

# TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)

[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

학생 - 하성용

[accept0108@naver.com](mailto:accept0108@naver.com)

tree\_ins

[illegible]

```

void find_max(tree **root, int *data)
{
    tree *tmp = root;
    while(*tmp)
    {
        if ((*tmp) -> right)
            tmp = (*tmp) -> right;
        else
            *data = (*tmp) -> data;
            *tmp = (*tmp) -> left;
    }
    break;
}

tree * chg_node(tree *root)
{
    tree *tmp = root;
    if (!root -> right)
        root = root -> left;
    else if (!root -> left)
        root = root -> right;
    free(tmp);
    return root;
}

```

## delete

```
void non_recur_delete_tree(tree **root, int data)
{
    tree **tmp = root;
    int num;
    while (*tmp)
    {
        if ((*tmp) -> data > data) tmp가 가리키는 data의 값이 data보다 크다면
            tmp = &(*tmp) -> left; tmp가 가리키는 left의 주소값을 tmp에 넣는다.
        else if ((*tmp) -> data < data) tmp가 가리키는 data의 값이 data보다 작다면
            tmp = &(*tmp) -> right; tmp가 가리키는 right의 주소값을 tmp에 넣는다.
        else if ((*tmp) -> left && (*tmp) -> right) tmp가 가리키는 left
            tmp가 가리키는 right
            {
                find_max(&(*tmp) -> left, &num);
                (*tmp) -> data = num; tmp가 가리키는 data의 값을 num에 넣는다.
                return;
            }
        else
        {
            (*tmp) = chg_node(*tmp);
            return;
        }
    }
    printf("Not Found\n");
}
```

## 사전평가문제

사전평가		현재문항 : 2 / 92
문제	Stack 및 Queue 외에 Tree 라는 자료구조가 있다.	
내용	이 중에서 Tree 는 Stack 이나 Queue 와는 다르게 어떠한 이점이 있는가 ?	

: Tree는 검색속도가 빠르다

사전평가		현재문항 : 7 / 92
문제	이것이 없으면 C 언어의 함수를 호출할 수 없다.	
내용	여기서 이야기하는 이것은 무엇일까 ?	

: 포인터

사전평가		현재문항 : 9 / 92
문제	다음 사항에 대해 기술하시오.	
내용	Memory Hierarchy(메모리 계층 구조)에 대해 기술하시오.	

: 레지스터, 캐시, 메모리, 하드디스크

사전평가		현재문항 : 11 / 92
문제	다음 사항에 대해 답하십시오.	
내용	파이프라인이 깨지는 경우에 대해 기술하십시오.	

: 분기명령어 파이프라인에 치명적인 손실점

사전평가		현재문항 : 13 / 92
문제	다음 질문에 대해 답하십시오.	
내용	goto 를 사용하는 이유에 대해 기술하십시오.	

: goto는 jmp 하나로 가볍다

사전평가		현재문항 : 19 / 92
문제	아래 자료 구조를 C 언어로 구현해보시오.	
내용	Stack 자료구조를 아래와 같은 포맷에 맞춰 구현해보시오. (힌트: 이중 포인터) <pre>ex)  int main(void) {     stack *top = NULL;     push(&amp;top, 1);     push(&amp;top, 2);     printf("data = %d\n", pop(&amp;top)); }</pre>	

```
#include<stdio.h>
#include<malloc.h>
#define EMPTY 0

typedef struct __stack{
int data;
struct __stack *link;
}stack;

stack *get_node();
{
stack *tmp;
tmp=(stack *)malloc(sizeof(stack));
tmp->link=data;
return tmp;
}

void push(stack **top, int data)
{
stack *tmp;
tmp=*top;
*top=get_node();
(*top)->data=data;
(*top)->link=tmp;
}

int pop(stack **top)
{
stack *tmp;
int num;
tmp=*top;
if(*top==EMPTY)
{
printf("Stack is empty!!\n");
return 0;
}
num=tmp->data;
*top=(*top)->link;
free(tmp);
return num;
}

int main(void)
{
stack *top = NULL;
push(&top,1);
push(&top,2);
printf("data=%d\n",pop(&top));

return 0;
}
```

사전평가		현재문항 : 20 / 92
문제	다음 질문에 답하십시오.	
내용	Binary Tree 나 AVL Tree, Red-Black Tree 와 같이 Tree 계열의 자료구조를 재귀 호출 없이 구현하고자 한다. 이 경우 반드시 필요한 것은 무엇인가 ?	

: 함수

사전평가		현재문항 : 22 / 92
문제	다음 질문에 대해 기술하십시오.	
내용	AVL 트리는 검색 속도가 빠르기로 유명하다. Red-Black 트리는 검색 속도가 빠르지만 AVL 트리보다 느리다. 그런데 어째서 SNS 솔루션등에서는 AVL 트리가 아닌 Red-Black 트리를 사용할까 ?	

: Red-Black 트리는 소규모, 대규모 정보처리에 용이

사전평가		현재문항 : 85 / 92
문제	기본 for 문 활용 문제다.	
내용	1 ~ 100 까지의 숫자중 홀수만 더해서 출력해보시오.	

```
#include<stdio.h>

int main(void)
{
    int i;
    int sum;

    for(i=0;i<=100;i++)
    {
        if(i%2==1)
        {
            sum+=i;

            printf("홀수의 합 : %d\n",&sum);

            return 0;
        }
    }
}
```

사전평가		현재문항 : 91 / 92
문제	차량과 토크의 관계	
내용	실제로 차량을 개발하고자할 때 토크 선정이 매우 중요한데 그 이유가 무엇인지 상세히 기술하십시오.	

: 토크는 타이어의 회전하는힘

빠르게 이동하거나 자동차가 경사로를 가는등 힘이 필요할 때 중요한 역할을함