1111

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

강사 - 이상훈

학생 - 윤연성
whatmatters@naver.com

888888

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

// 포인터도 사칙연산 가능   p의 값이 1000을 가리키면 p++은 형에따라 int는 1004


//     *p++  =  현재가리키는 p값을 가져온 후 p를 증가시킨다 = p를 증가시킴
        (*p)++ = p가 가리키는 대상의값을 증가시킨다 = 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
{
    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;
    }

    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)
```

```
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))
```

```
#include <stdio.h>

void aaa(void)

{
    printf("aaa called₩n");
}

void (* bbb(void(*p)(void)))(void)                    //리턴 : void (*)
bbb(void)
{                                                     //이름 : bbb
    p();                                              //인자 void (*) bbb(void)
    printf("bbb called₩n");
    return aaa;
}

int main(void)
{
    bbb(aaa);
    return 0 ;
}
```

```
------typedef----------

#include <stdio.h>
```

888888

```
                    //typedef

typedef int INT;          // int를 INT로 바꿈
typedef int* PINT;          // int*를   PINT로 바꿈

int main(void)
{
        INT num = 3;
    PINT ptr = &num;

    printf("num = %d₩n", *ptr);
    return 0;

}
```

```
#include <stdio.h>
struct pos
{
    double x_pos;
    double y_pos;
};
int main(void)
{
    double num;
    struct pos position;        // struct pos까지 새로운 데이터타입
    num = 1.2;                      //구조체 = 커스텀 데이터타입
    position.x_pos = 3.3;           //구조체 안의 x_pos를 쓰겠다라는건  .을
써야됨
    position.y_pos = 7.7;

printf("sizeof(position) = %lu₩n", sizeof(position));
return 0;
}
```

```
#include <stdio.h>
#define NAME_LEN 30
#define ID_LEN 18

 typedef struct __id_card         // struct 반드시 한칸을 띄어야함 /   __언더
바 두개의 의미 커널개발의 관습적표현 건들이지말아라
{
    char name[NAME_LEN];
    char id[ID_LEN];
    unsigned int age;
} id_card;                              //여기까지가 구조체_
```

```
int main(void)
{
    int i;
    id_card arr[2] =
     {
        {"Marth Kim", "800903 - 1012589", 34},
        {"July Eun", "830708 - 1023417", 31}
    };
    for (i= 0; i < 2; i++)
    {
        printf("name = %s, id = %s, age = %d₩n",arr[i].name, arr[i].id,
arr[i].age);
    }
return 0;



#include <stdio.h>

void aaa(void)                                          //함수포인터 함수
의 주소를 저장( 함수도 주소에 저장할수 있다)
{                                                       //프로토타입 = 리
턴, 함수명, 인자에 대한 기술서
    printf("aaa called₩n");
}

void bbb(void(*p)(void))                                    //aaa의 주소
값을 받음(함수의)
{
    p();                    //
    printf("bbb called₩n");
}

int main(void)
{
bbb(aaa);
return 0;

}
*/

}
```