

Athira Bijy
SIMCA
Roll no: 4.

1. Write a program to create binary search tree and determine the following.
- a. number of nodes in the tree
 - b. sum of all nodes
 - c. left or right from root is bit.

Algorithm

step 1: start

step 2: Declare a node in info, left, right, sum left and sum right

step 3: create a new node. set left and right is NULL. create it.

step 4: ~~if the root is null. add values to the nodes of left subtree.~~
calculate sum will calculate the sum of all node in binary tree.

~~later~~

a) if it is empty
if (root == NULL).

print null

return 0

b). calculate left subtree.

c) calculate right subtree

d) ~~calculate~~ calculate the left and right

step 5: Adding values to the trees.

step 6: Display all nodes are given by.

step 7: Display the sum of all nodes.

step 8: stop.

PROGRAM

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

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

```
struct node
```

```
{
```

```
    int info;
```

```
    struct node *left, *right, *sumLeft, *sumRight;
```

```
};
```

```
struct node *createnode(int key)
```

```
{
```

```
    struct node *newnode = (struct node*)malloc(sizeof(struct node));
```

```
    newnode->info = key;
```

```
    newnode->left = NULL;
```

```
    newnode->right = NULL;
```

```
    return(newnode);
```

```
}
```

```
static int count = 0;
```

```
int countnodes(struct node *root)
```

```
{
```

```
    if(root != NULL)
```

```
    {
```

```
        countnodes(root->left);
```

```
        count++;
```

```

        countnodes(root->right);
    }
    return count;
}

int main()
{
    struct node *newnode = createnode(25);
    newnode->left = createnode(28);
    newnode->right = createnode(19);
    newnode->left->left = createnode(18);
    newnode->left->right = createnode(92);
    newnode->right->left = createnode(14);
    newnode->right->right = createnode(56);

    printf("Number of nodes in tree 1 = %d ",countnodes(newnode));
    printf("\n");

    count = 0;

    struct node *node = createnode(1);
    node->right = createnode(2);
    node->right->right = createnode(3);
    node->right->right->right = createnode(4);
    node->right->right->right->right = createnode(5);

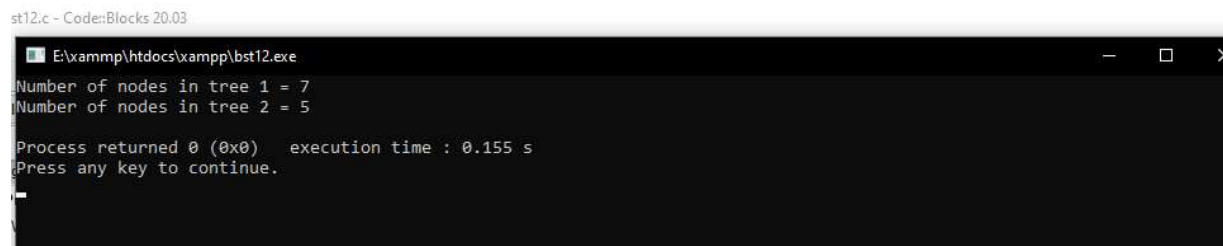
    printf("Number of nodes in tree 2 = %d ",countnodes(node));
    printf("\n");

    count = 0;

```

```
    printf("Sum of all nodes of binary tree: %d", calculateSum(root));  
  
    return 0;  
}
```

Output:



```
st12.c - Code::Blocks 20.03  
E:\xampp\htdocs\xampp\bst12.exe  
Number of nodes in tree 1 = 7  
Number of nodes in tree 2 = 5  
  
Process returned 0 (0x0)   execution time : 0.155 s  
Press any key to continue.  
_
```