Name : Dev Adnani
SID : 202212012
Subject : DSA
Topic : Graph
Lab : 7

Q1 :

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

int main()
{
	int m, n;
	cout << "Enter the number of Nodes: ";
	cin >> n;
	cout << "Enter the number of Edges: ";
	cin >> m;
	vector<int> ad[n + 1];
	for (auto i = 0; i < m; i++)
	{
	int a, b;
	cin >> a >> b;
	ad[a].push_back(b);
	ad[b].push_back(a);
	}
	vector<int> bs;
	vector<bool> vd(n, false);
	for (int i = 1; i <= n; i++)
	{
	queue<int> qe;
	if (!vd[i])
	{
		vd[i] = true;
```

```
            qe.push(i);
        }
        while (!qe.empty())
        {
            int tmp = qe.front();
            qe.pop();
            bs.push_back(tmp);
            for (auto p : ad[tmp])
            {
            }
        }
        }
        for (auto p : bs)
        {
        cout << p << " ";
        }
        return 0;
}
```

O/P :

Q2 :

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

class Node
{
public:
	int data;
	Node *left;
	Node *right;
	Node(int data)
	{
	this->data = data;
	this->left = NULL;
	this->right = NULL;
	}
};

Node *crtree(Node *root)
{
	int data;
	cin >> data;
	if (data == -1)
	return NULL;
	root = new Node(data);
	cout << "Enter the left element of " << data << " : ";
	root->left = crtree(root->left);
```

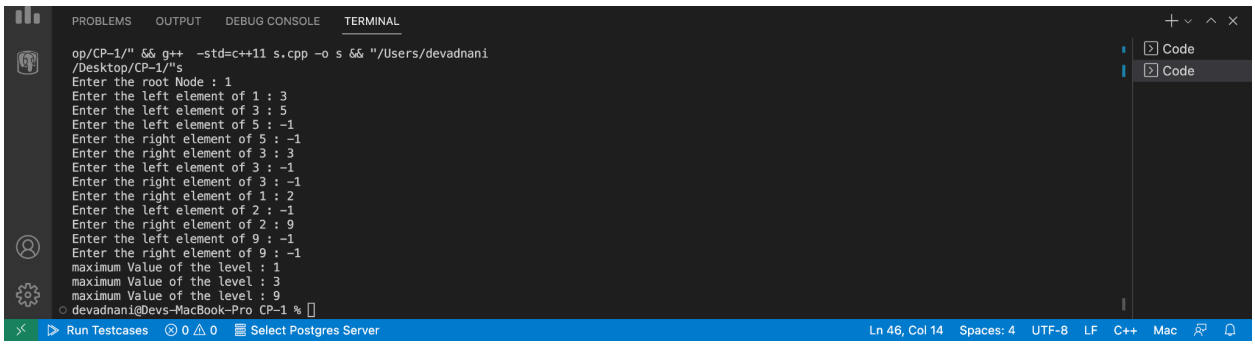
```
        cout << "Enter the right element of " << data << " : ";
        root->right = crtree(root->right);
        return root;
}

void maxVz(Node *root)
{
        queue<Node *> ax;
        ax.push(root);
        while (!ax.empty())
        {
        int maxa = 0;
        int size = ax.size();
        for (int i = 0; i < size; i++)
        {
                Node *temp = ax.front();
                ax.pop();
                maxa = max(maxa, temp->data);
                if (temp->left)
                {
                ax.push(temp->left);
                }
                if (
                temp->right)
                {
                ax.push(temp->right);
                }
        }
```

```
        cout << "maximum Value of the level : " << maxa << " "
<< endl;
        }
}

int main()
{
        Node *root;
        cout << "Enter the root Node : ";
        root = crtree(root);
        maxVz(root);
}
```

Output :



Q3 :

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

class Node
{
public:
	int data;
	Node *left;
	Node *right;
	Node *next;
	Node(int data)
	{
	this->data = data;
	this->left = NULL;
	this->right = NULL;
	this->next = NULL;
	}
};

Node *crtTree(Node *root)
{
	int data;
	cin >> data;
	if (data == -1)
	return NULL;
	root = new Node(data);
```

```
        cout << "Enter the left element of " << data << " : ";
        root->left = crtTree(root->left);
        cout << "enter the right element of " << data << " : ";
        root->right = crtTree(root->right);
        return root;
}

void nextLvl(Node *root)
{
        queue<Node *> q;
        q.push(root);
        cout << endl;
        while (!q.empty())
        {
        int size = q.size();
        for (int i = 0; i < size; i++)
        {
                Node *tmp = q.front();
                q.pop();
                if (!q.empty())
                {
                tmp->next = q.front();
                }
                if (
                i == size - 1)
                tmp->next = NULL;
                if (tmp->left)
                {
```

```
            q.push(tmp->left);
            }
            if (
            tmp->right)
            {
            q.push(tmp->right);
            }
        }
        }
}

void print(Node *root)
{
        if (root == NULL)
        return;
        Node *tmp = root;
        while (tmp)
        {
        cout << tmp->data << " ";
        tmp = tmp->next;
        }
        if (!tmp)
        cout << "Null"
                << " ";
        print(root->left);
}

int main()
```

```
{
        Node *root;
        cout << "Enter the root Node : ";
        root = crtTree(root);
        nextLvl(root);
        print(root);
}
```

O/P :