



# C언어 (CSE2035) (Chap2. Pointers) (1/2)

---

**Sungwon Jung, Ph.D.**

**Dept. of Computer Science and Engineering**

**Sogang University**

**Seoul, Korea**

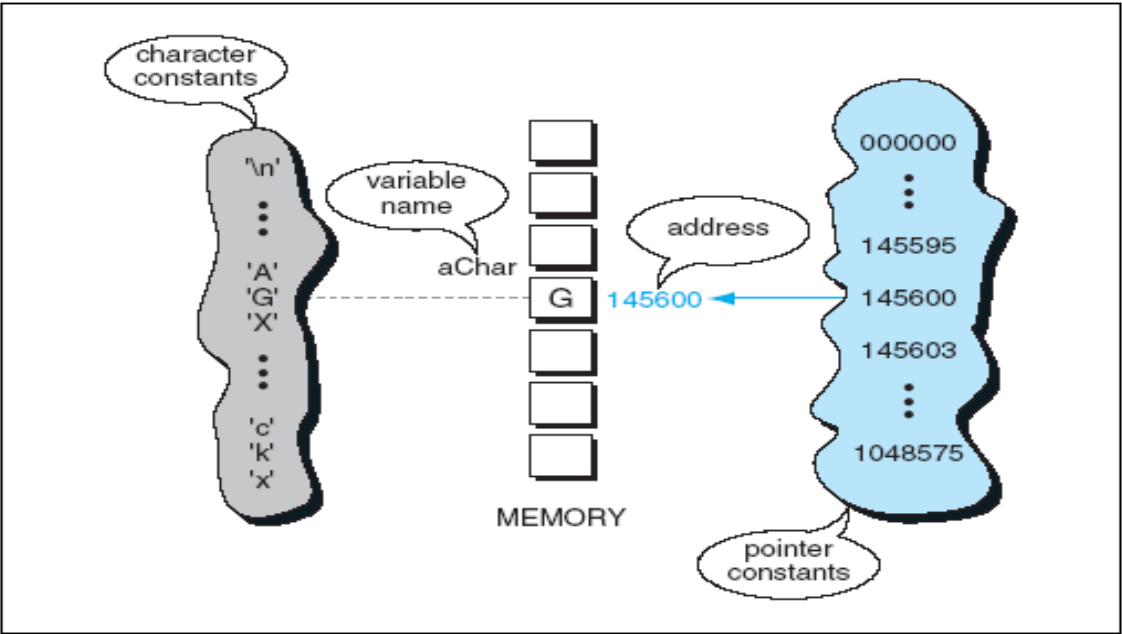
**Tel: +82-2-705-8930**

**Email : [jungsung@sogang.ac.kr](mailto:jungsung@sogang.ac.kr)**

# Concepts of pointers

## ■ Pointers

- 포인터는 메모리의 주소 값을 저장하기 위한 변수이다.
- 메모리에는 byte 단위로 주소가 지정되어 있다.
- 포인터 상수(pointer constants)는 이러한 메모리 주소(address)로 정의할 수 있다.





# Concepts of pointers

## ■ Pointer values

- 포인터 상수는 메모리의 주소(address)로 정의할 수 있다.
- C프로그램에서는 이 주소 값(address value)을 저장할 수 있을 뿐만 아니라 식별하여 사용할 수 있다.
- 주소 연산자(&)를 사용하면 해당 변수의 주소 값을 알 수 있다.

& 변수 이름



변수가 할당된 메모리의 시작 주소값

- 제공되는 주소 값은 컴퓨터에 따라, 프로그램을 실행시키는 시간에 따라 다른 값이 될 것이다.
  - 메모리상의 어떤 공간에 프로그램이 올라갈지는 실행시킬 때마다 다르다

# Concepts of pointers

## ■ Pointer values

```
// Print character addresses
#include <stdio.h>

int main (void)
{
    // Local Declarations
    char a;
    char b;
    // Statements
    printf ("%p\n %p\n", &a, &b);
    return 0;
} // main
```

주소 연산자를 이용한 변수 a, b의 출력

변수 a, b에 할당된  
메모리 주소

a 142300      b 142301

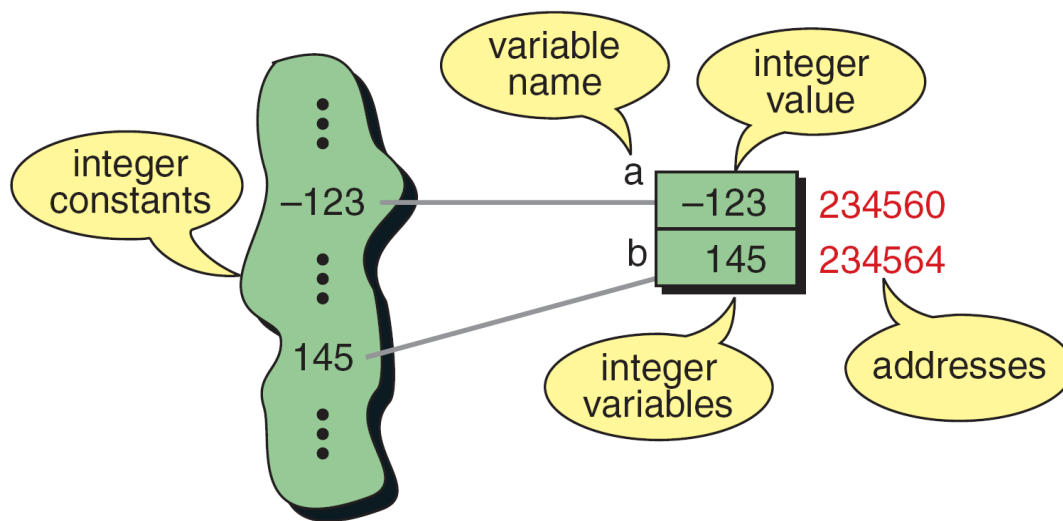
142300  
142301

- 컴퓨터는 변수 a, b가 선언될 때 저장공간을 1byte씩 제공
- 위 프로그램의 결과로 변수 a, b의 주소 값 142300, 142301이 출력

# Concepts of pointers

## ■ Pointer values

- 대부분의 컴퓨터는 변수의 메모리 영역을 할당할 때 인접한 메모리 영역을 할당하여 준다.
- character constants를 위한 메모리 저장 공간은 1byte이다.
- integer constants를 위한 메모리 저장 공간은 2 혹은 4bytes이다.
  - integer 변수의 주소는 4byte (혹은 2byte)중의 첫 번째 byte의 주소이다.



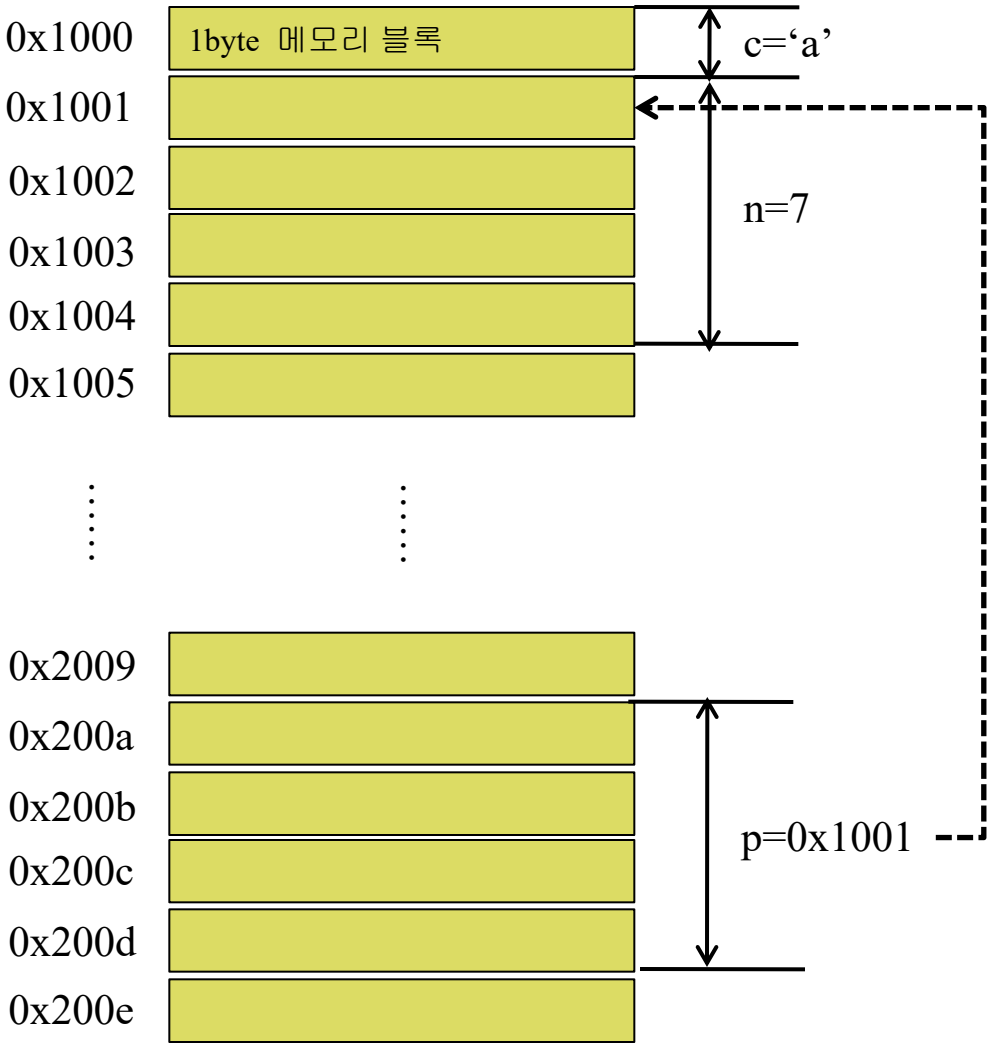
# Concepts of pointers

```
int main(void)
{
    char c='a';
    int n=7;
    int *p;

    p = &n;
}
```



address

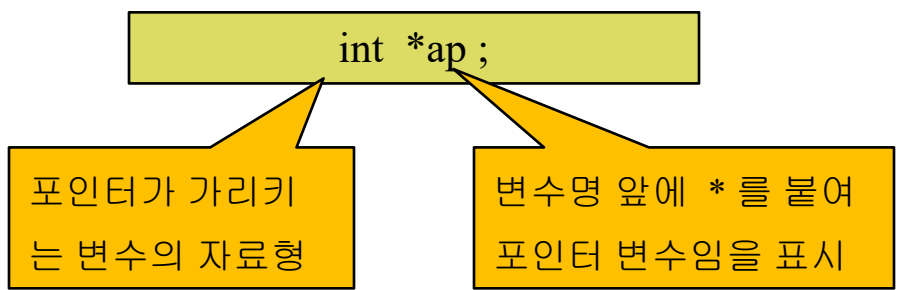


<포인터 변수의 메모리 할당>

# Pointer variables

## ■ Pointer variables

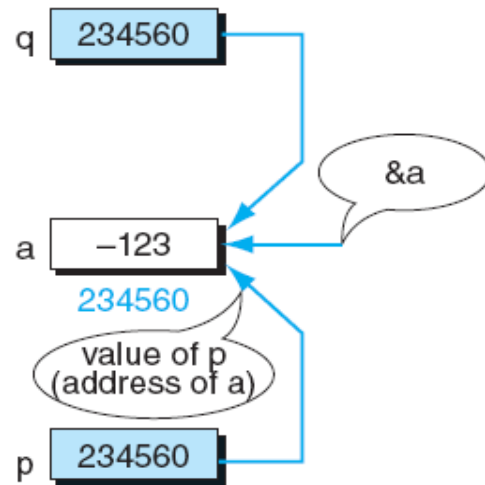
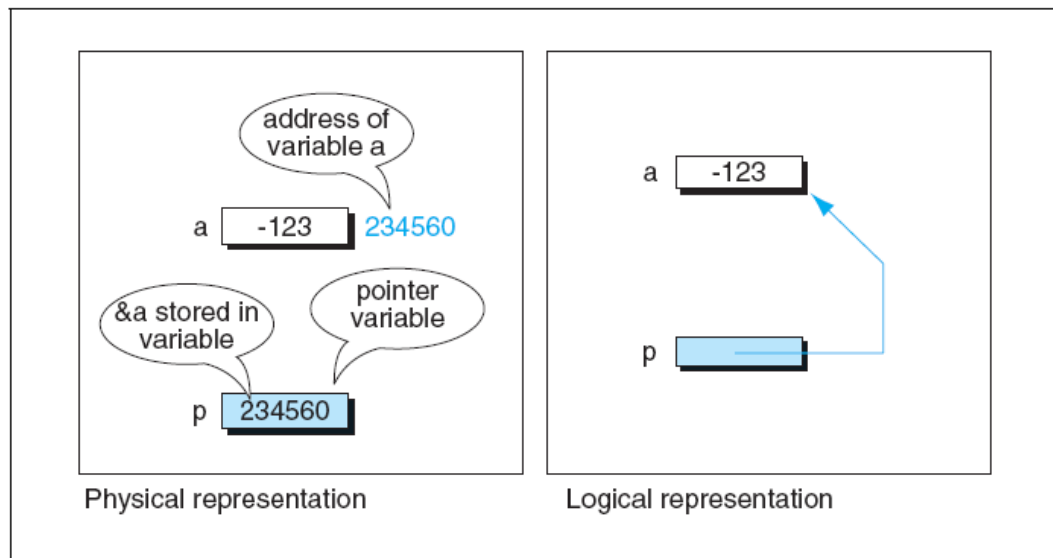
- 포인터 값(즉, 주소값)을 저장하는 변수를 포인터 변수라고 한다.
- 포인터가 가리키는 변수의 형태에 따라 포인터 변수의 형태도 달라진다.



< int형 변수의 시작주소값을 저장하는 포인터변수 ap >

- 포인터 변수는 주로 자신이 가리키는 기억공간을 참조하는데 사용되며, 이 또한 변수이기 때문에 저장된 값을 출력할 수 있다.

# Pointer variables



- 현재 integer 변수 `a`에 `-123`이라는 값이 저장되어있고 제공된 변수 `a`의 주소 값은 `234560`이라고 하자.
- 포인터 변수 `p`는 변수 `a`의 주소 값 `234560`을 값으로 저장함으로써 변수 `a`를 참조하게 된다.
- 하나의 변수 `a`를 참조하는 포인터 변수가 두 개 이상일 수도 있다.

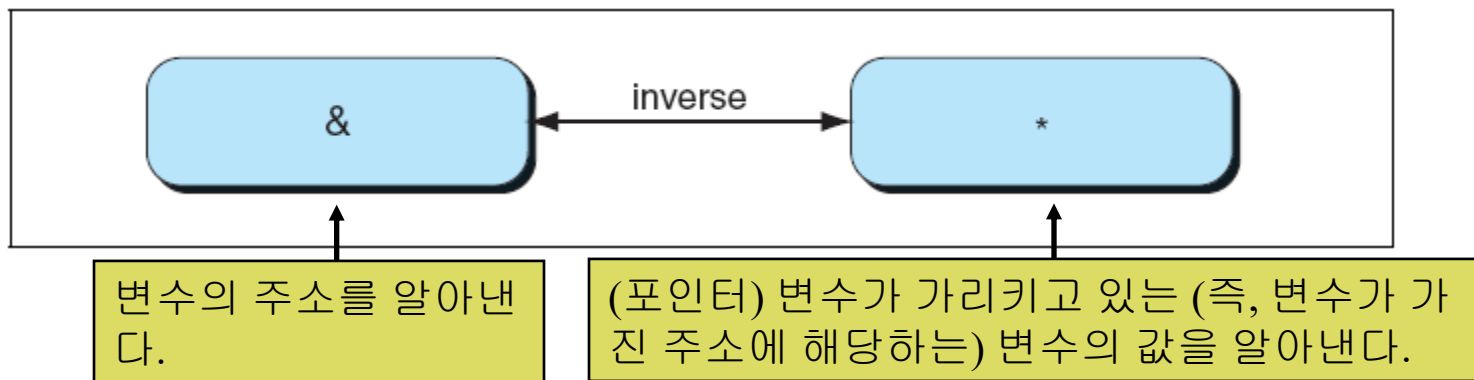


# Accessing variables through pointers

## ■ Indirection operator (\*)

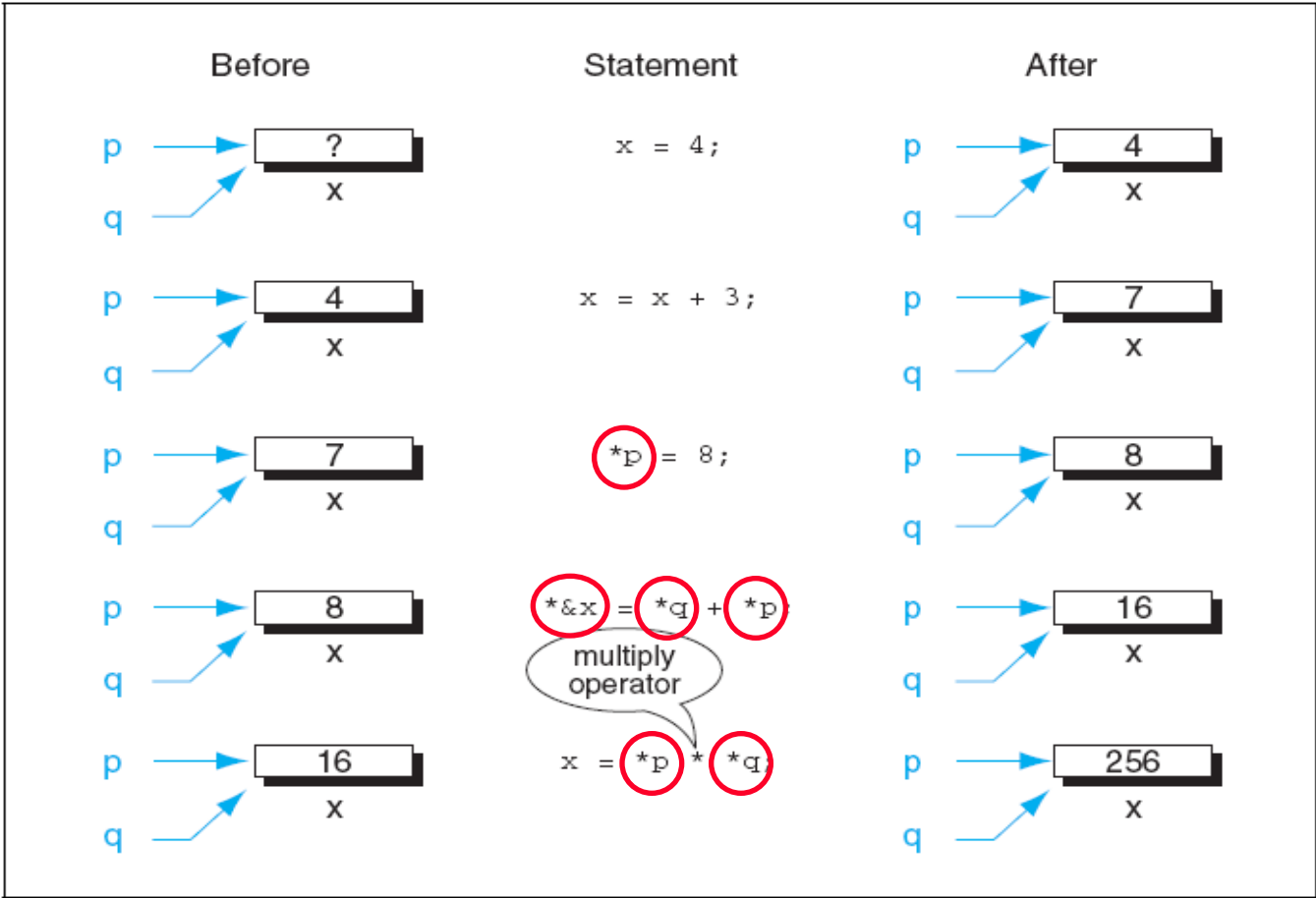
- 포인터 변수 앞에 \* 연산자를 붙이게 되면, 포인터가 가리키는 메모리 공간에 존재하는 값을 참조하겠다는 뜻이다.
- 다음 3개의 statement는 모두 a의 값을 1증가 시킨다.

```
a++;    a = a + 1;    *p = *p + 1;
```



# Accessing variables through pointers

- p, q가 변수 x를 가리키는 포인터 변수일 때, 다음과 같이 변수 x를 access하는 여러 가지 방법이 있다.



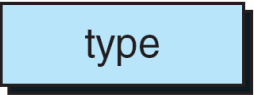
○ 표 한 것은 모두 'x'에 해당한다.



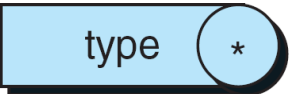
# Pointer declaration and definition

- **Pointer variable declaration**

data declaration



pointer declaration



- 서로 다른 타입을 참조하는 포인터 변수의 선언

|                       |  |                        |  |
|-----------------------|--|------------------------|--|
| <code>char a;</code>  |  | <code>char* p;</code>  |  |
| <code>int n;</code>   |  | <code>int* q;</code>   |  |
| <code>float x;</code> |  | <code>float* r;</code> |  |

# Pointer declaration and definition

## ■ 예제 프로그램 - 변수에 저장되는 값 비교

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a=10;
6     int *ap = &a;
7
8     printf("a : %d\n", a);
9     printf("*ap : %d\n", *ap);
10    printf("&a : %u\n", &a);
11    printf("ap : %u\n", ap);
12    printf("&ap : %u\n", &ap);
13
14    return 0;
15 }
```

a에 저장된 값

ap 가 가리키는 곳에 저장된 값

a의 시작 주소 값

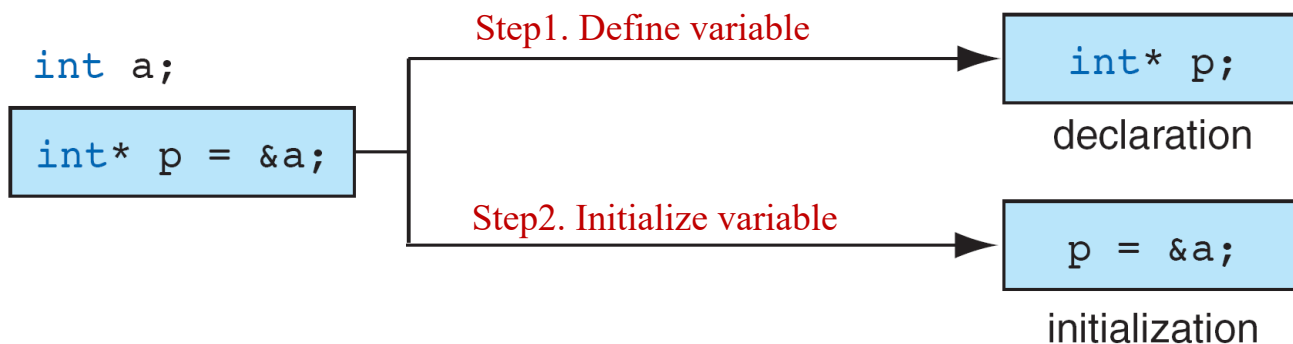
ap에 저장된 주소값

포인터변수 ap 의 시작주소값

```
[root@mclab chap9]# vi chap9-1.c
[root@mclab chap9]# gcc -o chap9-1 chap9-1.c
[root@mclab chap9]# ./chap9-1
a : 10
*ap : 10
&a : 3218317556
ap : 3218317556
&ap : 3218317552
[root@mclab chap9]#
```

# Initialization of pointer variables

- 일반적인 변수가 선언된 후에 초기화되지 않으면 쓰레기 값을 갖는 것처럼 포인터 변수도 초기화하지 않으면 엉뚱한 곳을 가리키게 된다.
- 포인터 변수는 주소값을 가지므로 주소 연산자(&)를 이용하여 초기화한다.



- 포인터 변수가 아무 변수도 가리키지 않도록 하려면 다음과 같이 초기화한다.

```
*p = NULL;
```



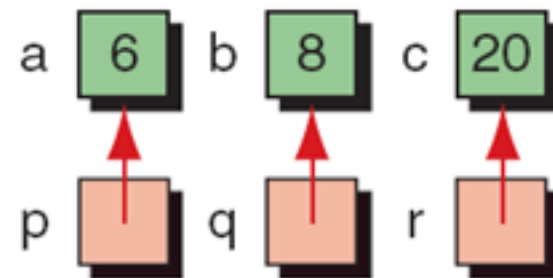
# 포인터 사용의 예

| 선언 및 초기화   |   |  |
|--|---|--|
| <pre>int i = 3, j =5, *p = &amp;i, *q = &amp;j, *r;<br/>double x;</pre>                                      |   |  |
| 수 식  | 등가 수식   | 값  |
| <pre>p == &amp; i;<br/>* * &amp; p;<br/>r = &amp; x;<br/>7 * * p / * q + 7;<br/>* (r = &amp; j) *= *p;</pre> | <pre>p == (&amp; i);<br/>* (* (&amp; p));<br/>r = (&amp; x);<br/>(((7 * (* p))) / (* q)) + 7;<br/>(* (r = (&amp; j))) *= (* p);</pre> | <pre>1<br/>3<br/>/* illegal */<br/>11<br/>15</pre> |

# Example program

- 예제 프로그램 - 포인터 변수를 사용한 예제

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int a, b, c;
6     int *p, *q, *r;
7
8     a = 6;
9     b = 2;
10    p = &b;
11
12    q = p;
13    r = &c;
14
15    p = &a;
16    *q = 8;
17
18    *r = *p
19
20    *r = a + *q + *&c;
21
22    printf("%d %d %d\n", a, b, c);
23    printf("%d %d %d\n", *p, *q, *r);
24
25    return 0;
26 }
```



```
[root@mclab chap9]# vi chap9-2.c
[root@mclab chap9]# gcc -o chap9-2 chap9-2.c
[root@mclab chap9]# ./chap9-2
6 8 20
6 8 20
[root@mclab chap9]#
```