#include <iostream>

using namespace std;

typedef struct node

{

int data;

struct node \*lchild;

struct node \*rchild;

}NODE, \*PNODE, \*\*PPNODE;

void Insert(PPNODE Head, int no)

{

PNODE temp = \*Head;

PNODE newn = new NODE;

newn->data = no;

newn->lchild = NULL;

newn->rchild = NULL;

if(\*Head == NULL)

{

\*Head = newn;

}

else

{

while(1) // Unconditional Loop for( ; ; )

{

if(temp->data == no)

{

cout<<"Duplicate node\n";

delete newn;

break;

}

else if(no < (temp->data)) // lahan data

{

if(temp -> lchild == NULL)

{

temp->lchild = newn;

break;

}

temp = temp -> lchild;

}

else if(no > (temp->data)) // motha data

{

if(temp -> rchild == NULL)

{

temp->rchild = newn;

break;

}

temp = temp->rchild;

}

}

}

}

bool Search(PNODE Head , int no)

{

if(Head == NULL) // if treee is empty

{

return false;

}

else // tree contains atleast one node

{

while(Head != NULL)

{

if(Head -> data == no) // node sapadala

{

break;

}

else if(no > (Head -> data))

{

Head = Head->rchild;

}

else if(no < (Head -> data))

{

Head = Head -> lchild;

}

}

if(Head == NULL)

{

return false;

}

else

{

return true;

}

}

}

int Count(PNODE Head)

{

static int iCnt = 0;

if(Head != NULL)

{

iCnt++;

Count(Head -> lchild);

Count(Head -> rchild);

}

return iCnt;

}

int main()

{

int no = 0, iret = 0;

bool bret = false;

PNODE first = NULL;

Insert(&first,51);

Insert(&first,21);

Insert(&first,101);

cout<<"Enter number to search\n";

cin>>no;

bret = Search(first,no);

if(bret == true)

{

cout<<"Data is there\n";

}

else

{

cout<<"Data is not there\n";

}

iret = Count(first);

cout<<"Number of nodes are "<<iret<<"\n";

return 0;

}

// Data to be inserted : 51, 21, 101, 55, 75, 20, 105, 16