

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - 하성용

accept0108@naver.com

8일차

자료구조 - 스택 push & pop

```

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

struct node {
    int data;
    struct node *link;
};

typedef struct node Stack;

Stack *get_node()
{
    Stack *tmp;
    tmp = (Stack *) malloc(sizeof(Stack));
    tmp->link = EMPTY;
    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 = EMPTY;
    push(&top, 10);
    push(&top, 20);
    push(&top, 30);
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    printf("%d\n", pop(&top));
    return 0;
}

```

- PUSH & POP

The diagram illustrates the push and pop operations on a stack. The stack grows downwards from higher memory addresses (4000) to lower ones (1000). The heap grows upwards. The diagram shows the state of the stack and heap after each operation:

- Initial State:** The stack is empty. The top pointer is at 4000. The heap starts at 2000.
- Push 10:** A new node with data 10 is created at 2000. The top pointer moves to 3000.
- Push 20:** A new node with data 20 is created at 2000. The top pointer moves to 3000.
- Push 30:** A new node with data 30 is created at 2000. The top pointer moves to 3000.
- Pop 30:** The node with data 30 is removed. The top pointer moves to 3000.
- Pop 20:** The node with data 20 is removed. The top pointer moves to 3000.
- Pop 10:** The node with data 10 is removed. The top pointer moves to 3000.

stack is empty!!! 출력