

C언어 - Pointer

1. 포인터 개념 정리
2. 포인터 keyword
3. 포인터 기본 문법
4. 함수 호출의 종류

1. 포인터 개념 정리

- 포인터는 메모리 주소 값을 저장하고 있는 변수
- 저장하고 있는 주소 값을 이용하여 해당 주소 값 근처나 해당 주소에 저장되어 있는 값을 읽고 쓰기가 가능함

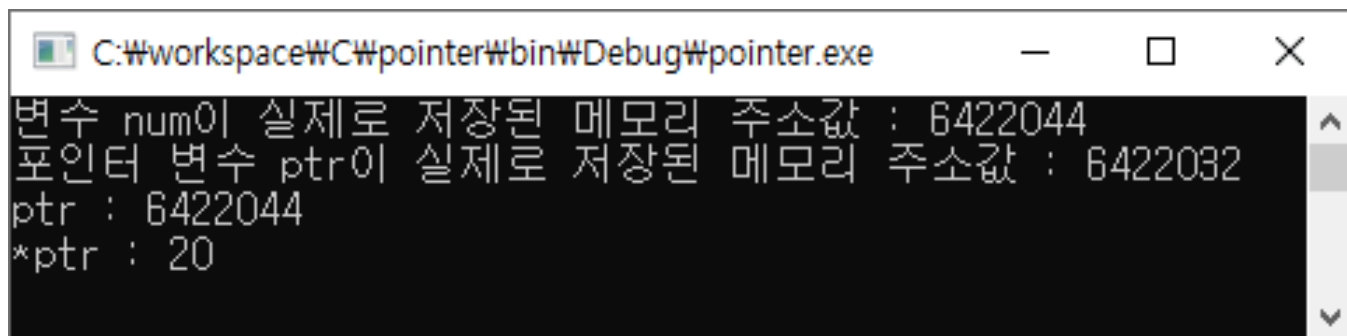
2. 포인터 Keyword

포인터는 메모리 주소 값만 저장하는 변수

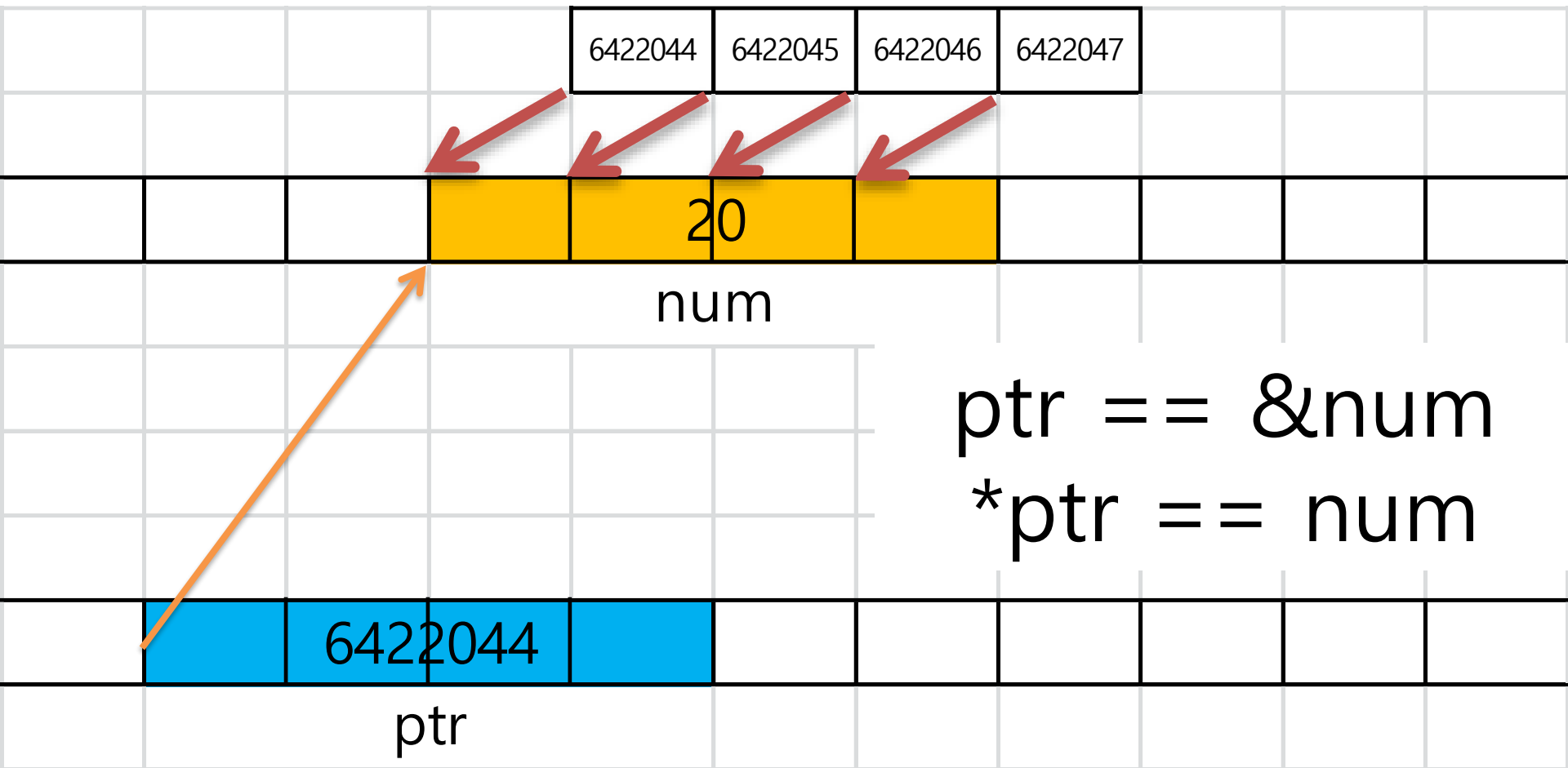
3. 포인터 기본 문법

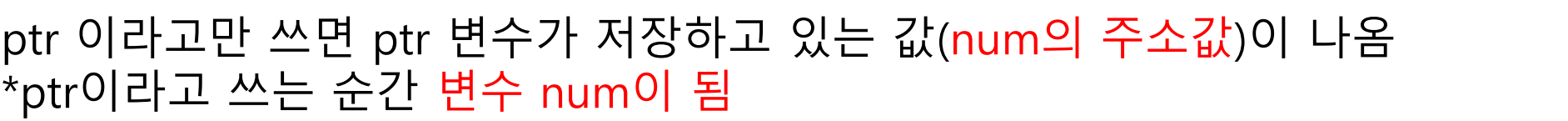
```
#include <stdio.h>

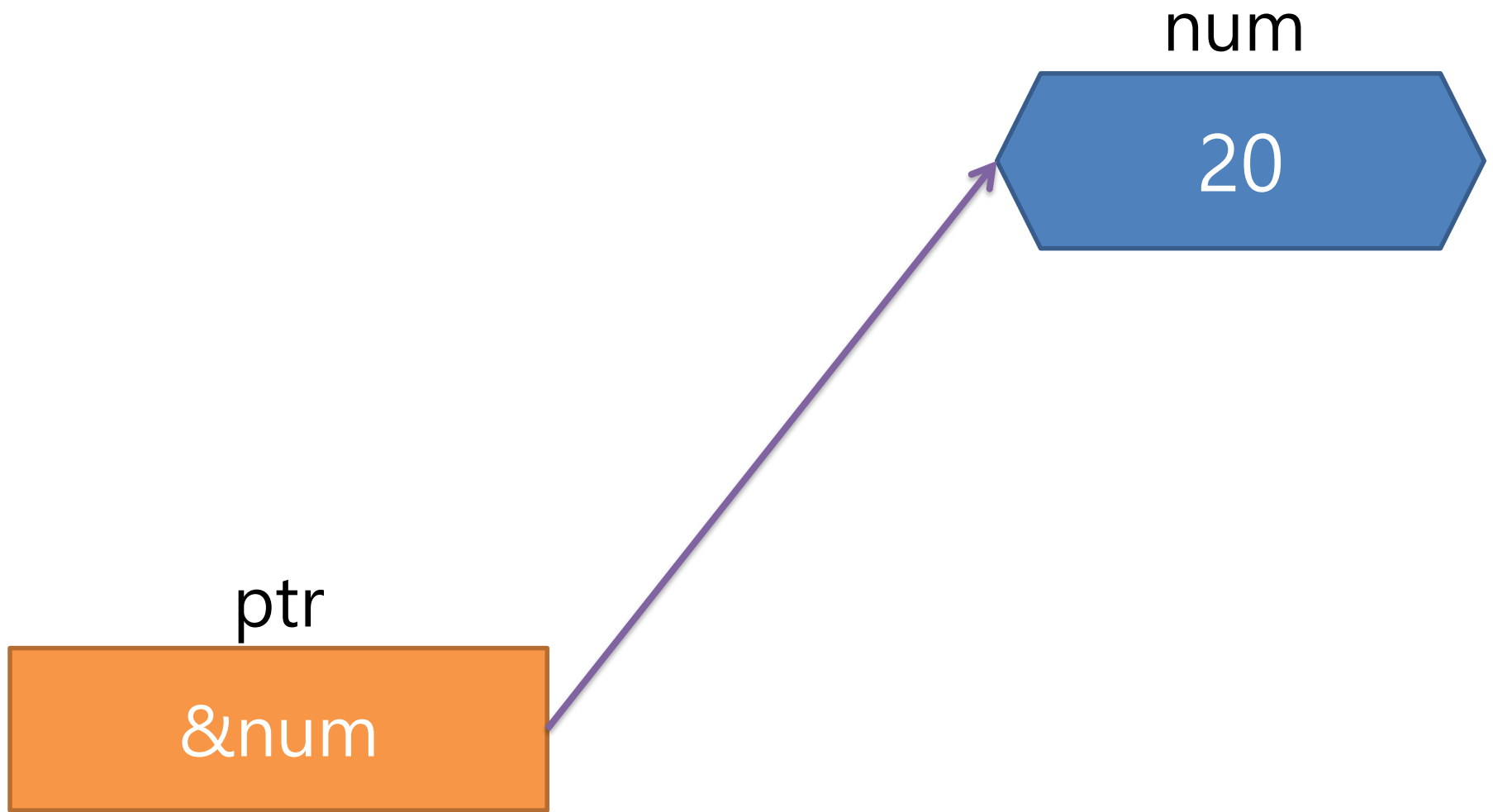
int main()
{
    int num = 20;
    int *ptr = &num;
    printf("변수 num이 실제로 저장된 메모리 주소값 : %d\n",&num);
    printf("포인터 변수 ptr이 실제로 저장된 메모리 주소값 : %d\n",&ptr);
    printf("ptr : %d\n",ptr);
    printf("*ptr : %d\n",*ptr);
    return 0;
}
```



```
C:\workspace\WC\pointer\bin\Debug\pointer.exe
변수 num이 실제로 저장된 메모리 주소값 : 6422044
포인터 변수 ptr이 실제로 저장된 메모리 주소값 : 6422032
ptr : 6422044
*ptr : 20
```







아래의 소스코드를 직접 작성해 보고
이전 슬라이드처럼 그림을 그려서 정리하시오

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int num1 = 10, num2 = 20;
```

```
    int *ptr = &num1;
```

```
    printf("num1의 주소값 : %d, ptr이 저장하고 있는 주소값 : %d\n",&num1, ptr);
```

```
    num1+=1;
```

```
    *ptr+=1;
```

```
    printf("num1이 저장하고 있는 값 : %d, ptr이 가리키고 있는 변수가 저장 되어 있는값 : %d\n",&num1, ptr);
```

```
    ptr = &num2;
```

```
    printf("num2의 주소값 : %d, ptr이 저장하고 있는 주소값 : %d\n",&num2, ptr);
```

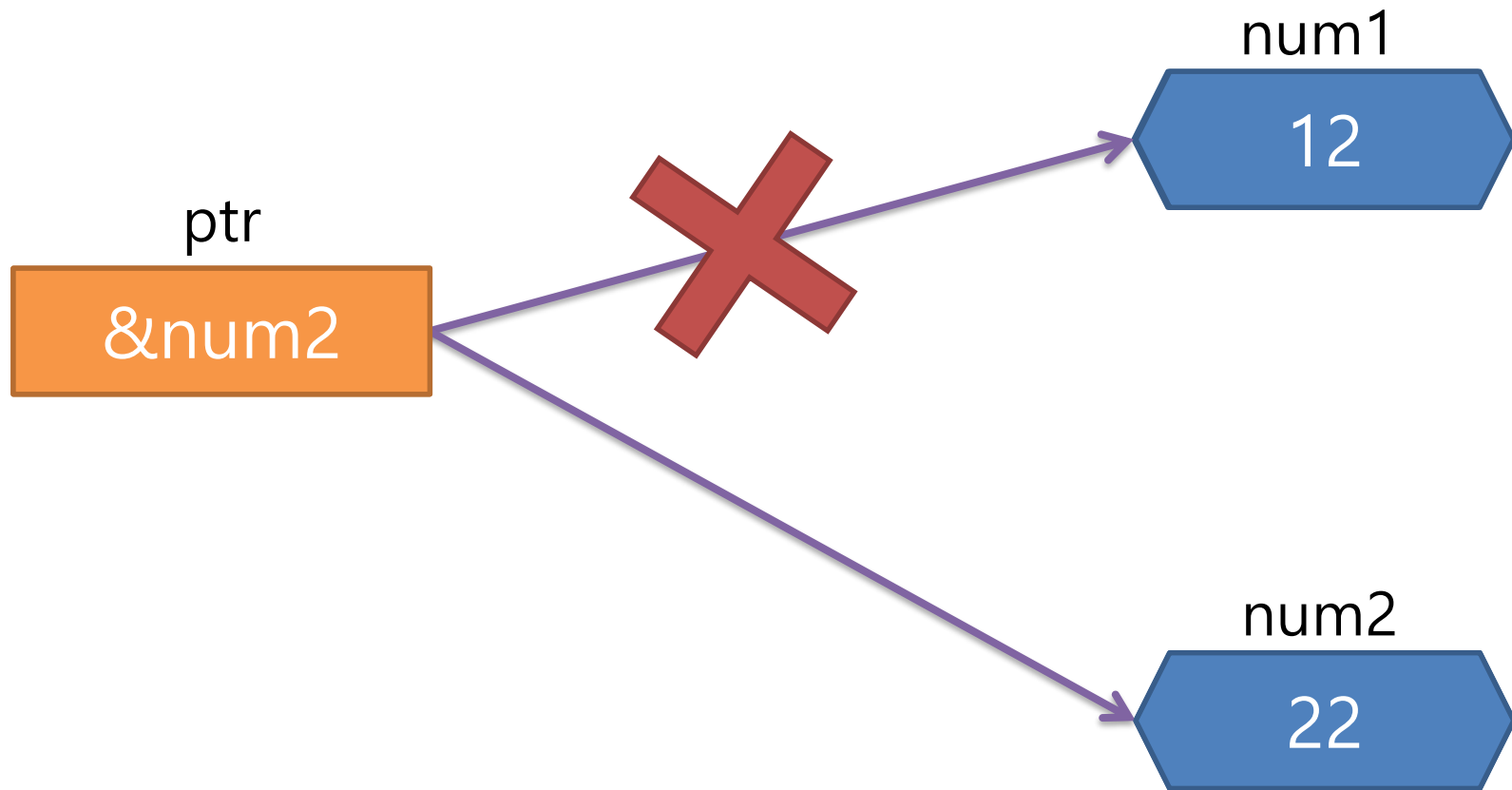
```
    num2+=1;
```

```
    *ptr+=1;
```

```
    printf("num2이 저장하고 있는 값 : %d, ptr이 가리키고 있는 변수가 저장 되어 있는값 : %d\n",&num2, ptr);
```

```
    return 0;
```

```
}
```

4. 함수 호출의 종류

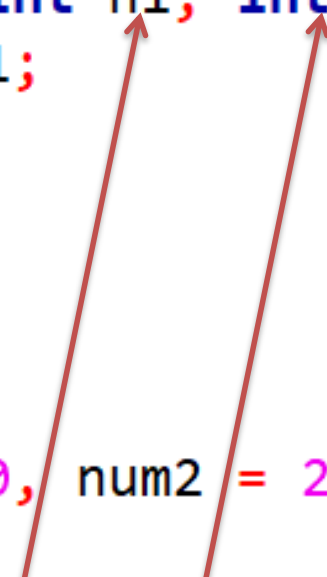
- Call by Value(값에 의한 호출)
 - 매개변수가 일반 변수인 경우
- Call by Reference(참조에 의한 호출)
 - 매개변수가 포인터인 경우

Call by Value

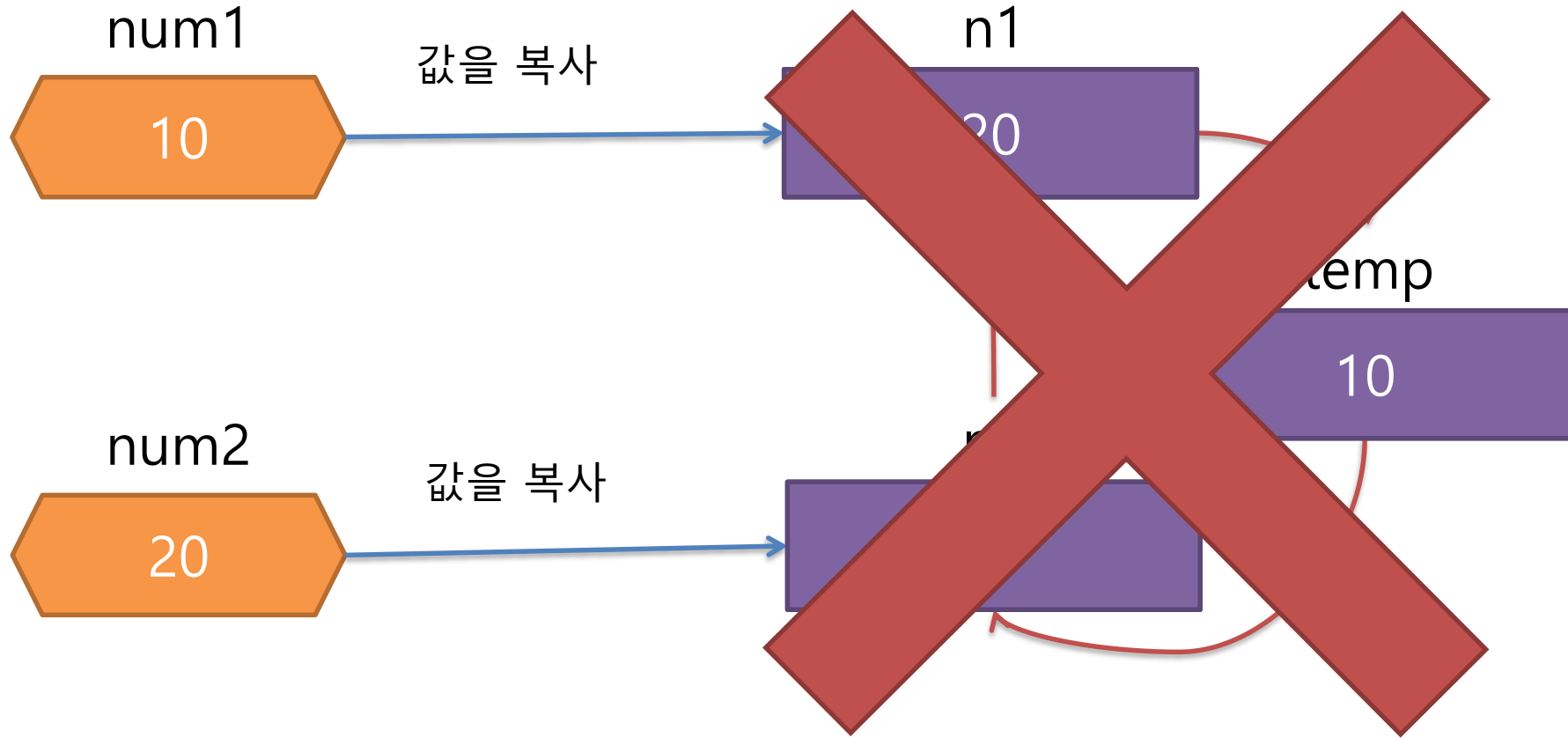
```
#include <stdio.h>
void SwapNumber(int n1, int n2){
    int temp = n1;
    n1 = n2;
    n2 = temp;
}
int main(void)
{
    int num1 = 10, num2 = 20;

    SwapNumber(num1, num2);

    printf("num1 : %d, num2 : %d\n", num1, num2);
    return 0;
}
```



이 때는 값만 복사해서
넘어가기 때문에
num1과 num2의
값이 변화가 없음

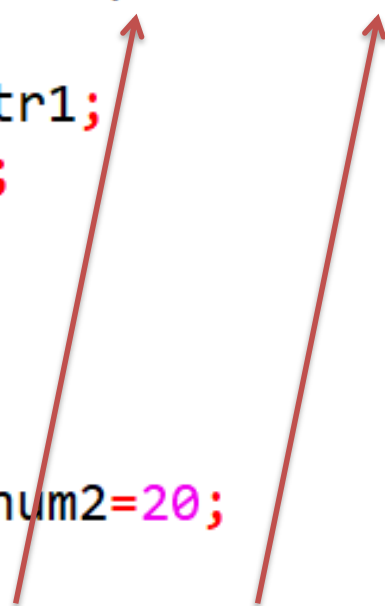


Call by Reference

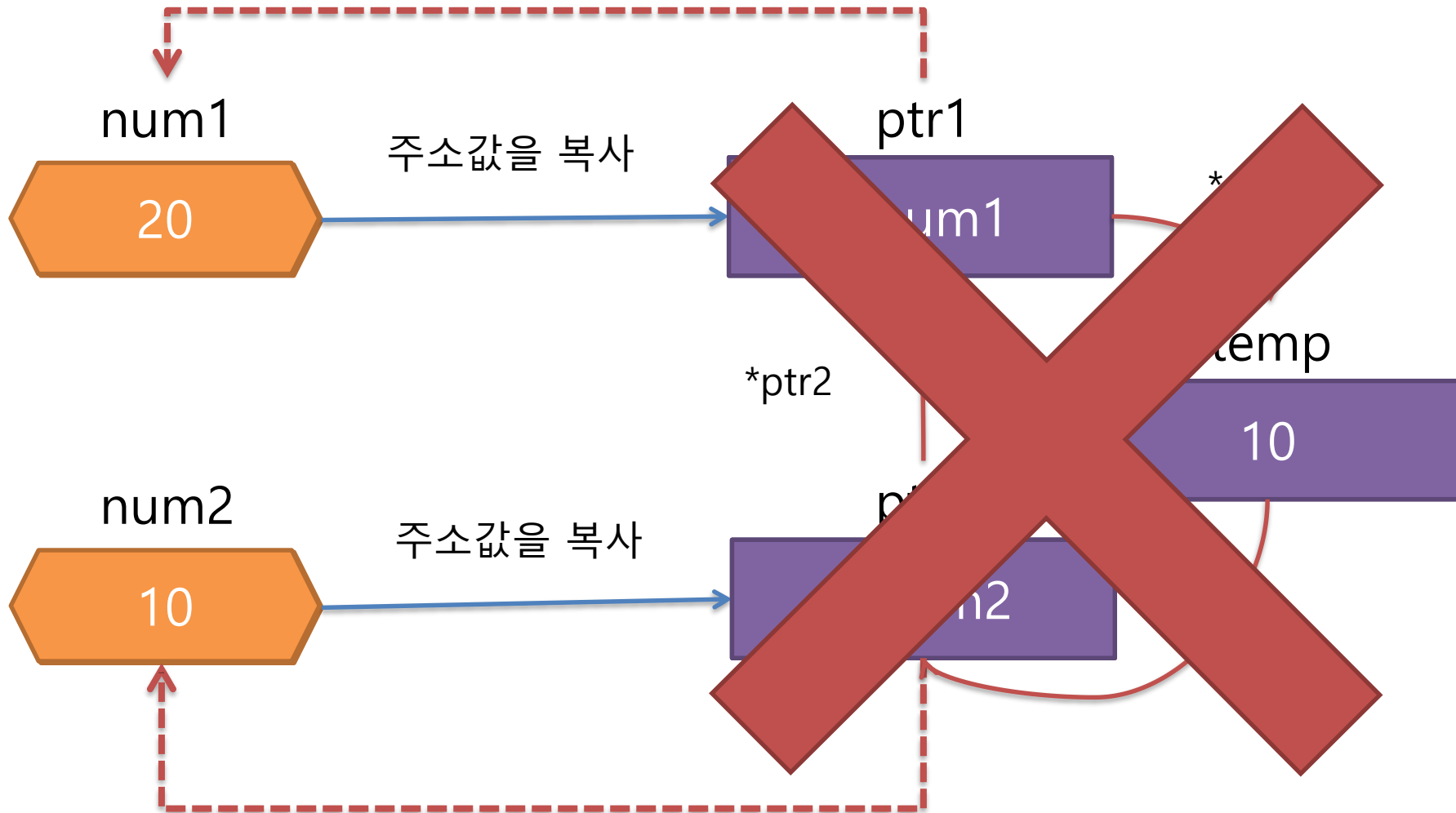
```
#include <stdio.h>
void SwapNumber(int *ptr1, int *ptr2)
{
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}
int main(void)
{
    int num1=10, num2=20;

    SwapNumber(&num1, &num2);

    printf("num1 : %d, num2 : %d\n", num1, num2);
    return 0;
}
```



이 때는 주소값을 복사해서
넘어가기 때문에
ptr1이 num1을 가리킴
ptr2는 num2를 가리킴
그래서 main에 있는
num1과 num2 값이 바뀜



마무리-문제

1. 임의의 숫자를 입력 받은 뒤 절댓값으로 변환하는 함수를 두가지 버전으로 만들고 그 차이점에 대해 설명하시오. 단 함수의 리턴타입은 void, 매개변수는 정수형 포인터로 선언해서 처리하시오.
다른 함수는 리턴타입을 정수형으로 매개변수는 정수형 변수 하나로 선언해서 처리하시오.

답안 파일

답안 링크: <http://gofile.me/4KZc6/JnUj5RDCt>

