

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

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 장성환

redmk1025@gmail.com

* 이진트리 재귀 풀기
<MY CODE>

<LEC CODE>

```
typedef struct _stack{
    tree* data;
    struct _stack *link;
}stack;

stack *get_stack(){
    stack *tmp = (stack*)malloc(sizeof(stack));
    tmp->link = NULL;
    return tmp;
}

tree* pop(stack **top){

    tree * root = (*top)->data;
    *top=(*top)->link;

    return root;
}

stack* push(stack *top, tree *root){
    stack *tmp = get_stack();
    tmp->data = root;
    tmp->link = top;
    top=tmp;
    return top;
}
```

```
typedef struct __stack
{
    void *data;
    //데이터 형을 void 로 받아서 어떤 데이터를 받아도 될 수 있도록 조정

    struct __stack *link;
} stack;

stack *get_stack_node(void)
{
    stack *tmp;
    tmp = (stack *)malloc(sizeof(stack));
    tmp->link = NULL;
    return tmp;
}

void *pop(stack **top)
{
    stack *tmp = *top;
    void *data = NULL;

    if(*top == NULL)
    {
        printf("stack is empty!\n");
        return NULL;
    }// 스택에 아무것도 들어 있지 않으면 NULL 반환

    data = (*top)->data;
    *top = (*top)->link;
```

```

void printf_tree(tree **root){
    stack * top=NULL;
    tree * tmp =NULL;

    while((*root)!=NULL){
        while((*root)!=NULL){
            if((*root)->right){
                top=push(top, *root);
            }
            printf("%d\n",(*root)->data);
            root = &(*root)->left;
        }
        if(top == NULL)
            break;
        *root = pop(&top)->right;
    }
}

```

```

    free(tmp) //동적 메모리 해제.
    //return (*top)->data;
    return data;
}

```

```

void push(stack **top, void *data)
{

```

```

    if(data == NULL) //데이터가 없으면 push 기능이 동작하지 않음!
        return;

```

```

    stack *tmp = *top;
    *top = get_stack_node();
    (*top)->data = malloc(sizeof(void *));
    (*top)->data = data;
    (*top)->link = tmp;
}

```

```

void print_tree(tree **root)

```

```

{ //후입선출의 개념이 도입되어 완벽히 트리를 순회한다. 오른쪽을 먼저 받고 왼쪽을
  받아서 나오는 값이 왼쪽이 우선순위가 된다.

```

```

    tree **tmp = root;
    stack *top = NULL;
    push(&top, *tmp);
    while(stack_is_not_empty(top))
    {
        tree *t = (tree *)pop(&top);
        tmp = &t;
        printf("data = %d, ", (*tmp)->data);
        push(&top, (*tmp)->right);
        push(&top, (*tmp)->left);
    }
}

```

```

}

```