<list>
3
4
5  using namespace std;
6
7  class Graph{
8  public:
9      int V;
10     list<int> *adj;
11
12     Graph(int val)
13     {
14         V = val;
15         adj = new list<int>[V];
16     }
17
18     void addEdge(int v, int w)
19     {
20         adj[v].push_back(w);
21     }
22
23     void BFS(int s)
24     {
25         bool *visited = new bool[V];
26         for(int i=0; i<V; i++)
27         {
28             visited[i] = false;
29         }
30
31         list<int> queue;
32         visited[s] = true;
33         queue.push_back(s);
```

```
Task#03.cpp x
Task#03.cpp > main()
20     adj[v].push_back(w);
21 }
22
23 void BFS(int s)
24 {
25     bool *visited = new bool[V];
26     for(int i=0; i<V; i++)
27     {
28         visited[i] = false;
29     }
30
31     list<int> queue;
32     visited[s] = true;
33     queue.push_back(s);
34     list<int>::iterator i;
35     while(!queue.empty())
36     {
37         s = queue.front();
38         cout << s << " ";
39         queue.pop_front();
40
41         for(i = adj[s].begin(); i!=adj[s].end(); i++)
42         {
43             if(!visited[*i])
44             {
45                 visited[*i] = true;
46                 queue.push_back(*i);
47             }
48         }
49     }
50 }
51 };
52
```

```
Task#03.cpp x
Task#03.cpp > main()
37     s = queue.front();
38     cout << s << " ";
39     queue.pop_front();
40
41     for(i = adj[s].begin(); i!=adj[s].end(); i++)
42     {
43         if(!visited[*i])
44         {
45             visited[*i] = true;
46             queue.push_back(*i);
47         }
48     }
49 }
50 }
51 };
52
53 int main()
54 {
55     Graph obj(10);
56     obj.addEdge(0, 1);
57     obj.addEdge(0, 2);
58     obj.addEdge(1, 5);
59     obj.addEdge(5, 0);
60     obj.addEdge(1, 2);
61     obj.addEdge(3, 6);
62     obj.addEdge(6, 9);
63     obj.addEdge(9, 1);
64     obj.addEdge(1, 2);
65     obj.addEdge(4, 5);
66     obj.BFS(4);
67 }
```

- **Output**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#03.cpp -o Task#03 } ; if ($?) { .\Task#03 }
4 5 0 1 2
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```

## Task#04:

- **Code**

```
Task#04.cpp X
Task#04.cpp > main()
1  #include<iostream>
2  #include<list>
3
4  using namespace std;
5
6  class Graph {
7  public:
8      int V;
9      bool *visited;
10     list<int> *adj;
11
12     Graph(int val)
13     {
14         V = val;
15         adj = new list<int>[V];
16     }
17
18     void addEdge(int v, int w)
19     {
20         adj[v].push_back(w);
21     }
22     void DFS(int v)
23     {
24         visited[v] = true;
25         cout << v << " ";
26         list<int>::iterator i;
27         for (i = adj[v].begin(); i != adj[v].end(); ++i)
28         {
29             if (!visited[*i])
30             {
31                 DFS(*i);
32             }
33         }
34     }
35 }
```

```
Task#04.cpp X
Task#04.cpp > main()
16
17
18 void addEdge(int v, int w)
19 {
20     adj[v].push_back(w);
21 }
22 void DFS(int v)
23 {
24     visited[v] = true;
25     cout << v << " ";
26     list<int>::iterator i;
27     for (i = adj[v].begin(); i != adj[v].end(); ++i)
28     {
29         if (!visited[*i])
30         {
31             DFS(*i);
32         }
33     }
34 }
35 };
36
37 int main()
38 {
39     Graph g(4);
40     g.addEdge(0, 1);
41     g.addEdge(0, 2);
42     g.addEdge(1, 2);
43     g.addEdge(2, 0);
44     g.addEdge(2, 3);
45     g.addEdge(3, 3);
46     g.DFS(2);
47 }
```

## • Output

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\FAST\3rd Semester\DS Lab\Lab Task 12> cd "e:\FAST\3rd Semester\DS Lab\Lab Task 12\" ; if ($?) { g++ Task#04.cpp -o Task#04 } ; if ($?) { .\Task#04 }
0 1 2 3
PS E:\FAST\3rd Semester\DS Lab\Lab Task 12>
```