

1111

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

강사 - 이상훈

학생 - 윤연성
whatmatters@naver.com

dddd

```
//포인터의 초기화 상태 *p = &a or *p = NULL
    *p = 100;은 안됨
```

```
// 포인터도 사칙연산 가능    p의 값이 1000을 가
리키면 p++은 형에따라 int는 1004
```

```
//    *p++ = 현재가리키는 p값을 가져온 후 p
를 증가시킨다 = p를 증가시킴
    (*p)++ = p가 가리키는 대상의값을 증가시
킨다 = 값을 증가시킴
```

```
// 배열 자체가 포인터임    a[] = {3, 4, 5, 6,
7};    출력하면 a로 하면 a[]시작주소값이 나옴
```

```
// 값에 대한 호출 : 복사값
// 참조에 대한 호출 : 원본
```

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

```
struct node
```

dddd

```
{
    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 == 0)
    {
        printf("Stack is empty!\n");
        return 0;
    }
}
```

```

                                dddd
    }

    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;
}

```

void (* bbb(void))(void)

리턴: void (*)(void)

이름: bbb

인자: void

void (*)(void) bbb(void)

dddd

```
void ccc(void (*p)(void))
```

리턴: void

이름: ccc

인자: void (*p)(void)

```
int (* ddd(void))(void)
```

리턴: int (*)(void)

이름: ddd

인자: void

```
int (*)(void) ddd(void)
```

```
void (* bbb(void (*p)(void)))(void)
```

리턴: void (*)(void)

이름: bbb

인자: void (*)(void)

```
void (*)(void) bbb(void (*p)(void))
```