7.4 The Relationship Between Arrays and Pointers

```
int a[20], *p ;
```

a : without index, array name is **pointer value**
p : **pointer variable**

(차이점) p 에는 pointer 값을 할당할 수 있다.
a는 포인터와 동일하게 사용 가능하지만, 항상 일정한 값 (주소)를 지칭하고 있다.
따라서 a에는 다른 주소 값을 넣을 수 없다.

```
a[i] = 1 ;
*(a+i) = 1   ;
```

```
p = a ;
p = &a[0] ;
p[i] = 1 ;
*(p+i) = 1 ;
```

* Invalid expressions

```
a = p ;          a cannot be assigned
++a ;
a += 2 ;
```

&a ; pointer to address

ex) summation example

```
#define N 100
:
int i,a[N],*p ;

(1)
for(i=0;i<N; i++)
        sum += a[i]   ;
(2)
for(i=0,p=a ;i<N; i++)
        sum += p[i]   ;
(3)
for(i=0;i<N; i++)
        sum += *(a+i);
(4)
for(i=0,p=a;i<N; i++)
        sum += *(p+i);
(5)
for(p=a;p<&a[N]; ++p)
        sum+ = *p ;
```

7.5 Pointer Arithmetic and Element Size

* Pointer 의 증감

p+1 : 포인터의 값을 하나 증가시킨다.
실제로 그 값은 p 가 어떤 값을 가리키느냐에 따라 증감 값이 다르다.

ex) char *c ;
int *p ;

c을 증가시킬 경우 : 1 씩 증가
p를 증가시킬 경우 : 2 씩 증가
실제로 주소 값이 얼마나 증가하느냐는 문제시 되지 않음. 마치 array index 가 증가하는 것과 동일하게 생각하면 된다.

ex) → see page 255 example

7.6 Array as Function Arguments

* Array in formal parameters in function definition
- actually pointer variable
- elements are not passed only address are passed.
- if you change the value of elements, original values are changed.

ex)

```
double sum(double a[], int n)
{
        :
}

double sum(double *a, int n)
{
        :
}
```

see example in page 256 to 257

6.7 An Exmaple Bubble sort

- homework

for given array ; page 258

write bubble sort program which shows every sorting step as in page. 258

email -address : homewk2@thermo.korea.ac.kr (4/23 –4/28)