Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee (이상훈)

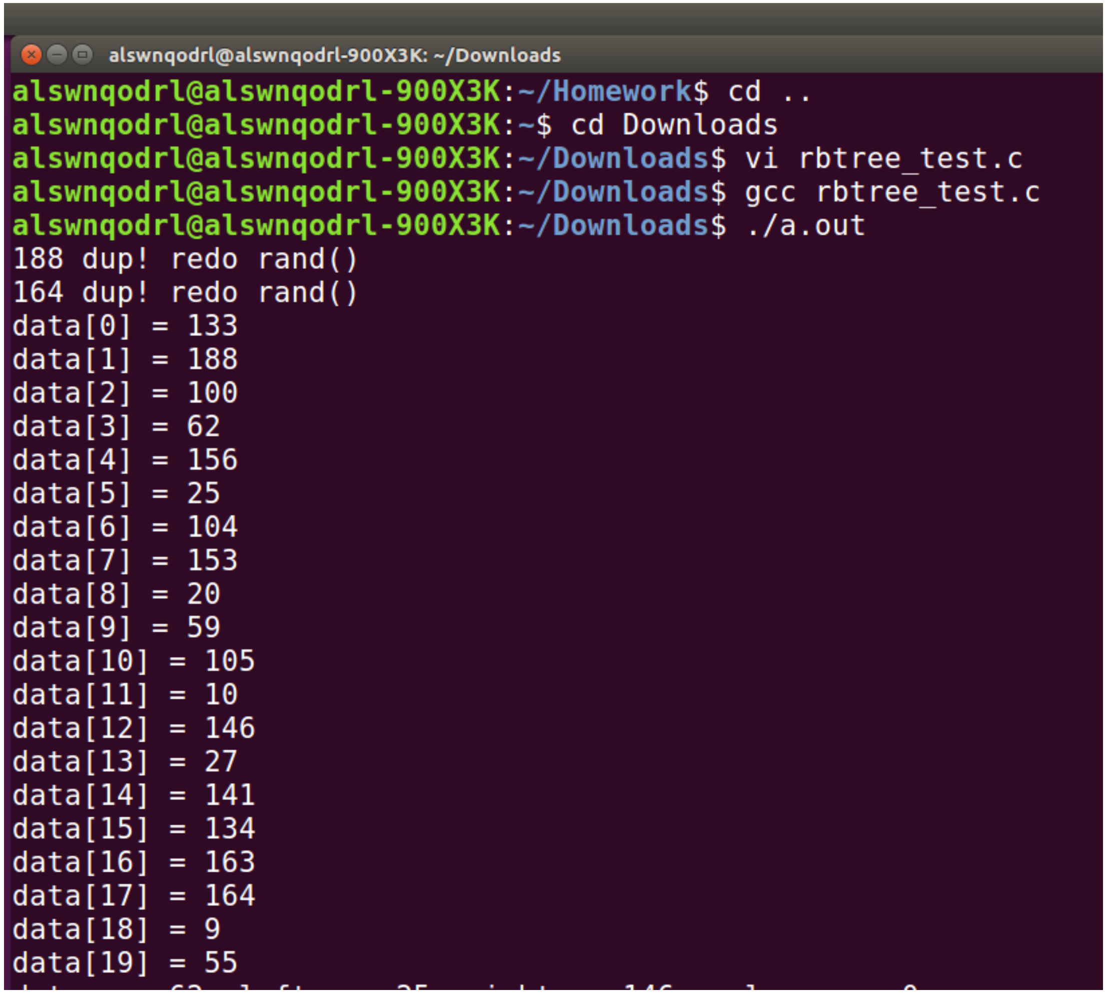
gcccompil3r@gmail.com

학생-김민주

alswnqodrl@naver.com

01

15일차 내용 복습 (rb tree)

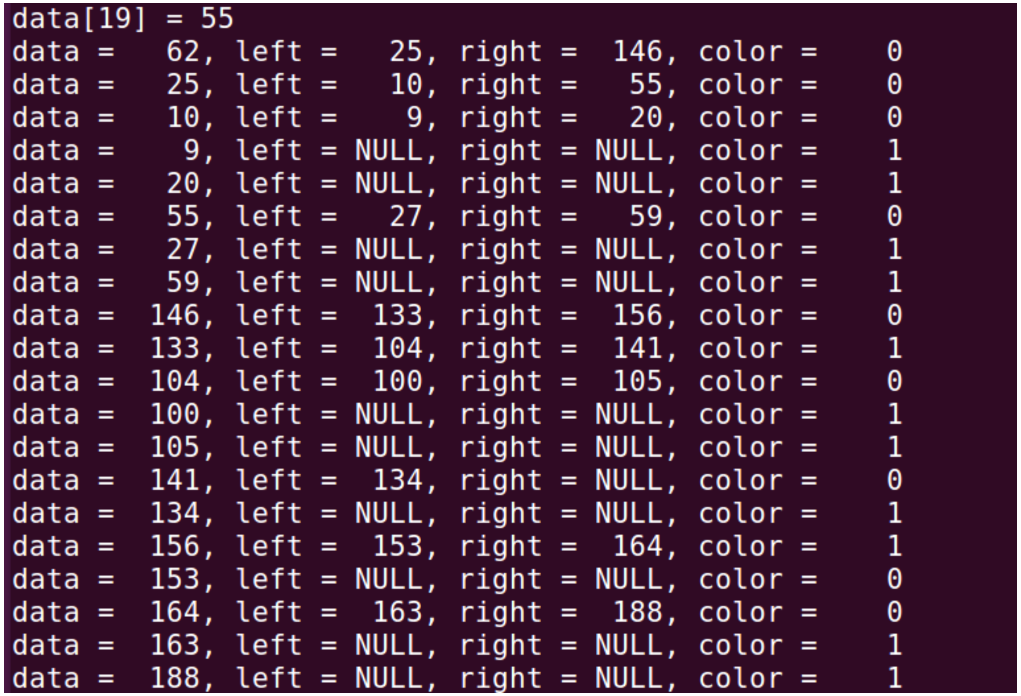


alswnqodrl@alswnqodrl-900X3K: ~/Downloads

```
alswnqodrl@alswnqodrl-900X3K:~/Homework$ cd ..
alswnqodrl@alswnqodrl-900X3K:~$ cd Downloads
alswnqodrl@alswnqodrl-900X3K:~/Downloads$ vi rbtree_test.c
alswnqodrl@alswnqodrl-900X3K:~/Downloads$ gcc rbtree_test.c
alswnqodrl@alswnqodrl-900X3K:~/Downloads$ ./a.out
188 dup! redo rand()
164 dup! redo rand()
data[0] = 133
data[1] = 188
data[2] = 100
data[3] = 62
data[4] = 156
data[5] = 25
data[6] = 104
data[7] = 153
data[8] = 20
data[9] = 59
data[10] = 105
data[11] = 10
data[12] = 146
data[13] = 27
data[14] = 141
data[15] = 134
data[16] = 163
data[17] = 164
data[18] = 9
data[19] = 55
```

01

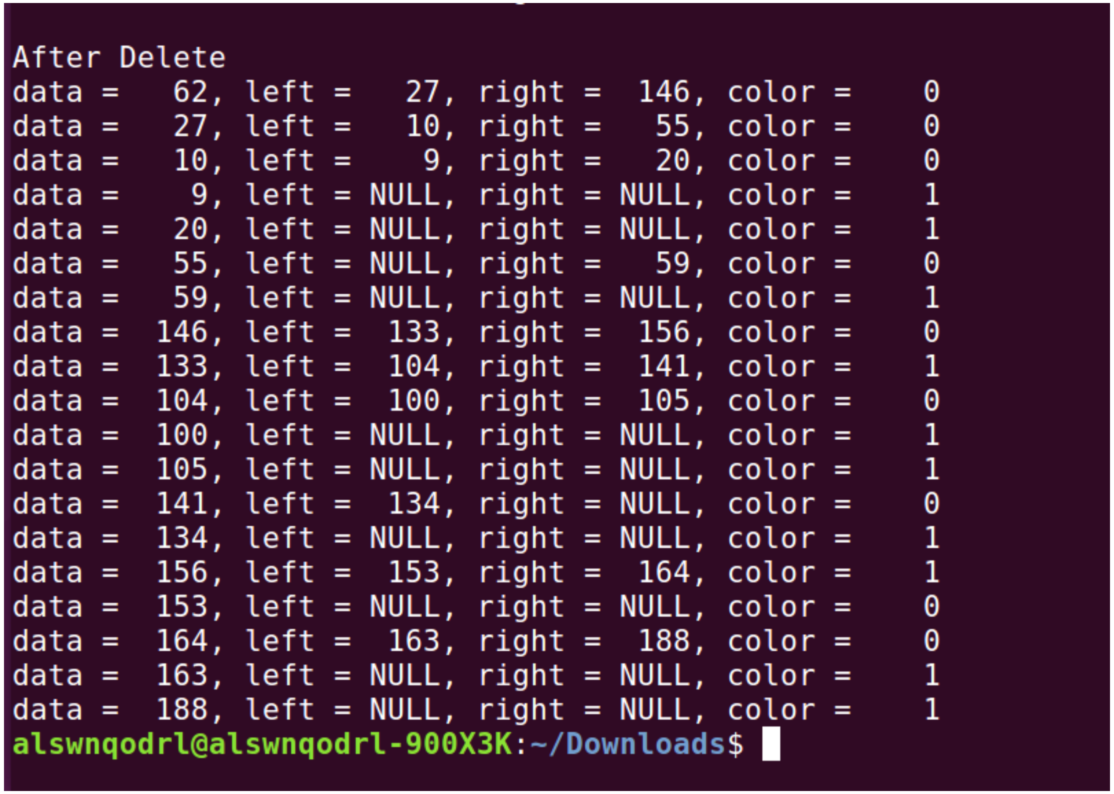
15일차 내용 복습 (rb
tree)



```
data[19] = 55
data =   62, left =   25, right =  146, color =    0
data =   25, left =   10, right =   55, color =    0
data =   10, left =    9, right =   20, color =    0
data =    9, left = NULL, right = NULL, color =    1
data =   20, left = NULL, right = NULL, color =    1
data =   55, left =   27, right =   59, color =    0
data =   27, left = NULL, right = NULL, color =    1
data =   59, left = NULL, right = NULL, color =    1
data =  146, left =  133, right =  156, color =    0
data =  133, left =  104, right =  141, color =    1
data =  104, left =  100, right =  105, color =    0
data =  100, left = NULL, right = NULL, color =    1
data =  105, left = NULL, right = NULL, color =    1
data =  141, left =  134, right = NULL, color =    0
data =  134, left = NULL, right = NULL, color =    1
data =  156, left =  153, right =  164, color =    1
data =  153, left = NULL, right = NULL, color =    0
data =  164, left =  163, right =  188, color =    0
data =  163, left = NULL, right = NULL, color =    1
data =  188, left = NULL, right = NULL, color =    1
```

01

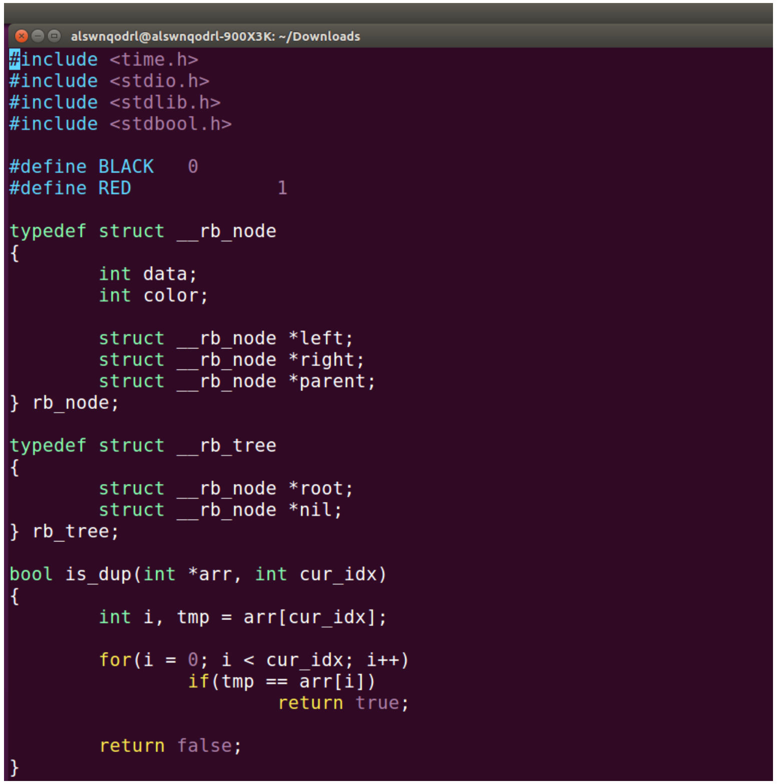
15일차 내용 복습 (rb tree)



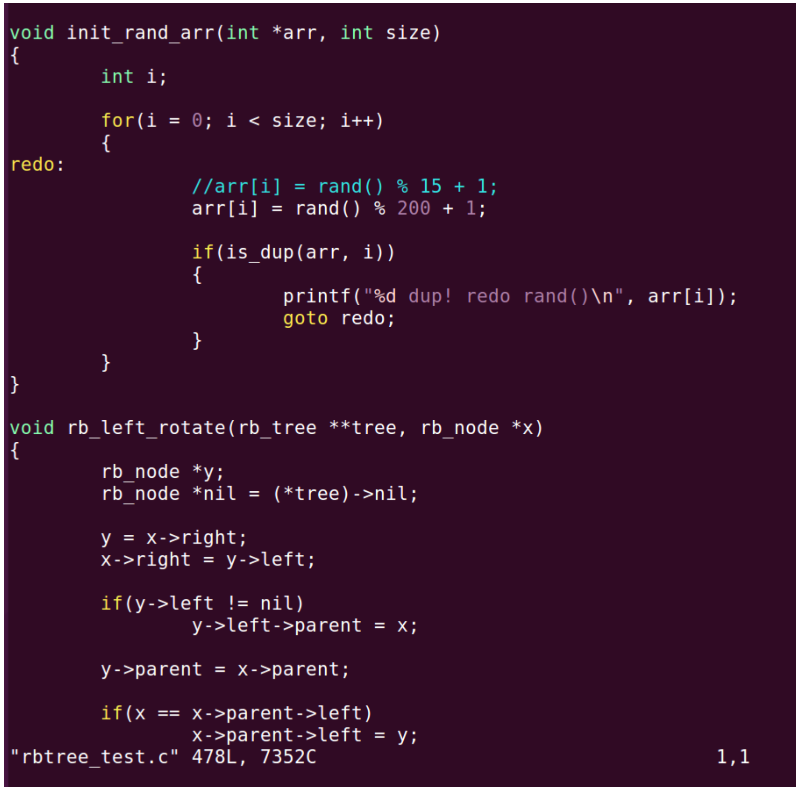
```
After Delete
data =   62, left =   27, right =  146, color =    0
data =   27, left =   10, right =   55, color =    0
data =   10, left =    9, right =   20, color =    0
data =    9, left = NULL, right = NULL, color =    1
data =   20, left = NULL, right = NULL, color =    1
data =   55, left = NULL, right =   59, color =    0
data =   59, left = NULL, right = NULL, color =    1
data =  146, left =  133, right =  156, color =    0
data =  133, left =  104, right =  141, color =    1
data =  104, left =  100, right =  105, color =    0
data =  100, left = NULL, right = NULL, color =    1
data =  105, left = NULL, right = NULL, color =    1
data =  141, left =  134, right = NULL, color =    0
data =  134, left = NULL, right = NULL, color =    1
data =  156, left =  153, right =  164, color =    1
data =  153, left = NULL, right = NULL, color =    0
data =  164, left =  163, right =  188, color =    0
data =  163, left = NULL, right = NULL, color =    1
data =  188, left = NULL, right = NULL, color =    1
alswnqodrl@alswnqodrl-900X3K:~/Downloads$
```

02

15일차 내용 복습 (rb tree)



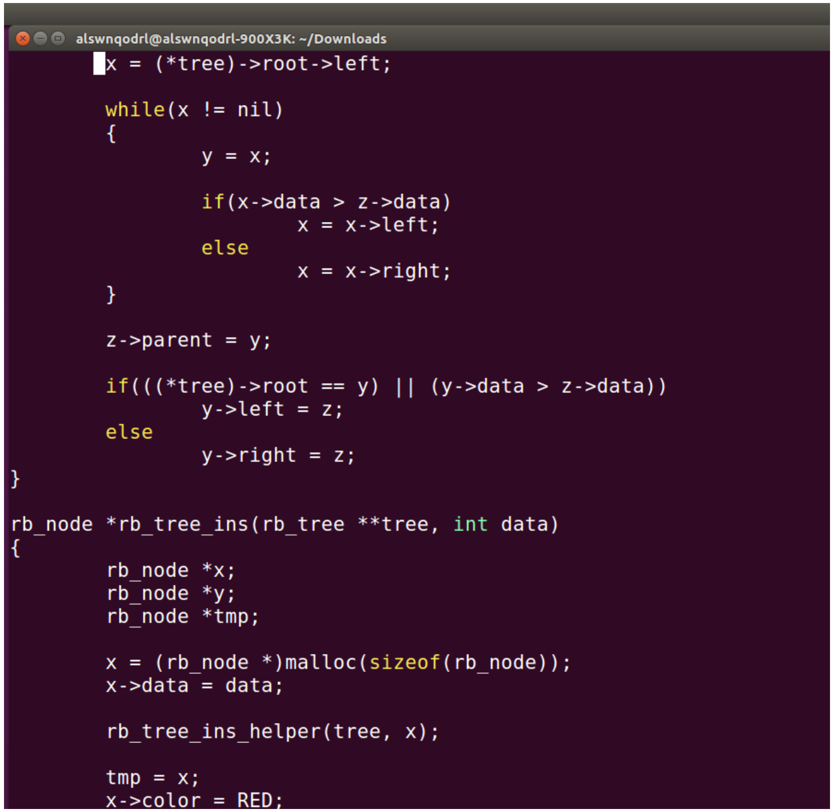
```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define BLACK   0
#define RED             1

typedef struct __rb_node
{
        int data;
        int color;

        struct __rb_node *left;
        struct __rb_node *right;
        struct __rb_node *parent;
} rb_node;

typedef struct __rb_tree
{
        struct __rb_node *root;
        struct __rb_node *nil;
} rb_tree;

bool is_dup(int *arr, int cur_idx)
{
        int i, tmp = arr[cur_idx];

        for(i = 0; i < cur_idx; i++)
                if(tmp == arr[i])
                        return true;

        return false;
}
```



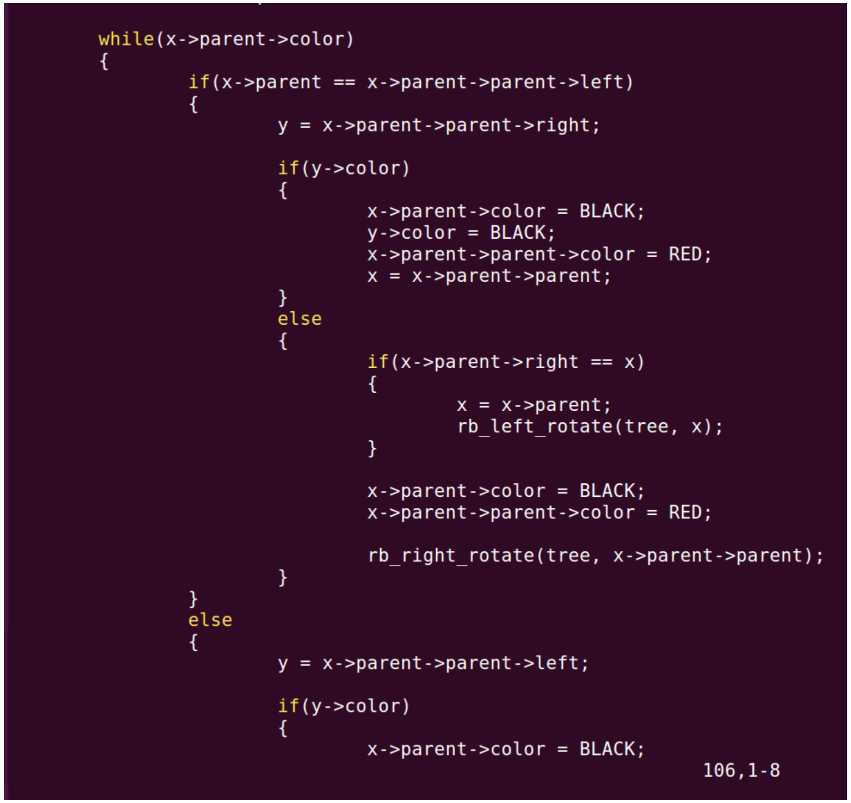
```
void init_rand_arr(int *arr, int size)
{
        int i;

        for(i = 0; i < size; i++)
        {
redo:
                //arr[i] = rand() % 15 + 1;
                arr[i] = rand() % 200 + 1;

                if(is_dup(arr, i))
                {
                        printf("%d dup! redo rand()\n", arr[i]);
                        goto redo;
                }
        }
}

void rb_left_rotate(rb_tree **tree, rb_node *x)
{
        rb_node *y;
        rb_node *nil = (*tree)->nil;

        y = x->right;
        x->right = y->left;

        if(y->left != nil)
                y->left->parent = x;

        y->parent = x->parent;

        if(x == x->parent->left)
                x->parent->left = y;
"rbtree_test.c" 478L, 7352C                                   1,1
```

02

15일차 내용 복습 (rb tree)



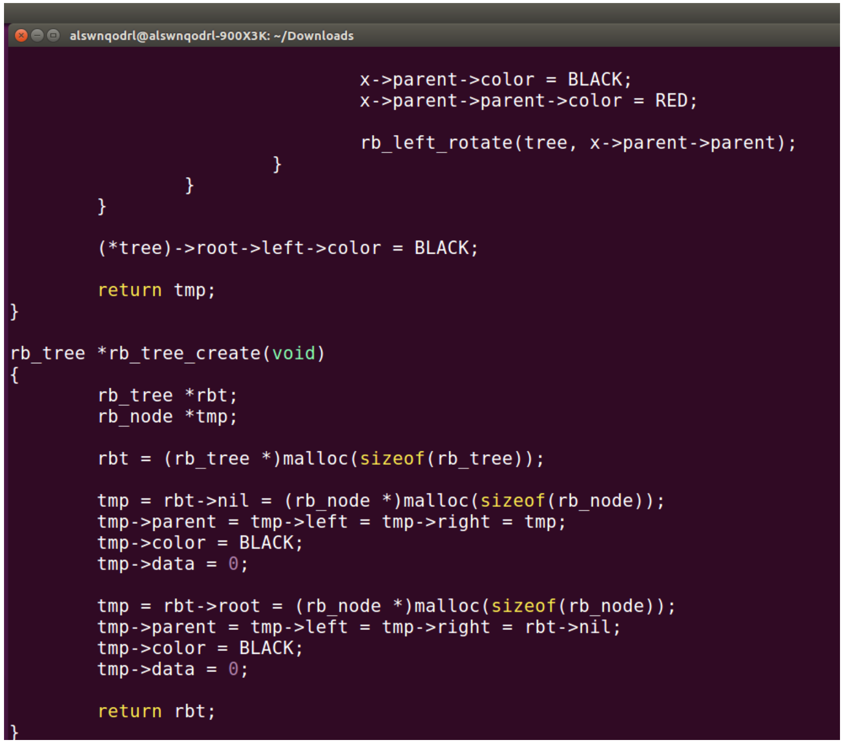
```
        x = (*tree)->root->left;

        while(x != nil)
        {
                y = x;

                if(x->data > z->data)
                        x = x->left;
                else
                        x = x->right;
        }

        z->parent = y;

        if(((*tree)->root == y) || (y->data > z->data))
                y->left = z;
        else
                y->right = z;
}

rb_node *rb_tree_ins(rb_tree **tree, int data)
{
        rb_node *x;
        rb_node *y;
        rb_node *tmp;

        x = (rb_node *)malloc(sizeof(rb_node));
        x->data = data;

        rb_tree_ins_helper(tree, x);

        tmp = x;
        x->color = RED;
```



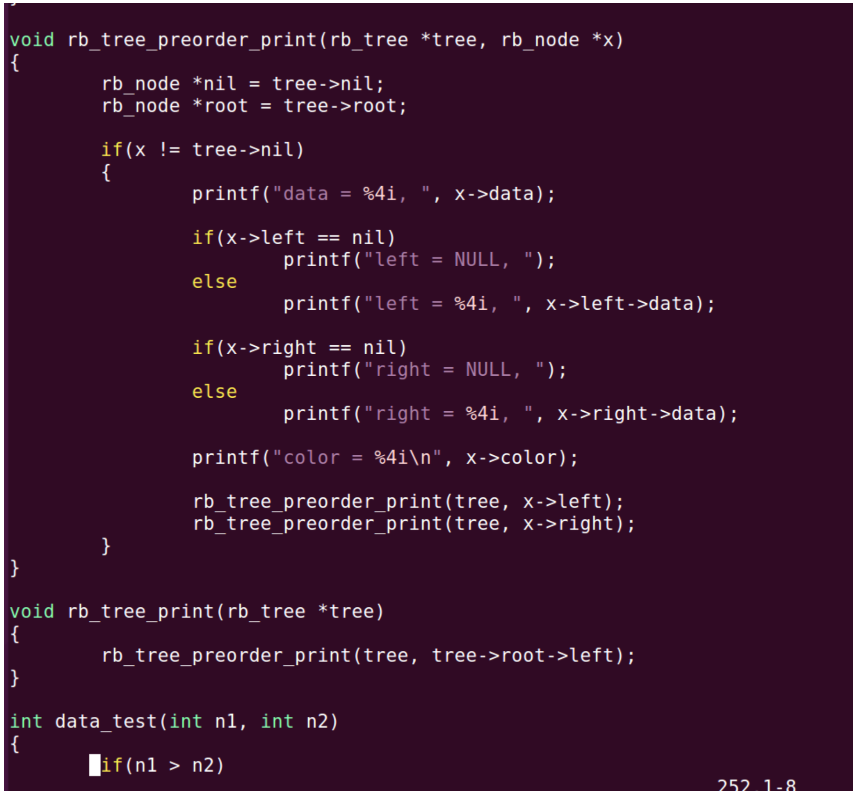
```
        while(x->parent->color)
        {
                if(x->parent == x->parent->parent->left)
                {
                        y = x->parent->parent->right;

                        if(y->color)
                        {
                                x->parent->color = BLACK;
                                y->color = BLACK;
                                x->parent->parent->color = RED;
                                x = x->parent->parent;
                        }
                        else
                        {
                                if(x->parent->right == x)
                                {
                                        x = x->parent;
                                        rb_left_rotate(tree, x);
                                }

                                x->parent->color = BLACK;
                                x->parent->parent->color = RED;

                                rb_right_rotate(tree, x->parent->parent);
                        }
                }
                else
                {
                        y = x->parent->parent->left;

                        if(y->color)
                        {
                                x->parent->color = BLACK;
                                                                        106,1-8
```

02

15일차 내용 복습 (rb tree)



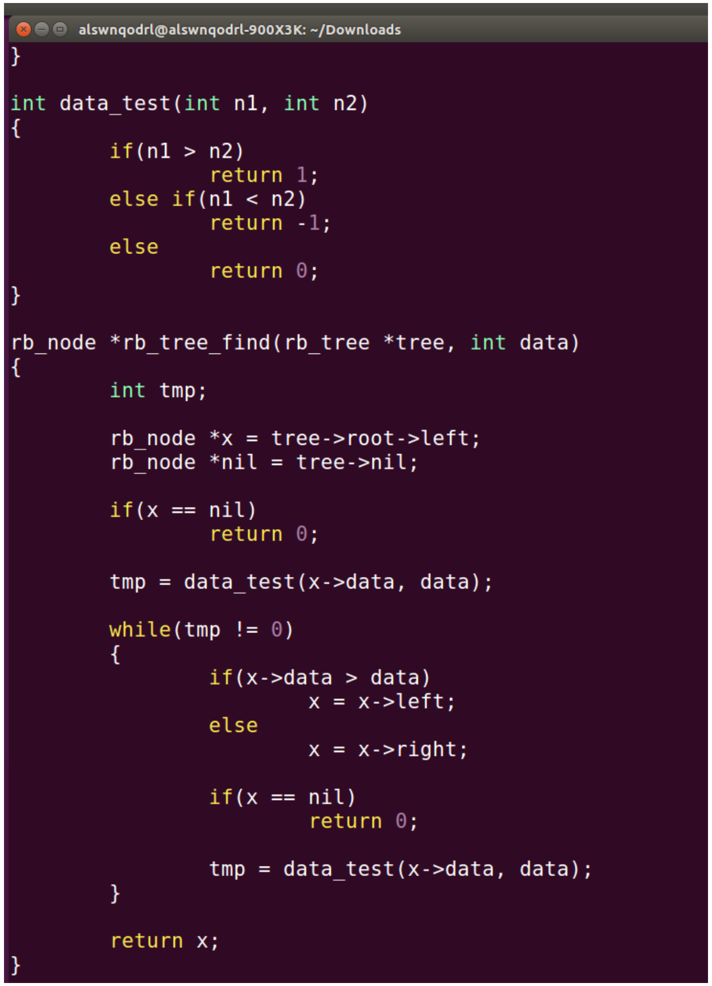
```
                                x->parent->color = BLACK;
                                x->parent->parent->color = RED;

                                rb_left_rotate(tree, x->parent->parent);
                        }
                }
        }

        (*tree)->root->left->color = BLACK;

        return tmp;
}

rb_tree *rb_tree_create(void)
{
        rb_tree *rbt;
        rb_node *tmp;

        rbt = (rb_tree *)malloc(sizeof(rb_tree));

        tmp = rbt->nil = (rb_node *)malloc(sizeof(rb_node));
        tmp->parent = tmp->left = tmp->right = tmp;
        tmp->color = BLACK;
        tmp->data = 0;

        tmp = rbt->root = (rb_node *)malloc(sizeof(rb_node));
        tmp->parent = tmp->left = tmp->right = rbt->nil;
        tmp->color = BLACK;
        tmp->data = 0;

        return rbt;
}
```



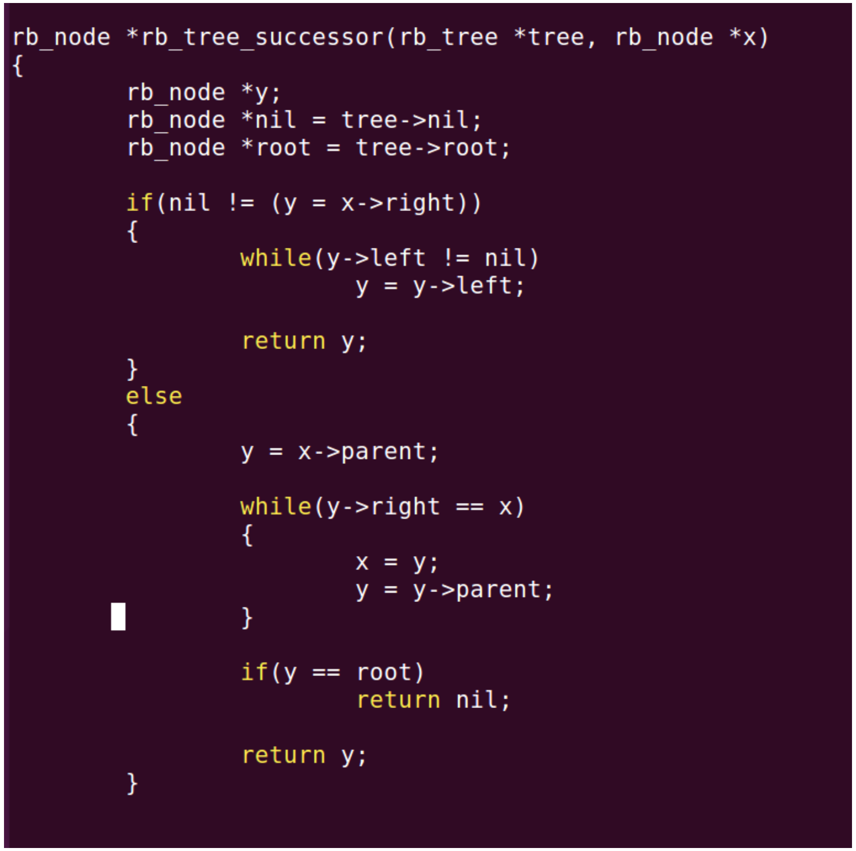
```
void rb_tree_preorder_print(rb_tree *tree, rb_node *x)
{
        rb_node *nil = tree->nil;
        rb_node *root = tree->root;

        if(x != tree->nil)
        {
                printf("data = %4i, ", x->data);

                if(x->left == nil)
                        printf("left = NULL, ");
                else
                        printf("left = %4i, ", x->left->data);

                if(x->right == nil)
                        printf("right = NULL, ");
                else
                        printf("right = %4i, ", x->right->data);

                printf("color = %4i\n", x->color);

                rb_tree_preorder_print(tree, x->left);
                rb_tree_preorder_print(tree, x->right);
        }
}

void rb_tree_print(rb_tree *tree)
{
        rb_tree_preorder_print(tree, tree->root->left);
}

int data_test(int n1, int n2)
{
        if(n1 > n2)
                                                                252 1-8
```

02

15일차 내용 복습 (rb tree)



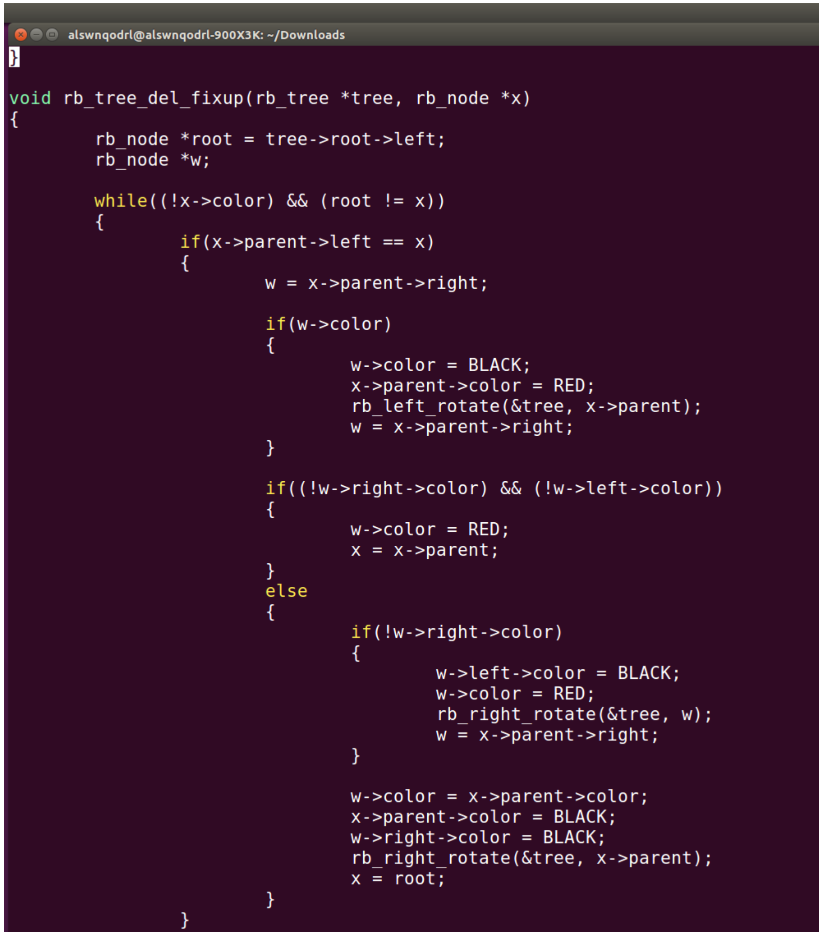
```
}

int data_test(int n1, int n2)
{
        if(n1 > n2)
                return 1;
        else if(n1 < n2)
                return -1;
        else
                return 0;
}

rb_node *rb_tree_find(rb_tree *tree, int data)
{
        int tmp;

        rb_node *x = tree->root->left;
        rb_node *nil = tree->nil;

        if(x == nil)
                return 0;

        tmp = data_test(x->data, data);

        while(tmp != 0)
        {
                if(x->data > data)
                        x = x->left;
                else
                        x = x->right;

                if(x == nil)
                        return 0;

                tmp = data_test(x->data, data);
        }

        return x;
}
```



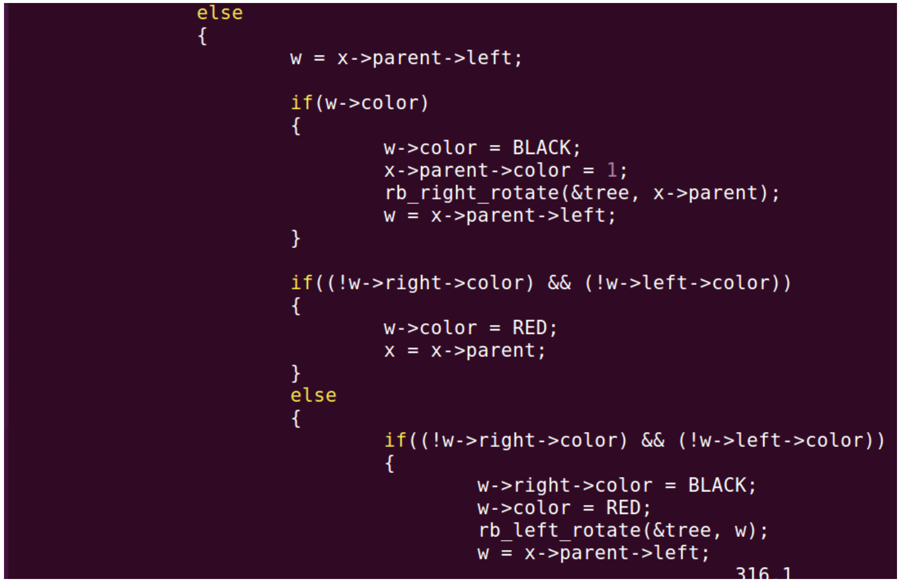
```
rb_node *rb_tree_successor(rb_tree *tree, rb_node *x)
{
        rb_node *y;
        rb_node *nil = tree->nil;
        rb_node *root = tree->root;

        if(nil != (y = x->right))
        {
                while(y->left != nil)
                        y = y->left;

                return y;
        }
        else
        {
                y = x->parent;

                while(y->right == x)
                {
                        x = y;
                        y = y->parent;
                }

                if(y == root)
                        return nil;

                return y;
        }
```

02

15일차 내용 복습 (rb tree)



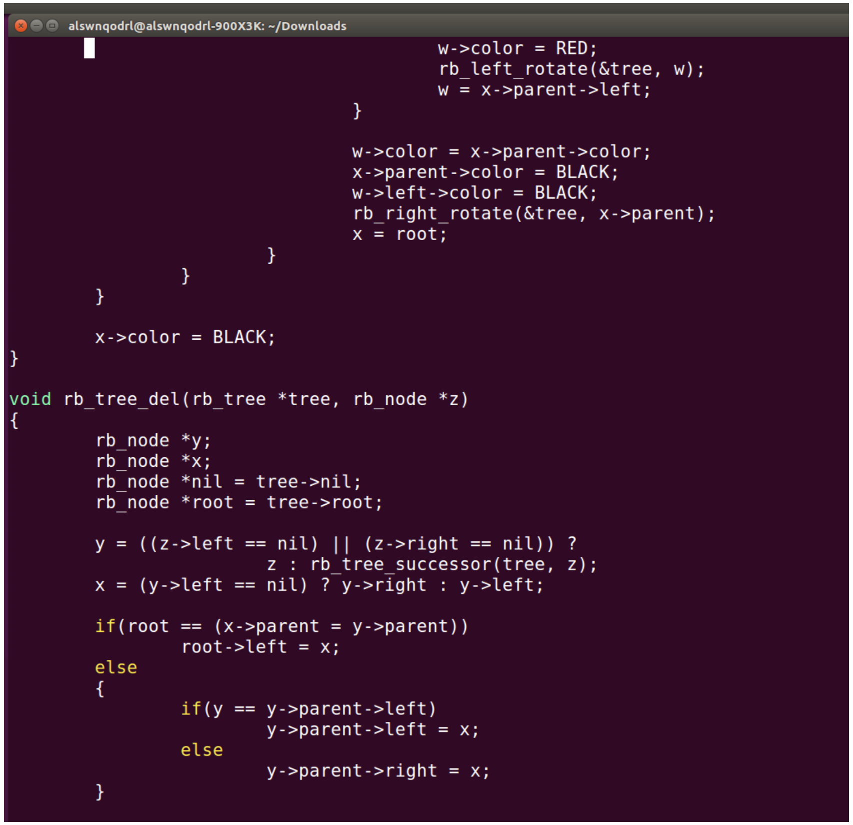
```
}

void rb_tree_del_fixup(rb_tree *tree, rb_node *x)
{
        rb_node *root = tree->root->left;
        rb_node *w;

        while((!x->color) && (root != x))
        {
                if(x->parent->left == x)
                {
                        w = x->parent->right;

                        if(w->color)
                        {
                                w->color = BLACK;
                                x->parent->color = RED;
                                rb_left_rotate(&tree, x->parent);
                                w = x->parent->right;
                        }

                        if((!w->right->color) && (!w->left->color))
                        {
                                w->color = RED;
                                x = x->parent;
                        }
                        else
                        {
                                if(!w->right->color)
                                {
                                        w->left->color = BLACK;
                                        w->color = RED;
                                        rb_right_rotate(&tree, w);
                                        w = x->parent->right;
                                }

                                w->color = x->parent->color;
                                x->parent->color = BLACK;
                                w->right->color = BLACK;
                                rb_right_rotate(&tree, x->parent);
                                x = root;
                        }
                }
```



```
                else
                {
                        w = x->parent->left;

                        if(w->color)
                        {
                                w->color = BLACK;
                                x->parent->color = 1;
                                rb_right_rotate(&tree, x->parent);
                                w = x->parent->left;
                        }

                        if((!w->right->color) && (!w->left->color))
                        {
                                w->color = RED;
                                x = x->parent;
                        }
                        else
                        {
                                if((!w->right->color) && (!w->left->color))
                                {
                                        w->right->color = BLACK;
                                        w->color = RED;
                                        rb_left_rotate(&tree, w);
                                        w = x->parent->left;
                                                                  316,1
```

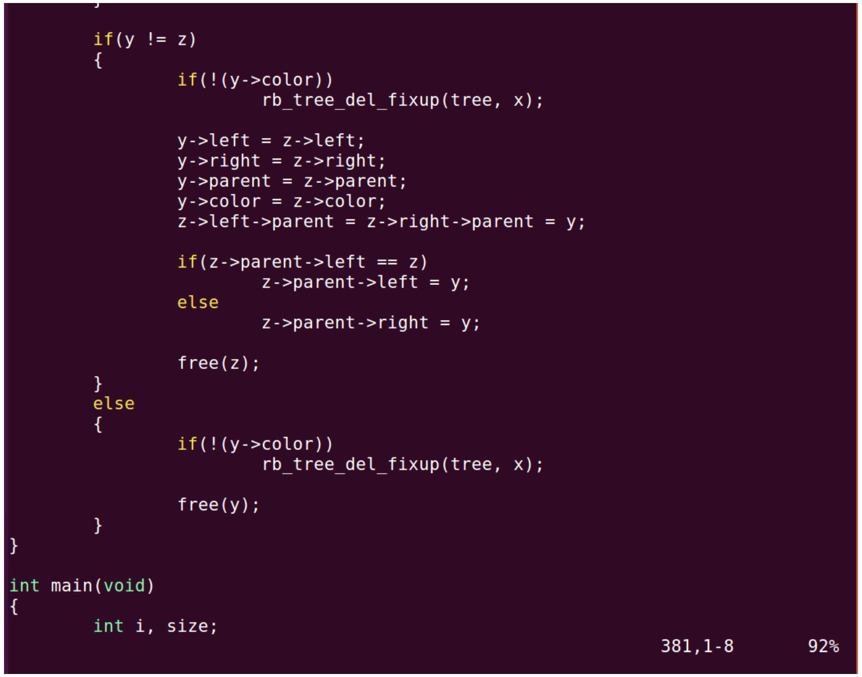
02

15일차 내용 복습 (rb tree)



alswnqodrl@alswnqodrl-900X3K: ~/Downloads

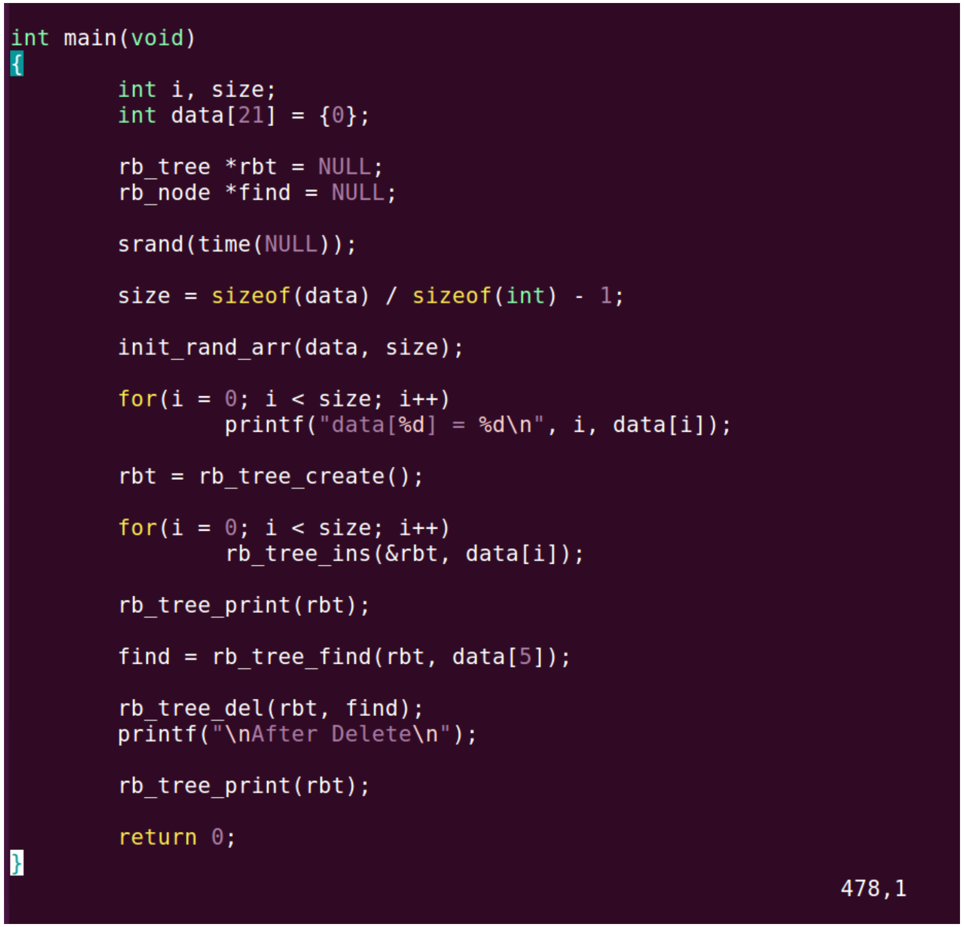
```
                                        w->color = RED;
                                        rb_left_rotate(&tree, w);
                                        w = x->parent->left;
                                }

                                w->color = x->parent->color;
                                x->parent->color = BLACK;
                                w->left->color = BLACK;
                                rb_right_rotate(&tree, x->parent);
                                x = root;
                        }
                }
        }

        x->color = BLACK;
}

void rb_tree_del(rb_tree *tree, rb_node *z)
{
        rb_node *y;
        rb_node *x;
        rb_node *nil = tree->nil;
        rb_node *root = tree->root;

        y = ((z->left == nil) || (z->right == nil)) ?
                        z : rb_tree_successor(tree, z);
        x = (y->left == nil) ? y->right : y->left;

        if(root == (x->parent = y->parent))
                root->left = x;
        else
        {
                if(y == y->parent->left)
                        y->parent->left = x;
                else
                        y->parent->right = x;
        }
```



```
        if(y != z)
        {
                if(!(y->color))
                        rb_tree_del_fixup(tree, x);

                y->left = z->left;
                y->right = z->right;
                y->parent = z->parent;
                y->color = z->color;
                z->left->parent = z->right->parent = y;

                if(z->parent->left == z)
                        z->parent->left = y;
                else
                        z->parent->right = y;

                free(z);
        }
        else
        {
                if(!(y->color))
                        rb_tree_del_fixup(tree, x);

                free(y);
        }
}

int main(void)
{
        int i, size;
                                                              381,1-8       92%
```

02

15일차 내용 복습 (rb tree)



```
int main(void)
{
        int i, size;
        int data[21] = {0};

        rb_tree *rbt = NULL;
        rb_node *find = NULL;

        srand(time(NULL));

        size = sizeof(data) / sizeof(int) - 1;

        init_rand_arr(data, size);

        for(i = 0; i < size; i++)
                printf("data[%d] = %d\n", i, data[i]);

        rbt = rb_tree_create();

        for(i = 0; i < size; i++)
                rb_tree_ins(&rbt, data[i]);

        rb_tree_print(rbt);

        find = rb_tree_find(rbt, data[5]);

        rb_tree_del(rbt, find);
        printf("\nAfter Delete\n");

        rb_tree_print(rbt);

        return 0;
}
                                                                478,1
```