

# 포인터

ex)

선언 : 자료형 \* 변수명;  
(int \* pth;)

→ int i = 3  
int \* p = &i; i의 주소값.  
p: 주소값 (0x ~ ~)  
\*p: 내용 (3)

"포인터 변수는 메모리의 위치"

int a = 10;  
int \* pth = &a;

%p [ a의 주소값: (0x ~ ~) &a = pth, pth의 주소값 = \*pth  
%d [ a의 값: a = \*pth

'b

"포인터 연산"

int a = 10;  
int \* pth = &a;

✗ \*pth = a + 2; → a = 12.

# 포인터 Swap

```
void Swap(int a, int b)
{
    temp = b;
    temp = a;
    a = b;
    b = temp;
}
```

⇒ "a와 b가  
안바뀜"

↳ 주소의 내용을 바꿔보라.  
(포인터)

```
void Swap(int *a, int *b); 함수명
int main(void)
{
```

① ~~Swap(x, y)~~ 함수 호출  
Swap(&x, &y)

```
void Swap(int *a, int *b) 함수 정의.
{
```

```
    int = temp;
    temp = *a; *a
```

```
    *a = *b;
    *b = temp;
}
```

① 함수 포인터로 호출

→ 메모리 주소.

② 포인터 바뀐의 \*a, \*b로  
바뀌야함

→ 메모리 주소.

③ 주소를 받아와야 하니  
&x, &y로 변환

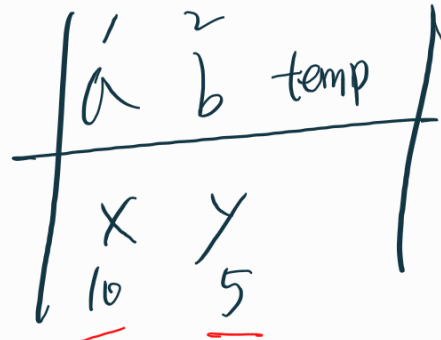
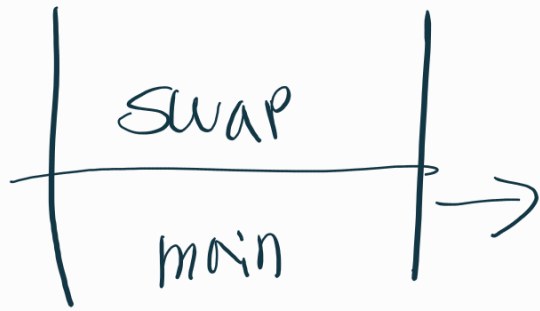
→ 성공.

a와 b가 안바뀜.

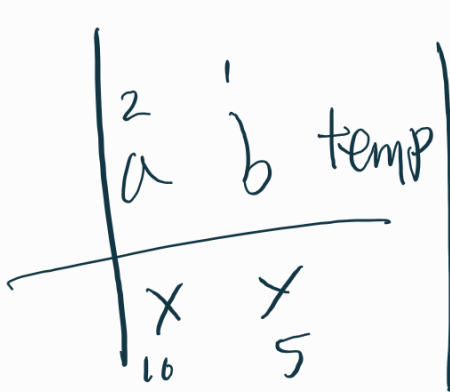
Why?



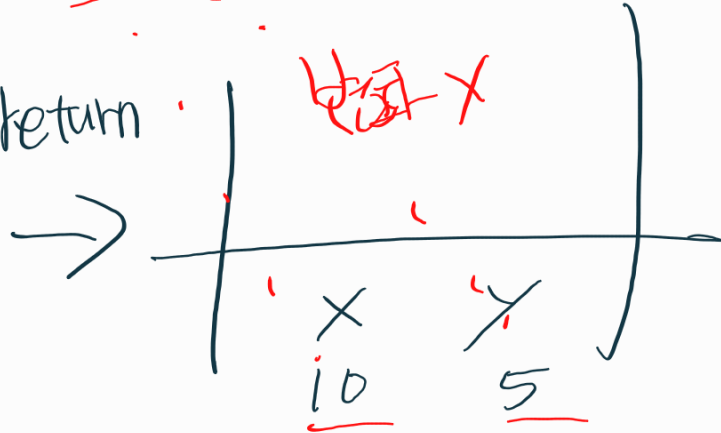
temp을 호출 변수로 사용



temp = a  
a = b  
b = temp

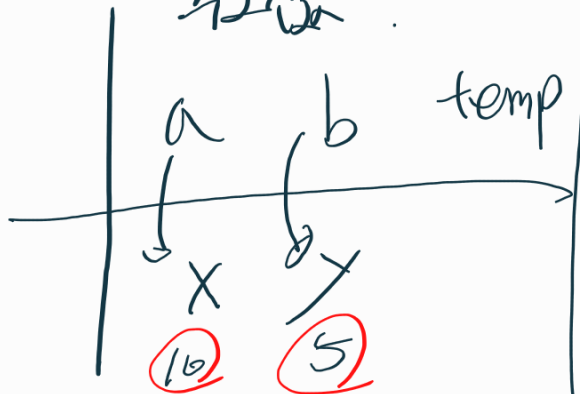
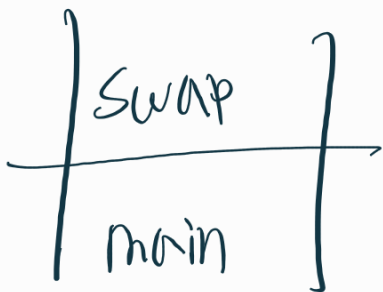


return

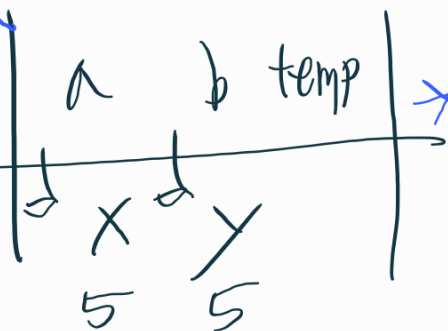


→ 포인터 사용 시 ("call by reference")

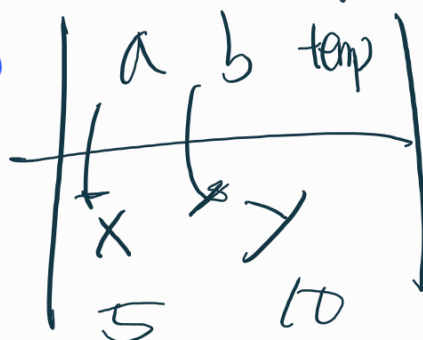
변수



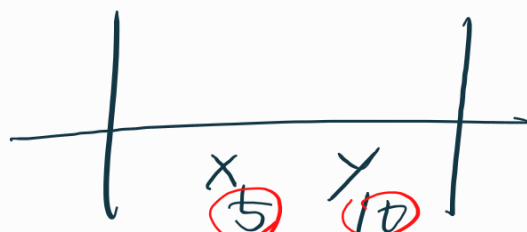
temp = \*a  
\*a = \*b



\*b = temp



return



# 포인터와 배열

## ■ 배열의 주소를 표현하는 방법

표 10-1 배열 a의 주소를 표현하는 방법

구분	첫 번째 주소	두 번째 주소	세 번째 주소	네 번째 주소
방법 1	<code>&amp;a[0]</code>	<code>&amp;a[1]</code>	<code>&amp;a[2]</code>	<code>&amp;a[3]</code>
방법 2	<code>a</code>	<code>a+1</code>	<code>a+2</code>	<code>a+3</code>
방법 3	<code>p</code>	<code>p+1</code>	<code>p+2</code>	<code>p+3</code>

## ■ 배열의 값을 표현하는 방법

표 10-2 배열 a의 값을 표현하는 방법

구분	첫 번째 값	두 번째 값	세 번째 값	네 번째 값
방법 1	<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>
방법 2	<code>*a</code>	<code>*(a+1)</code>	<code>*(a+2)</code>	<code>*(a+3)</code>
방법 3	<code>*p</code>	<code>*(p+1)</code>	<code>*(p+2)</code>	<code>*(p+3)</code>
방법 4	<code>p[0]</code>	<code>p[1]</code>	<code>p[2]</code>	<code>p[3]</code>

주소:  $\&a[0] = a = p$

값:  $a[0] = *a = *p = p[0]$

"배열 불러오기"

배열전체

VS

포인터 변수만

함수 선언: `int Array(int Array[], int size) | int Array(int *pA, int size)`

함수 호출: `X = Array(a, 배열의 개수)`

함수 정의: `Array[] | pA[]`

