DATA STRUCTURE AND ALGORITHM

EXERCISE 9-Binary and Binary Search Tree

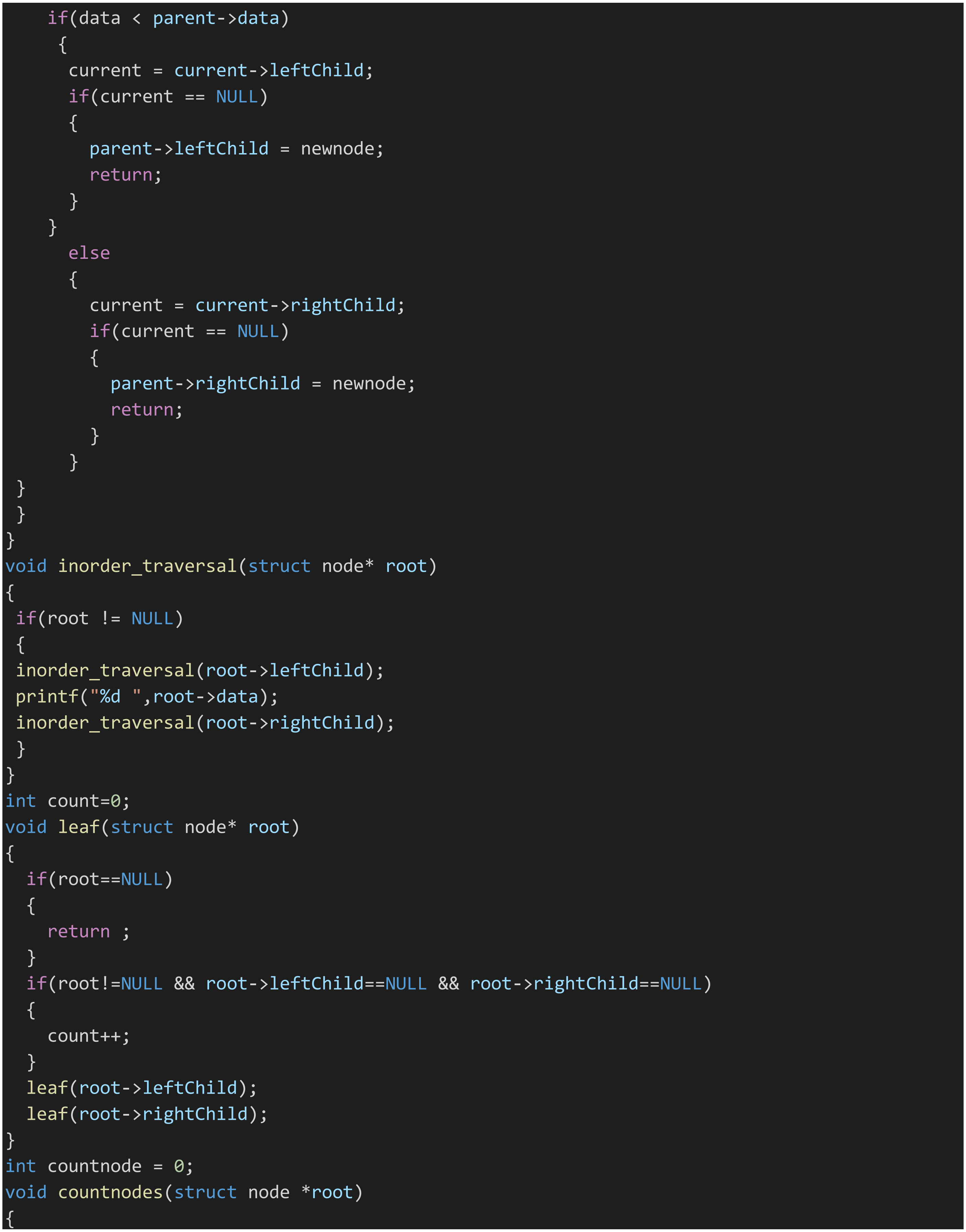
Name: A H K Swarun

Reg no: 21BPS1252

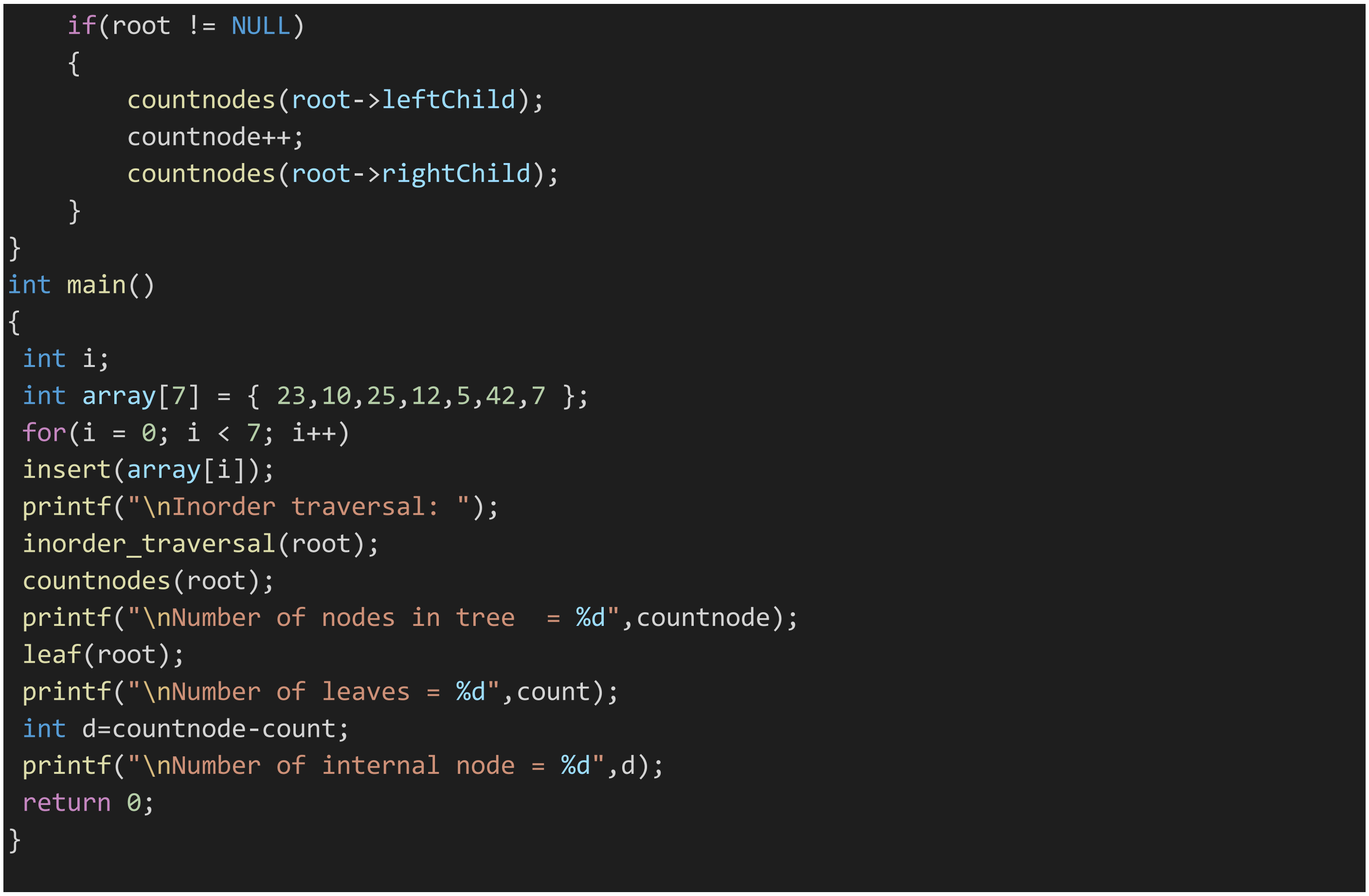
Q. Create a BST and count the leaf nodes and internal Nodes in the tree

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct node {
 int data;
 struct node *leftChild;
 struct node *rightChild;
};
struct node *root = NULL;
void insert(int data)
{
 struct node *newnode = (struct node*) malloc(sizeof(struct node));
 struct node *current;
 struct node *parent;
 newnode->data = data;
 newnode->leftChild = NULL;
 newnode->rightChild = NULL;
 //if tree is empty
 if(root == NULL)
 {
  root = newnode;
 }
 else
 {
  current = root;
  parent = NULL;
  while(1)
  {
    parent = current;
```

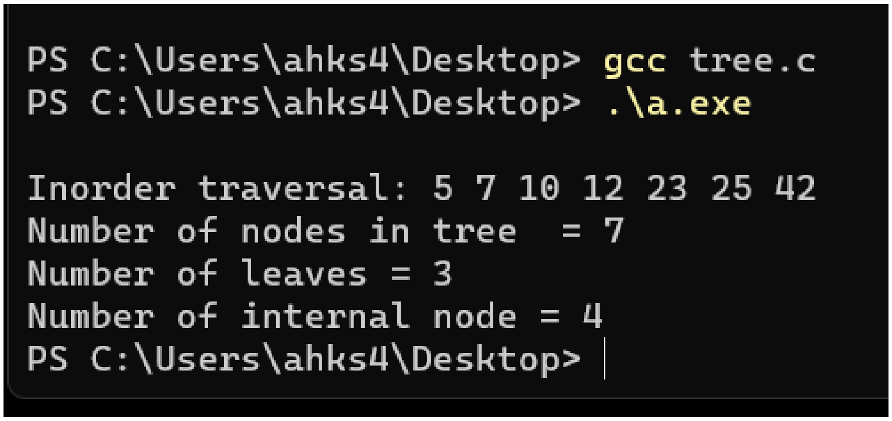


```
    if(data < parent->data)
     {
      current = current->leftChild;
      if(current == NULL)
      {
        parent->leftChild = newnode;
        return;
      }
    }
      else
      {
        current = current->rightChild;
        if(current == NULL)
        {
          parent->rightChild = newnode;
          return;
        }
      }
 }
 }
}
void inorder_traversal(struct node* root)
{
 if(root != NULL)
 {
 inorder_traversal(root->leftChild);
 printf("%d ",root->data);
 inorder_traversal(root->rightChild);
 }
}
int count=0;
void leaf(struct node* root)
{
  if(root==NULL)
  {
    return ;
  }
  if(root!=NULL && root->leftChild==NULL && root->rightChild==NULL)
  {
    count++;
  }
  leaf(root->leftChild);
  leaf(root->rightChild);
}
int countnode = 0;
void countnodes(struct node *root)
{
```



```
    if(root != NULL)
    {
        countnodes(root->leftChild);
        countnode++;
        countnodes(root->rightChild);
    }
}
int main()
{
 int i;
 int array[7] = { 23,10,25,12,5,42,7 };
 for(i = 0; i < 7; i++)
 insert(array[i]);
 printf("\nInorder traversal: ");
 inorder_traversal(root);
 countnodes(root);
 printf("\nNumber of nodes in tree  = %d",countnode);
 leaf(root);
 printf("\nNumber of leaves = %d",count);
 int d=countnode-count;
 printf("\nNumber of internal node = %d",d);
 return 0;
}
```

OUTPUT:

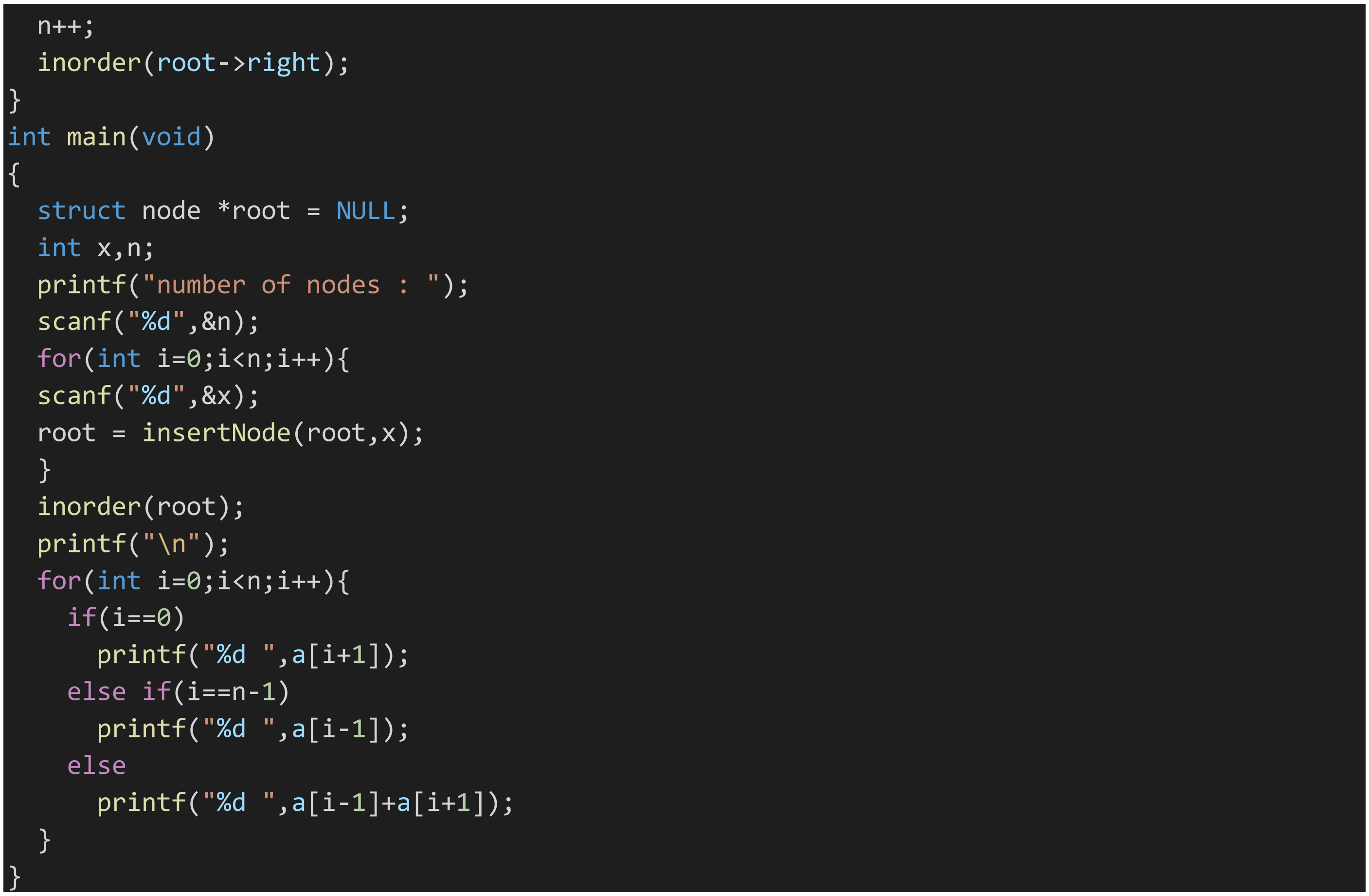


```
PS C:\Users\ahks4\Desktop> gcc tree.c
PS C:\Users\ahks4\Desktop> .\a.exe

Inorder traversal: 5 7 10 12 23 25 42
Number of nodes in tree  = 7
Number of leaves = 3
Number of internal node = 4
PS C:\Users\ahks4\Desktop>
```

Q. Create a Binary Tree with N Nodes. Write a C Program to replace each node in the Binary tree with the sum of inorder predecessor and inorder successor.

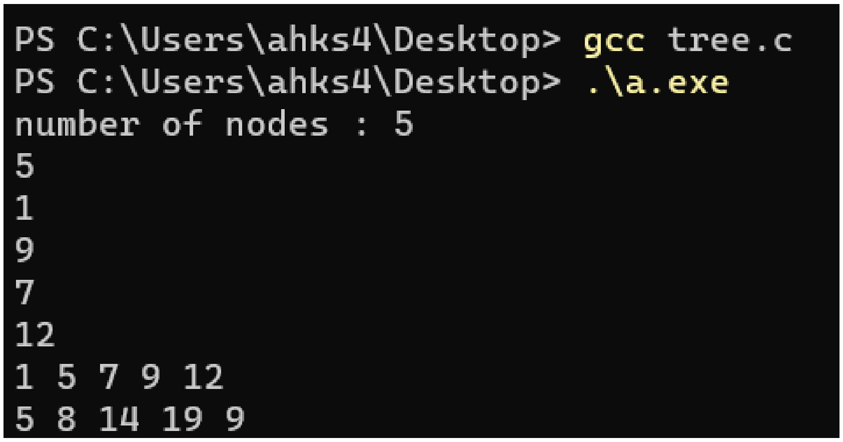
Code:

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

int n=0;
int a[50];

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

struct node *getNode(int val)
{
  struct node *newNode;
  newNode = malloc(sizeof(struct node));
  newNode->key   = val;
  newNode->left  = NULL;
  newNode->right = NULL;
  return newNode;
}

struct node *insertNode(struct node *root, int val)
{
  if(root == NULL)
    return getNode(val);
  if(root->key < val)
    root->right = insertNode(root->right,val);
  if(root->key > val)
    root->left = insertNode(root->left,val);
    return root;
}
void inorder(struct node *root)
{
  if(root == NULL)
  return;
  inorder(root->left);
  printf("%d ",root->key);
  a[n]=root->key;
```



```
  n++;
  inorder(root->right);
}
int main(void)
{
  struct node *root = NULL;
  int x,n;
  printf("number of nodes : ");
  scanf("%d",&n);
  for(int i=0;i<n;i++){
  scanf("%d",&x);
  root = insertNode(root,x);
  }
  inorder(root);
  printf("\n");
  for(int i=0;i<n;i++){
    if(i==0)
      printf("%d ",a[i+1]);
    else if(i==n-1)
      printf("%d ",a[i-1]);
    else
      printf("%d ",a[i-1]+a[i+1]);
  }
}
```

Output:



```
PS C:\Users\ahks4\Desktop> gcc tree.c
PS C:\Users\ahks4\Desktop> .\a.exe
number of nodes : 5
5
1
9
7
12
1 5 7 9 12
5 8 14 19 9
```