Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정 #10

강사 : Innova Lee(이 상훈)

학생 : 김 시윤

1. 배운내용 복습

1) delete tree (소스코드)

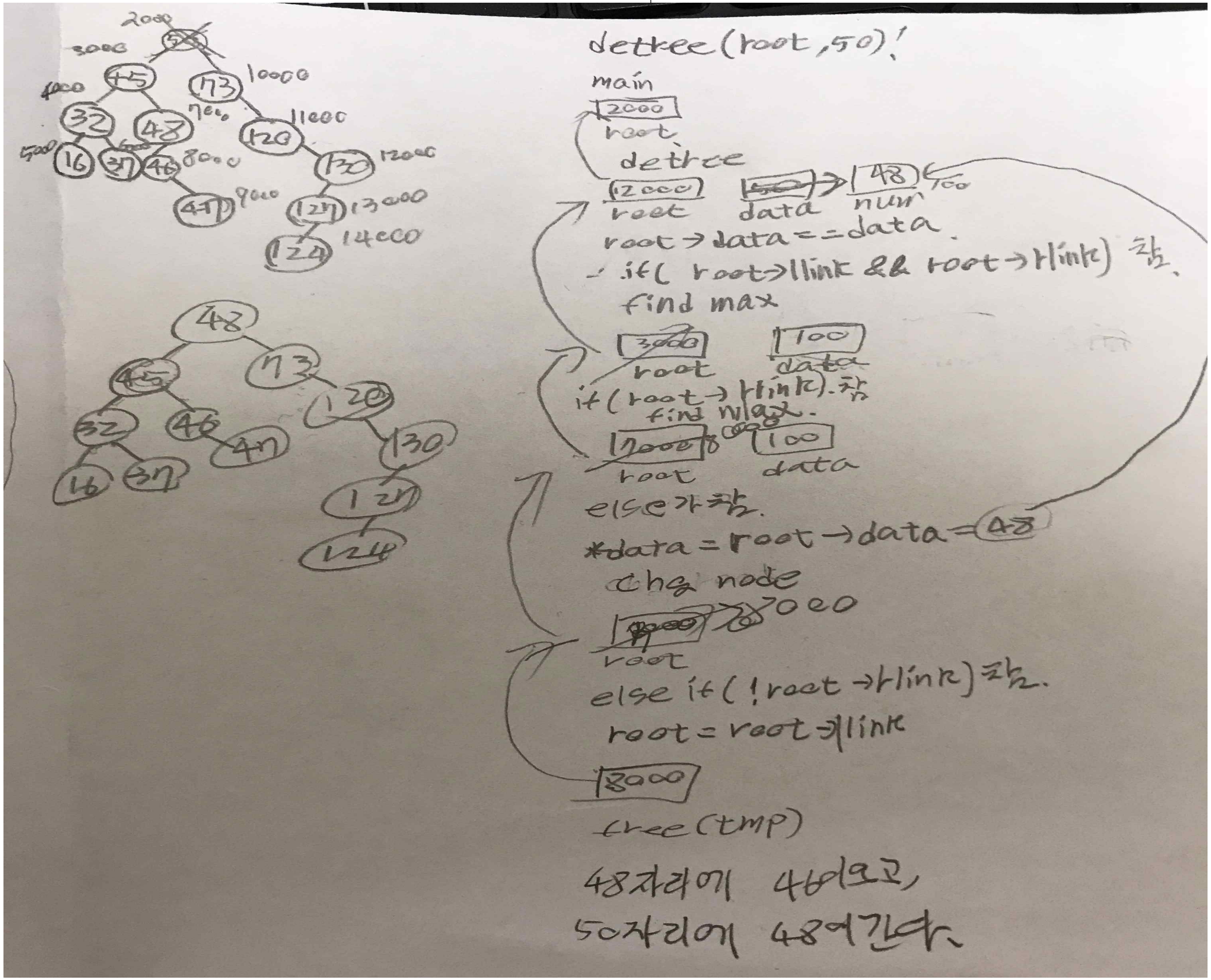
```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>

#define EMPTY 0

struct node{
        int data;
        struct node *rlink;
        struct node *llink;
};
typedef struct node tree;

tree *get_node()
{
        tree *tmp;
        tmp = (tree *)malloc(sizeof(tree));

        tmp->llink=EMPTY;
        tmp->rlink=EMPTY;
        return tmp;
}
void insert(tree **root,int data)
{
        if(*root ==NULL)
        {
                *root = get_node();
                (*root)->data = data;
                return;
        }
        if((*root)->data > data)
        {
                insert(&(*root)->llink,data);
        }
        else
        {
                insert(&(*root)->rlink,data);}
void print_tree(tree *root)
```

```

tree *find_max(tree *root,int *data)
{
        if(root->rlink)
        {
        root->rlink = find_max(root->rlink,data);
        }
        else
        {
        *data = root->data;
        root = chg_node(root);
        }
        return root;
}
tree *detree(tree *root,int data)
{
        int num;
        if(root == NULL)
        {
                printf("There are no data that you delete₩n");
                return NULL;
        }
        else if(root->data > data)
        {
                root->llink =  detree(root->llink,data);
        }
        else if(root->data<data)
        {
         root->rlink = detree(root->rlink,data);
        }
        else if(root->llink && root->rlink)
        {
                root->llink = find_max(root->llink,&num);
                root->data = num;
        }
```

```
{
        if(root)
        {
                printf("%d , ",root->data);
                print_tree(root->llink);
                print_tree(root->rlink);
        }
}

}
tree *chg_node(tree *root)
{
        tree *tmp=root;
        if(!root->llink)
        {
        root = root->rlink;

        }else if(!root->rlink)
        {
        root= root->llink;
        }

        free(tmp);
        return root;
}
```

```
        else
        {
                root=chg_node(root);
        }
        return root;
}
int main(void)
{       tree *root=EMPTY;
        int i=0;
        int a[20]={50,45,73,32,48,46,16,37,120,47,130,127,124};
case1           for(i=0;i<a[i];i++)
        {
                insert(&root,a[i]);
        }
                detree(root,50);

        print_tree(root);
        printf("₩n");

cas2            for(i=0;i<a[i];i++)
        {
                insert(&root,a[i]);
        }
                detree(root,45);

        print_tree(root);
        printf("₩n");

cas3            for(i=0;i<a[i];i++)
        {
                insert(&root,a[i]);
        }
                detree(root,127);

        print_tree(root);
        printf("₩n");
     return 0;
}
```

delete tree 50

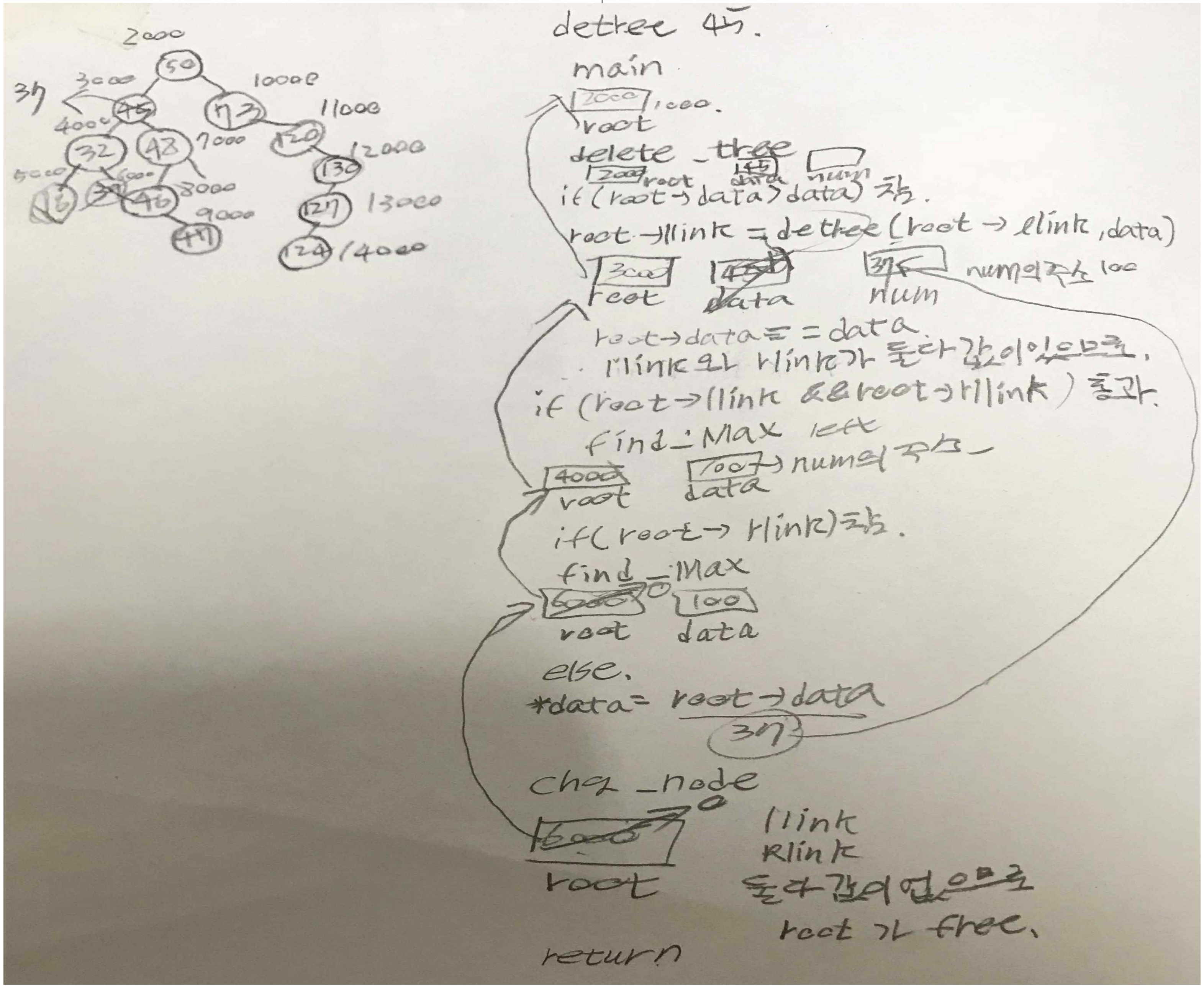

2000
~~50~~
3000
45
73
10000
4000
32
48
7000
120
11000
5000
16
37
6000
46
8000
130
12000
47
9000
127
13000
124
14000
48
45
73
120
32
46
47
130
16
37
127
124

```
detree(root,50)!
main
2000
root
 detree
12000        ~~50~~ → 48  ← 100
root         data      num
root→data==data.
 if( root→llink && root→rlink) 참.
  find max
  ~~3000~~          100
  root          data
if(root→rlink).참
   find max.
~~12000~~ ~~8000~~   100
  root          data
else가 참.
*data=root→data=48
 chg node
~~8000~~ → ~~8000~~ 7000
  root
else if(!root→rlink)참.
 root=root→llink
8000
free(tmp)
```

48자리에 46이오고,

50자리에 48이간다.

delete tree 45


2000
50
3000
10000
37
45
73
11000
4000
32
48
7000
120
12000
5000
16
37
6000
46
8000
130
9000
127
13000
47
124
14000

detree 45.

main

2000 1000.
root

delete _tree
2000 root 45 data num

if(root→data>data) 참.

root→llink = detree(root→llink,data)

3000 root 45 data 37 num num의주소 100

root→data = = data

rlink와 rlink가 둘다 값이있으므로,

if (root→llink && root→rlink) 통과.

find_Max left

4000 root 100 data → num의 주소

if(root→rlink)참.

find_Max

~~6000~~ root 100 data

else.

*data= root→data

37

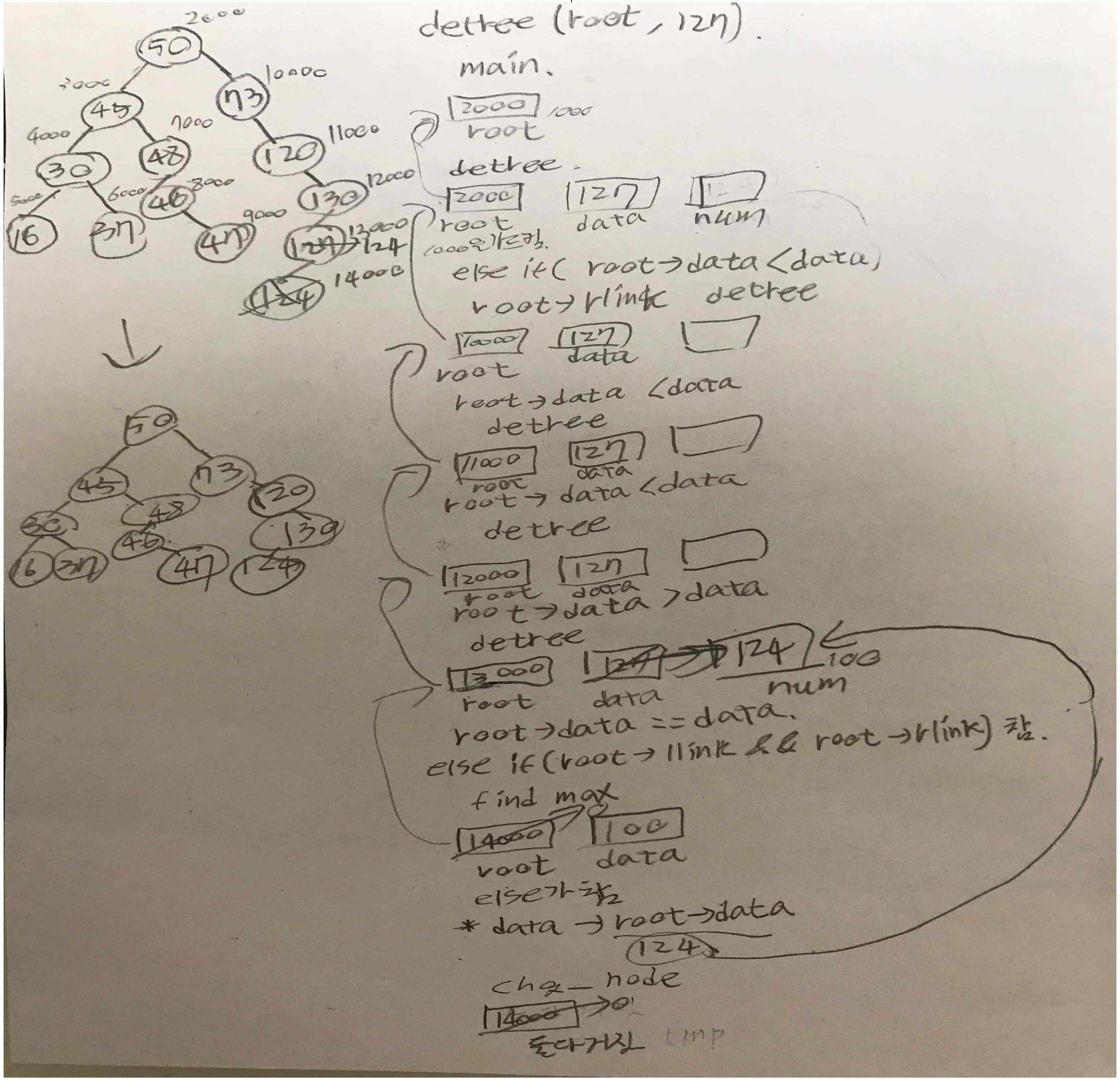
chg_node

~~6000~~
root

llink
Rlink
둘다 값이 없으므로
root 가 free.

return

delete tree 127



detree (root, 127).

main.

2000 1000

root

detree.

2000 127 12

root data num

1000을가르킴.

else if(root→data<data)

root→rlink detree

10000 127

root data

root→data <data

detree

11000 127

root data

root→data <data

detree

12000 127

root data

root→data >data

detree

13000 ~~127~~ → 124 100

root data num

root→data == data.

else if(root→llink && root→rlink) 참.

find max

14000 100

root data

else가 참

* data → root→data

124

chg_node

14000 → 0

돌아가짐 tmp