

## 1번째 코드

```
#include <stdio.h>

int main(void)
{
    int num1 = 3;        // int형 num1 은 3으로 초기화
    int *p;              // int형의 포인터 p 선언
    p = &num1;           // p는 num1의 주소 값을 전달
    *p = 4;              // p가 가진 주소 값 속 실질적 값을 4로 변경

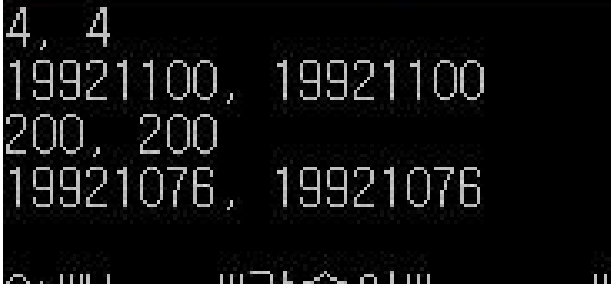
    printf("%d, %d \n", num1, *p);
                        // num1의 값과 p가 가리키는 값인 4가 출력

    printf("%d, %d \n", &num1, p);
                        // num1의 주소 값 (p와 동일)이 출력

    int num2 = 100;      // int형 num2는 100으로 초기화
    p = &num2;           // int* p에 num2의 주소 값을 전달
    *p = 200;           // p가 가진 주소 값 속 실질 값을 200으로 변경

    printf("%d, %d \n", num2, *p); // num2의 값 200 출력
    printf("%d, %d \n", &num2, p); // num2의 주소 값 출력
}
```

 Microsoft Visual Studio 디버



```
4, 4
19921100, 19921100
200, 200
19921076, 19921076
```

## 2번째 코드

```
#include <stdio.h>

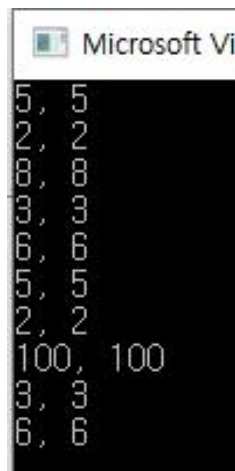
int main(void)
{
    int ary[5] = { 5, 2, 8, 3, 6 };
    int *p;          // int형 포인터 생성
    p = ary           // ary의 시작주소 전달

    for (int i = 0; i < 5; i++)
        printf("%d, %d \n", ary[i], *(p + i));
        // ary[i]값이 두 번씩 출력

    p++;             // 포인터 값 1 증가, int형 주소이므로 4가 증가

    p[1] = 100;
    // 기존 p[2] 값을 100으로 변경하였기 때문에
    // ary = {5, 2, 100, 3, 6}

    for (int i = 0; i < 5; i++)
        printf("%d, %d \n", ary[i], *(p - 1 + i)); // ?
        // ary = {5, 2, 100, 3, 6} 순서대로 두 번 씩 출력
}
```



```
Microsoft Vi
5, 5
2, 2
8, 8
3, 3
6, 6
5, 5
2, 2
100, 100
3, 3
6, 6
```