NAME: Deepanshu Gupta

SEC: AI & ML

CLASS ROLL NO.: 10

PROBLEM STATEMENT: Assuming that you already have a BST with address root. Write following function (a) Write a function to count total number of nodes in BST. (b) Write a function to count total number of leaf nodes in BST. (c) Write a function to count total number of nodes which are having only one child.

CODE:

#include<stdio.h>

#include<stdlib.h>

typedef struct node

{

struct node \*left;

int info;

struct node \*right;

} NODE;

NODE \*insert(NODE \*tree, int ele)

{

if (tree==NULL)

{

tree=(NODE\*)malloc(sizeof(NODE));

tree->left=NULL;

tree->info=ele;

tree->right=NULL;

}

else if (ele>(tree->info))

tree->right=insert(tree->right, ele);

else

tree->left=insert(tree->left, ele);

return tree;

}

int count\_nodes(NODE \*tree)

{

if(tree==NULL)

return 0;

else

return 1+count\_nodes(tree->left)+count\_nodes(tree->right);

}

int count\_leaf\_nodes(NODE \*tree)

{

if(tree==NULL)

return 0;

else if ((tree->left)==NULL && (tree->right)==NULL)

return 1;

else

return count\_leaf\_nodes(tree->left)+count\_leaf\_nodes(tree->right);

}

int count\_one\_child\_nodes(NODE \*tree)

{

if(tree==NULL)

return 0;

else if (((tree->left)==NULL && (tree->right)!=NULL)||((tree->left)!=NULL && (tree->right)==NULL))

return 1;

else

return count\_one\_child\_nodes(tree->left)+count\_one\_child\_nodes(tree->right);

}

int main()

{

NODE \*tree=NULL;

int x, total\_nodes, total\_leaf\_nodes, total\_one\_child\_nodes;

char y='Y';

do

{

printf("Enter an element: ");

scanf("%d", &x);

tree=insert(tree, x);

printf("Do you want to enter more elements: ");

scanf("%s", &y);

} while (y=='Y' || y=='y');

total\_nodes=count\_nodes(tree);

printf("Total number of nodes: %d\n", total\_nodes);

total\_leaf\_nodes=count\_leaf\_nodes(tree);

printf("Total number of leaf nodes: %d\n", total\_leaf\_nodes);

total\_one\_child\_nodes=count\_one\_child\_nodes(tree);

printf("Total number of one child nodes: %d\n", total\_one\_child\_nodes);

return 0;

}

OUTPUT:

Enter an element: 5

Do you want to enter more elements: y

Enter an element: 3

Do you want to enter more elements: y

Enter an element: 4

Do you want to enter more elements: y

Enter an element: 2

Do you want to enter more elements: y

Enter an element: 8

Do you want to enter more elements: y

Enter an element: 1

Do you want to enter more elements: y

Enter an element: 6

Do you want to enter more elements: y

Enter an element: 7

Do you want to enter more elements: y

Enter an element: 9

Do you want to enter more elements: n

Total number of nodes: 9

Total number of leaf nodes: 4

Total number of one child nodes: 2