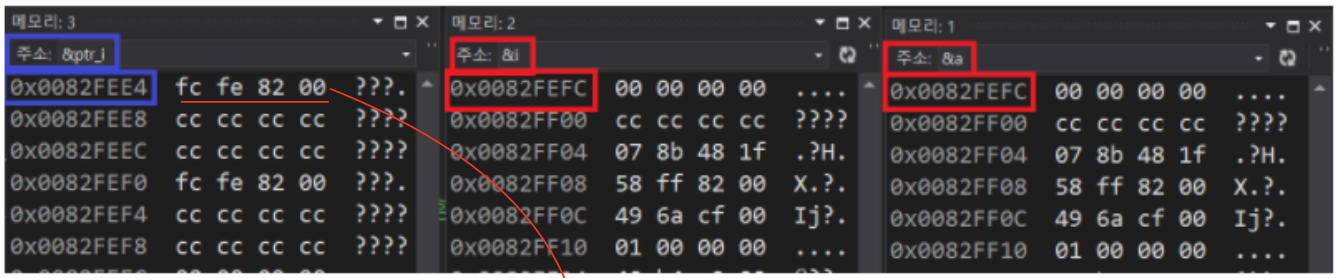
## 1. 참조자 사용법

```
#include<cstdio>
#include<string>


int _tmain(int argc, _TCHAR* argv[])
{
	// (int &) => 참조 자료형 / (int *) => 포인터 자료형
	int a = 0;


	// 참조 자료형(참조형) 을 가지는 변수의 선언
	int& i = a;

	// 포인터 자료형을 가지는 변수 선언
	int* ptr_i = &a;

	return 0;
}
```

위의 코드는 참조형 (참조 자료형의 변수) 와 포인터 (포인터 자료형의 변수)를 각각 생성한 것이다. 아래 자료를 보면서 차이를 확인해 보자




변수 a => 주소 0x 0082FEFC

참조자 i => 주소 0x 0082FEFC

포인터 prt_i => 주소 0x 0082FEE4 이며 해당 주소의 메모리의 값은 0082FEFC 이다.

참조 자료형은 참조자(참조 자료형의 변수) 의 우측에 오는 피연산자 즉 r-value 를 참조한다는 의미이다. 즉 r-value 를 참조 함 에 따라 r-value 와 같은 주소를 가지게 된다. 쉽게 말해 참조자는 r-value 의 또다른 식별자 (이름) 가 되는 것이다.

포인터 자료형은 포인터 (포인터 자료형의 변수) 의 내부에 있는 값을 주소로 인식하여 ' * ' 간접지정 연산자를 사용하여 접근시 포인터 변수의 값을 주소로 보고 해당 주소에 해당하는 메모리로 접근한다는 의미이다.

위와 같은 차이로 보았을 때 포인터는 자신의 값을 주소로 보는 것이다. 하지만 참조자는 참조하는 객체의 또다른 식별자이다. 따라서 참조하는 대상이 없다면 참조자는 선언이 불가하다. 즉 r-value가 있을 때 참조자의 선언이 가능하다. 또한 참조의 대상은 상수가 될수 없다.

## 포인터

**메모리 주소**를 갖는 자료형

## 참조자

**변수의 다른 이름?**

TCPL에도 정확한 정의가 안나와있음

내부 구현은 포인터로 되어있다 생각하면 편할듯.. 컴파일러에서 할당 없게 최적화 해줄 때도 있을거고.... 딱 이정도

# 객체를 참조할 때 쓰인다

## 포인터

```
int * intPointer = tempInt;
*intPointer = 50; ← 역참조
```

## 참조자

```
int& intReference = tempInt;
intReference = 50; ← 역참조
```

## 참조자는 Null이 될 수 없다.

무조건 메모리에 공간이 할당된 객체를 참조해야함.
-> 선언과 함께 초기화를 해주어야 한다.

그렇지 않다면 포인터를 사용해야한다.

## 참조자의 장점

무조건 유효한 객체를 가리키고 있다는 불변속성을 가진다.

-> 인자 유효성 검사를 할 필요가 없음.

# 포인터의 장점

1. 주소값을 바꾸어 다른 객체들을 참조할 수 있다.

참조자는 변경 불가능

2. 무효한 상태(null)로 존재할 수 있다.
-> 참조자는 무조건 유효해야 함