

1번.

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#define max(a,b) (((a)>(b)) ? (a):(b))
typedef struct TreeNode {
    int data;
    struct TreeNode *left, *right;
} TreeNode;

int get_node_count(TreeNode *node)
{
    int count = 0;
    if(node != NULL)
        count = 1 + get_node_count(node->left) +
            get_node_count(node->right);
    return count;
}

int get_leaf_count(TreeNode *node)
{
    int count = 0;
    if(node != NULL){
        if(node->left == NULL && node->right == NULL)
            return 1;
        else
            count = get_leaf_count(node->left) + get_leaf_count(node->right);
    }
    return count;
}

int get_height(TreeNode *node)
{
    int height = 0;
    if(node != NULL)
        height = 1 + max(get_height(node->left),
            get_height(node->right));
    return height;
}

int main(void)
{
    TreeNode *n1, *n2, *n3, *n4, *n5, *n6, *n7;
    n1 = (TreeNode *)malloc(sizeof(TreeNode));
    n2 = (TreeNode *)malloc(sizeof(TreeNode));
    n3 = (TreeNode *)malloc(sizeof(TreeNode));
    n4 = (TreeNode *)malloc(sizeof(TreeNode));
    n5 = (TreeNode *)malloc(sizeof(TreeNode));
    n6 = (TreeNode *)malloc(sizeof(TreeNode));
    n7 = (TreeNode *)malloc(sizeof(TreeNode));

    n1->data = 15;
    n1->left = n2;
    n1->right = n4;
    n2->data = 4;
    n2->left = n3;
    n2->right = NULL;
```

```

n3->data=1;
n3->left =NULL;
n3->right=NULL;
n4->data=20;
n4->left=n5;
n4->right=n6;
n5->data =18;
n5->left=n7;
n5->right =NULL;
n6->data=25;
n6->left=NULL;
n6->right=NULL;
n7->data=16;
n7->left=NULL;
n7->right=NULL;
TreeNode *root =n1;
printf("높이=%d\n", get_height(root));
printf("리프=%d\n", get_leaf_count(root));
printf("개수=%d\n", get_node_count(root));
}

```

2번.

1)

```

void isBalanced(TreeNode*node)
{
int left_height =get_height(node->left);
int right_height =get_height(node->right);
int height;
if(left_height>right_height)
height=left_height-right_height;
else
height=right_height-left_height;
if (height <=1 && height >=-1)
printf("균형트리입니다.");
else
printf("균형트리가 아닙니다.");
}

```

2)

```

int get_sum(TreeNode*node) {
if (node ==NULL)
return 0;
int sum =node->data;
sum =sum +get_sum(node->left) +get_sum(node->right);
return sum;
}

```

3.

```
문제   출력   디버그 콘솔   터미널

균형트리입니다.
노드의 합은 99입니다.
dongsebi@seodongseob-ui-MacBookAir 자료구조2 %
```

4.

```
문제   출력   디버그 콘솔   터미널

균형트리가 아닙니다.
노드의 합은 98입니다.
dongsebi@seodongseob-ui-MacBookAir 자료구조2 %
```

전체코드(3,4번)

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#define max(a,b) (((a)>(b))?(a):(b))
typedef struct TreeNode {
    int data;
    struct TreeNode *left, *right;
} TreeNode;

int get_node_count(TreeNode *node)
{
    int count = 0;
    if(node != NULL)
        count = 1 + get_node_count(node->left) +
            get_node_count(node->right);
    return count;
}

int get_leaf_count(TreeNode *node)
{
    int count = 0;
    if(node != NULL){
        if(node->left == NULL && node->right == NULL)
            return 1;
        else
            count = get_leaf_count(node->left) + get_leaf_count(node->right);
    }
}
```

```

}
return count;
}
int get_height(TreeNode *node)
{
int height =0;
if(node !=NULL)
height =1 +max(get_height(node->left),
get_height(node->right));
return height;
}
void isBalanced(TreeNode*node)
{
int left_height =get_height(node->left);
int right_height =get_height(node->right);
int height;
if(left_height>right_height)
height=left_height-right_height;
else
height=right_height-left_height;
if (height <=1 && height >=-1)
printf("균형트리입니다.");
else
printf("균형트리가 아닙니다.");
}
int get_sum(TreeNode*node) {
if (node ==NULL)
return 0;
int sum =node->data;
sum =sum +get_sum(node->left) +get_sum(node->right);
return sum;
}
int main(void)
{
TreeNode *n1, *n2, *n3, *n4, *n5, *n6;
n1=(TreeNode *)malloc(sizeof(TreeNode));
n2=(TreeNode *)malloc(sizeof(TreeNode));
n3=(TreeNode *)malloc(sizeof(TreeNode));
n4=(TreeNode *)malloc(sizeof(TreeNode));
n5=(TreeNode *)malloc(sizeof(TreeNode));
n6=(TreeNode *)malloc(sizeof(TreeNode));
n1->data =15;
n1->left=n2;
n1->right=n3;
n2->data =4;
n2->left =NULL;
n2->right =NULL;
n3->data=20;
n3->left=n4;
n3->right=n5;
n4->data =18;
n4->left=n6;
n4->right =NULL;
n5->data=25;
n5->left=NULL;
n5->right=NULL;
n6->data=16;
n6->left=NULL;
n6->right=NULL;
TreeNode *root =n1;
isBalanced(root);

```

```
printf("\n");  
printf("노드의 합은 %d 입니다.", get_sum(root));  
}
```