Name - SHIVAM GUPTA
Roll - 2109510C
Branch - ECE

VIJAYANT
PAGE NO.:
DATE:

Ques 1)

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

int depthFinder( char tree[], int n, int & index)
{
    if (index >= n || tree[index] == 'l')
        return 0;

    index ++;
    int left = depthFinder ( tree, n, index);

    index ++;
    int right = depthFinder ( tree, n, index);

    return max ( left, right) + 1;

int FindDepth ( char tree[] , int n)
{
    int index = 0;
    depthFinder ( tree, n, index);
}

int main ()
{
    char tree[]= " nlnnlll";
    int n = strlen(tree);

    cout << findDepth (tree, n) << endl;

    return 0;
}
```

Output : 3

Soln 2)

```
# include <stdio.h>
# include <stdlib.h>

struct node
{
    int data;
    struct node * left;
    struct node * right;
}

int isBSTU( struct node * node, int min, int max);

int isBST ( struct node * node)
{
    return (isBSTU (node, INT_MIN, INT_MAX));
}

int isBSTU ( struct node * node, int min, int max)
{
    if ( node == NULL)
        return 1;

    if (node -> data < min || node -> data > max)
        return 0;

    return
        isBSTU ( node -> left, min, node -> data - 1) &&
        isBSTU ( node -> right, node -> data + 1, max);
}
```

```
struct node * newNode (int data)
{
    struct node* node = (struct node*) malloc (sizeof (struct node));
    node → data = data;
    node → left = NULL;
    node → right = NULL;

    return (node);
}

int main ()
{
    struct node* root = newNode (7);
    root → left = newNode (9);
    root → right = newNode (5);
    root → left → left = newNode (6);
    root → left → right = newNode (1);

    if (isBST (root))
        printf ("Is BST");
    else
        printf ("Not a BST");

    getchar ();
    return 0;
}
```

Output :

Not a BST