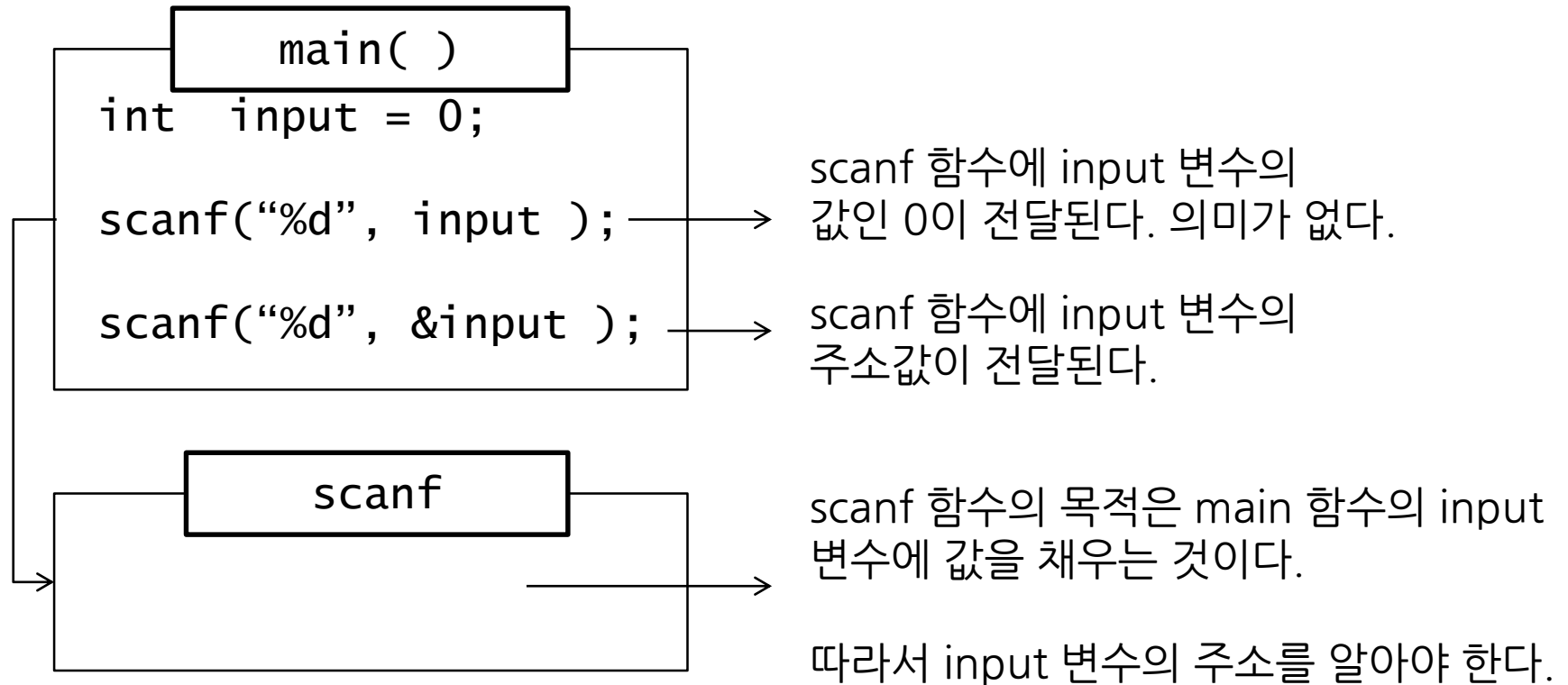


5차시 - 포인터(5)

scanf 함수에는 왜 &을 붙이나



◆ caller의 변수를 callee가 채울 수 있는 방법은 포인터 뿐이다.

scanf("%s", str);에는 왜 &를 안 붙였나

◆ 문자열 입력 받는 %s의 경우

```
char str[10];  
scanf("%s", str );    // scanf("%s", &str[0]) 과 동일
```

◆ str은 배열로 선언되어 있으며, 배열 변수의 이름만 쓸 경우 포인터 상수

◆ 따라서 &가 없어도 이미 포인터가 전달되기 때문

```
char input[3][10];  
scanf("%s", input[0] ); // 이차원 배열의 1차원 첨자도 포인터
```

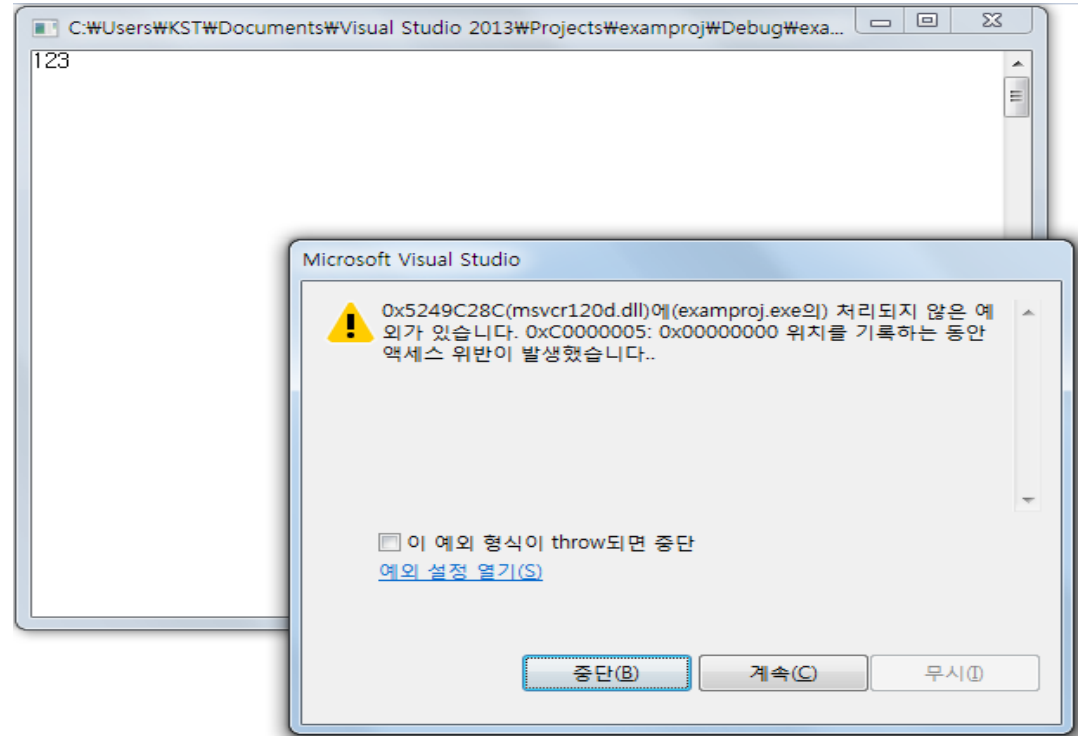
scanf 에 &을 안 붙이면 어떻게 될까

①

```
int a = 0;  
scanf("%d", a );
```

②

```
int a ;  
scanf("%d", a );
```



◆ 허가되지 않은 메모리 영역 접근

- 프로그램 비정상 종료

scanf_s는 왜 필요했나

①

```
char a;  
scanf("%d", &a);
```

◆ scanf 함수의 입장을 생각해 보자.

- 전달되는 포인터가 int형인지, char 형인지, 할당된 메모리가 얼마인지 모른다.

②

```
char str[10];  
scanf("%s", str );
```

◆ 30글자를 입력하면 어떻게 될까?

◆ 입력 크기를 지정하여 범위가 초과되는 것을 방지

```
scanf_s("%c", &c, 1);          // 크기를 지정한다.  
scanf_s("%s", s, 10);  
scanf_s("%s %s", s, 10, t, 10);
```

- %c와 %s에서만 크기를 지정한다. 지정하지 않으면 이상한 결과가 나온다.
- %d, %f에서는 크기를 지정할 필요 없다. (이미 알고 있으므로)

◆ 유사한 문제의 해결

- strcpy_s, strcat_s (문자열)

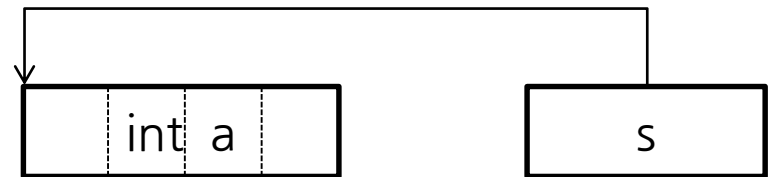
scanf_s는 표준이 아니며, Microsoft가 제안한 해결법이다.

int 형은 어떻게 메모리에 저장될까?

- ◆ `int a = 0x12345678; // int형은 4바이트다.`
`// 메모리에 어떻게 저장되는지 보려면?`

```
#include <stdio.h>
int main( ) {
    int a = 0x12345678;
    char *s = (char *) &a; // 경고를 방지하기 위한 형 변환
    int i;

    for (i = 0; i < 4; i++) {
        printf("%X ", *s);
        s++;
    }
}
```



int 형은 어떻게 메모리에 저장될까?

◆ 출력 결과

◆ 78 56 34 12

```
int a = 0x12345678;
```

78	56	34	12
----	----	----	----

◆ 이 결과는 어떤 의미일까

- int 형은 4바이트이며, 하위바이트 먼저 저장된다.

◆ 코드는 어떤 의미가 있을까.

- 왜 char * 형으로 s 변수를 선언했을까

포인터에 관련된 몇 가지 정보들

◆ 레지스터 변수의 주소는 어디일까?

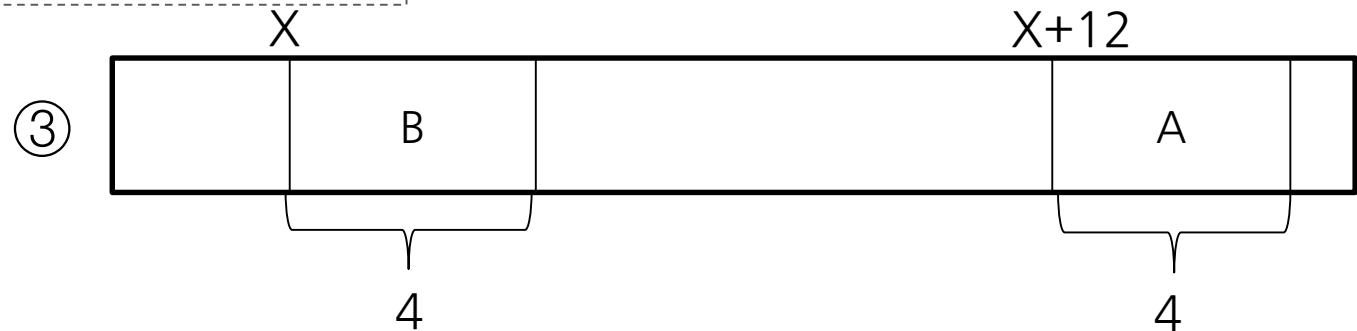
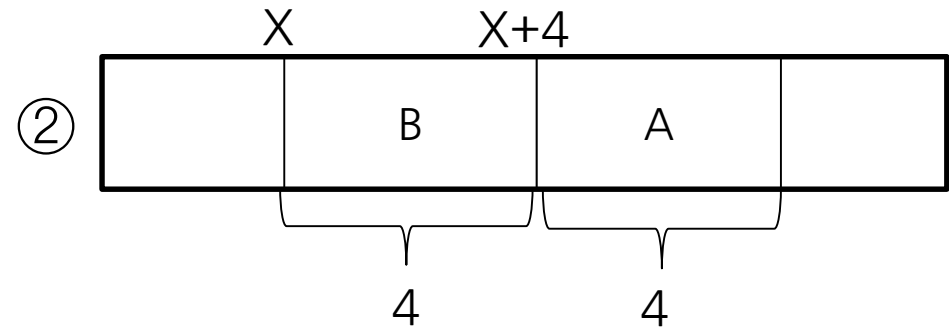
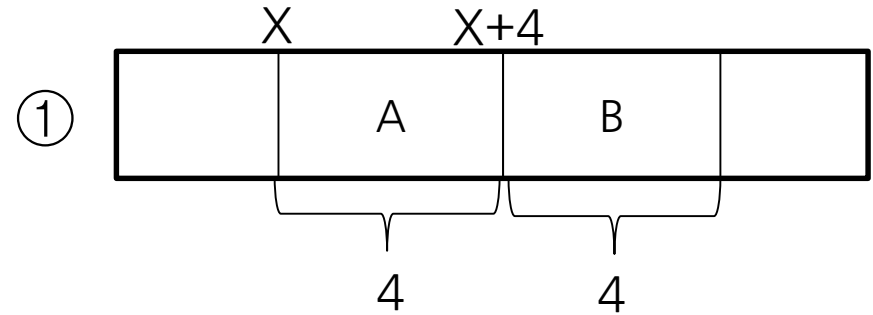
```
register int i;  
int *b;
```

```
b = &i;    // compile error : 레지스터 변수에 &가 있습니다.
```

포인터에 관련된 몇 가지 정보들

◆ 변수는 어떻게 생성될까?

```
int main( )  
{  
    int    A, B;  
    printf("&A : %p\n", &A);  
    printf("&B : %p\n", &B);  
    printf("diff %d\n", (char *)&A  
        - (char *)&B);  
}
```



컴파일러마다 다르다

포인터에 관련된 몇 가지 정보들

◆ NULL 은 주소 0 이다.

- #define NULL ((void *) 0)
- NULL은 의미 있는 주소 (0으로 초기화된 주소. 가리키지 않는 포인터)

◆ 잘못된 NULL의 사용

- NULL은 0 이 아니다.
- 예러는 아니지만, 잘못된 사용이다.(경고가 나온다)

```
int    i = 0 ; // 무엇이 잘못됐을까?  
i = NULL;  
if ( i == NULL)  
    printf("NULL\n");
```

포인터에 관련된 몇 가지 정보들

◆ 배열을 넘나드는 포인터

```
int main( )    // visual studio 에서만 동작하며,  
               // 다른 컴파일러의 결과는 다르다.  
{  
    int  A[5] = { 0 }, B[4] = { 0 };  
    int  *ptr = &B[6];  
    *ptr = 1;  
    printf("%d\n", A[0] );  
}
```

◆ 1이 출력되는 이유는 무엇일까.

– 엉뚱한 변수에 값이 들어가는 이유

