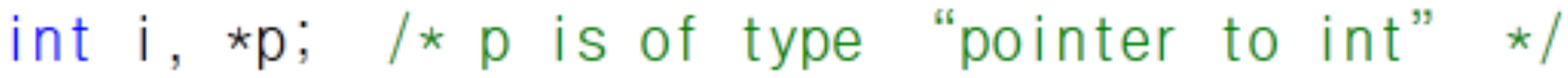
C Programming

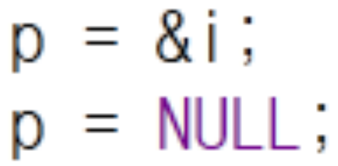
Practice 8

Pointer Declaration and Assignment

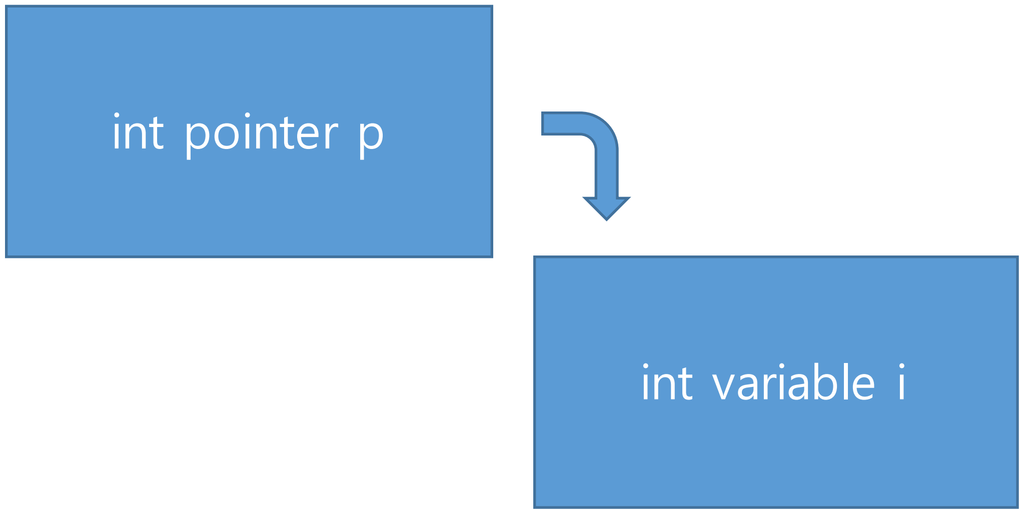
1. Size of pointer = 4 bytes
2. Save the address value



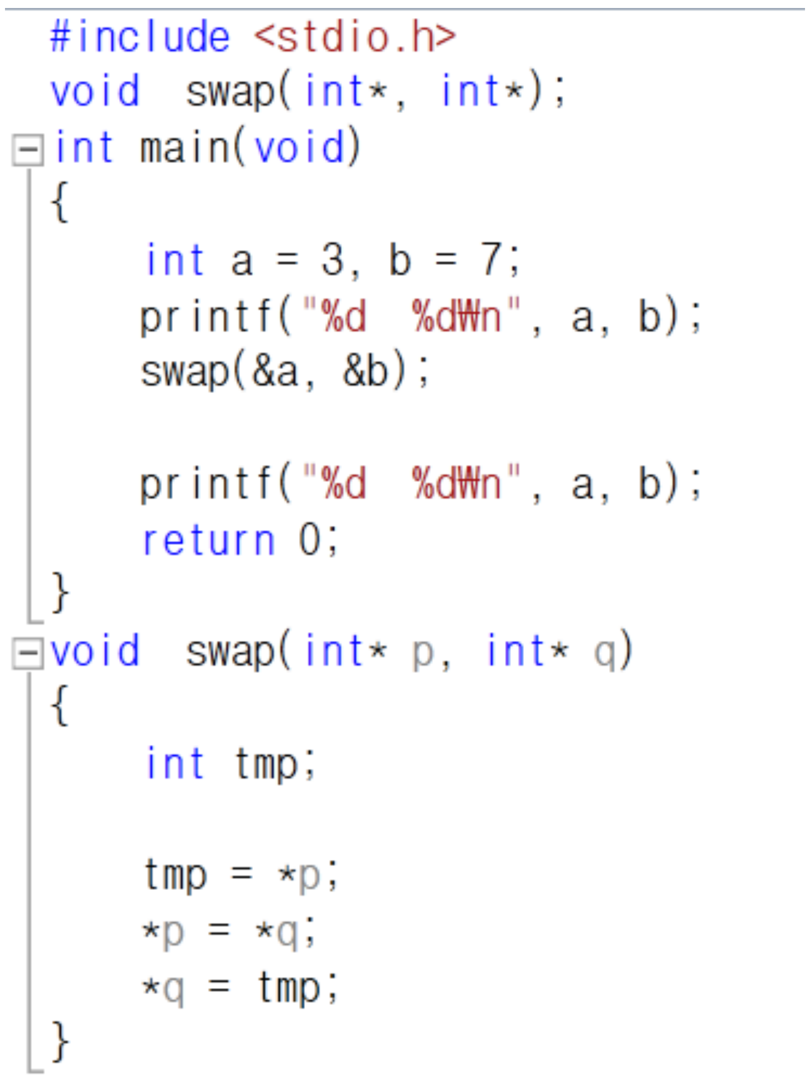
```
int i, *p;  /* p is of type "pointer to int"  */
```



```
p = &i;
p = NULL;
```



Call-by-Reference

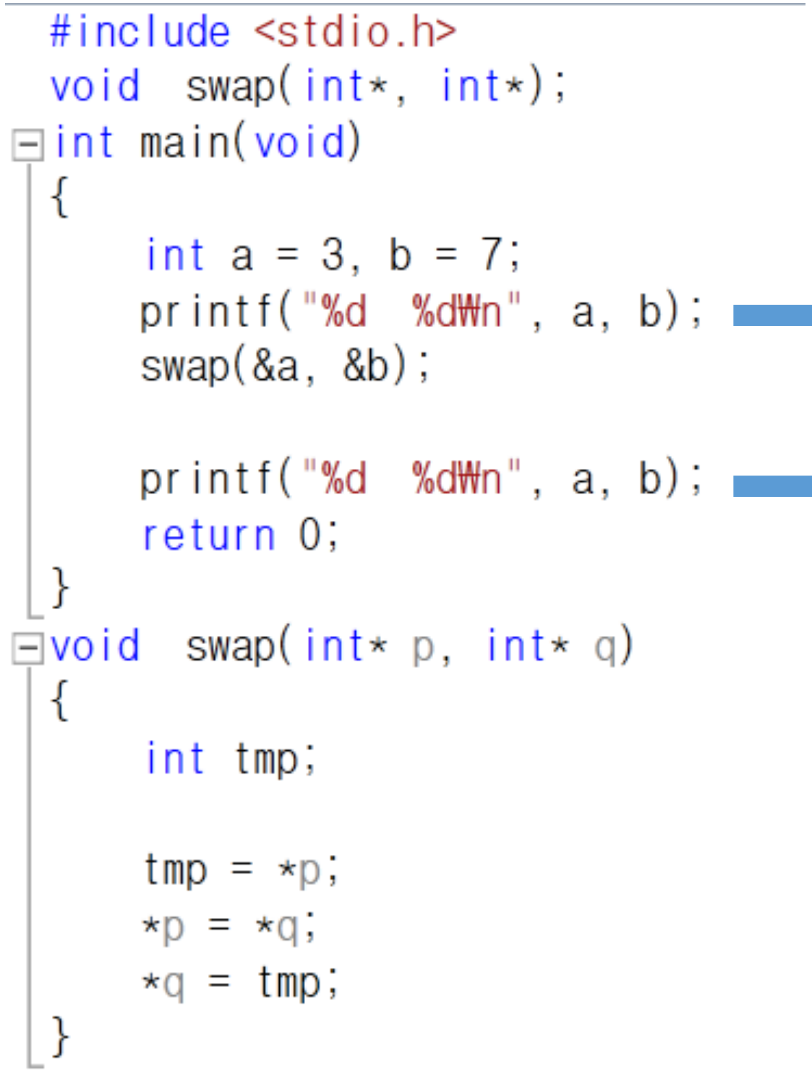


```
#include <stdio.h>
void  swap(int*, int*);
int main(void)
{
    int a = 3, b = 7;
    printf("%d  %d₩n", a, b);
    swap(&a, &b);

    printf("%d  %d₩n", a, b);
    return 0;
}
void  swap(int* p, int* q)
{
    int tmp;

    tmp = *p;
    *p = *q;
    *q = tmp;
}
```

Call-by-Reference



```c
#include <stdio.h>
void  swap(int*, int*);
int main(void)
{
    int a = 3, b = 7;
    printf("%d  %d₩n", a, b);
    swap(&a, &b);

    printf("%d  %d₩n", a, b);
    return 0;
}
void  swap(int* p, int* q)
{
    int tmp;

    tmp = *p;
    *p = *q;
    *q = tmp;
}
```

3 7

7 3

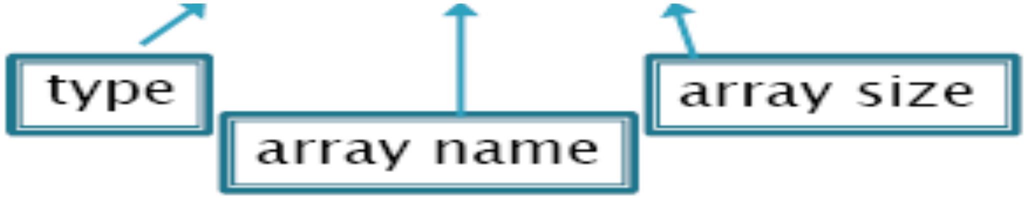
Array

- A sequence of data items that:
  - are of the same type,
  - can be indexed, and
  - are stored contiguously.

Array

- Define array

```
int grade [3];
int grade [3] = {0};
char characters[3];
```

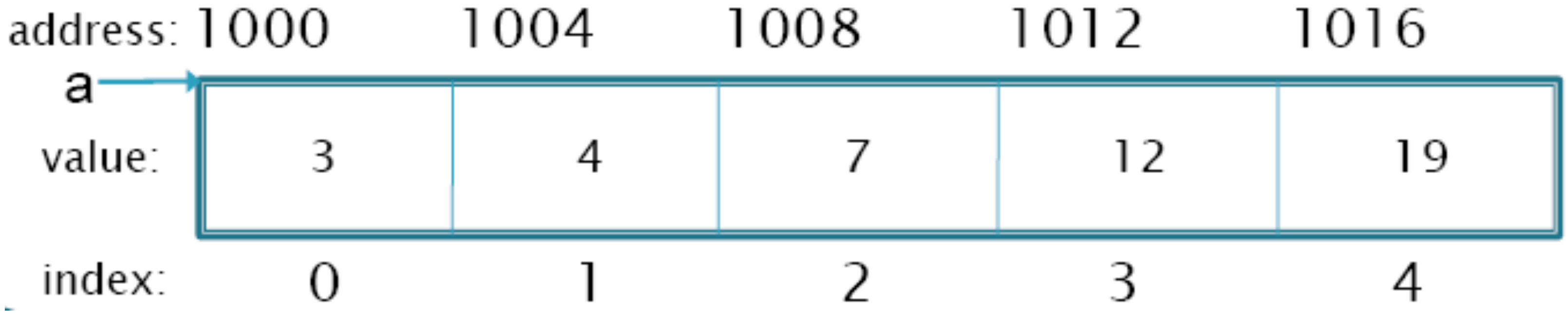


```
int a[] = {3, 4, 5};
```

Array

- The indexing of array elements always starts at 0.

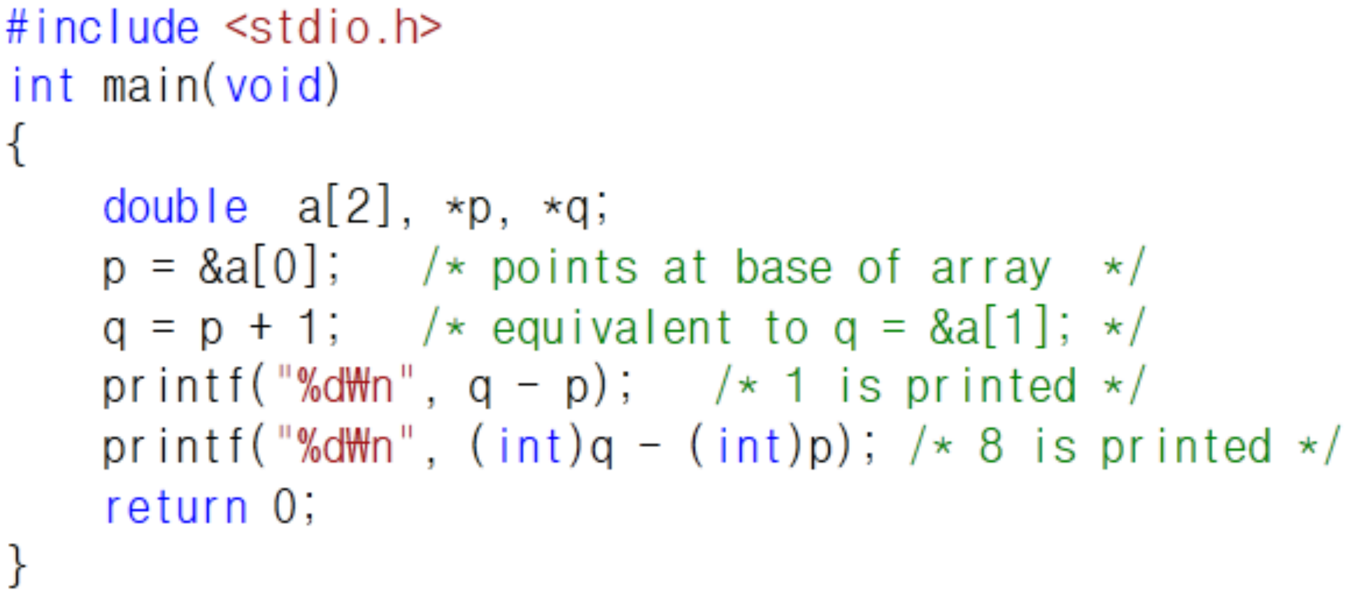
int a[5] = {3, 4, 7, 12, 19}



address:	1000	1004	1008	1012	1016
value:	3	4	7	12	19
index:	0	1	2	3	4

Array

- An array name is pointer value. (a fixed address)



```
#include <stdio.h>
int main(void)
{
    double  a[2], *p, *q;
    p = &a[0];    /* points at base of array  */
    q = p + 1;    /* equivalent to q = &a[1]; */
    printf("%d₩n", q - p);    /* 1 is printed */
    printf("%d₩n", (int)q - (int)p); /* 8 is printed */
    return 0;
}
```

Array as a function argument

```
double sum(double a[], int n)  /* n is the size a[] */
{
   int     i;
   double  sum = 0.0;
   for (i = 0; i < n; ++i)
       sum += a[i];
   return sum;
}
```

‖

```
double sum(double *a, int n)   /* n is the size a[] */
{
   .....
```

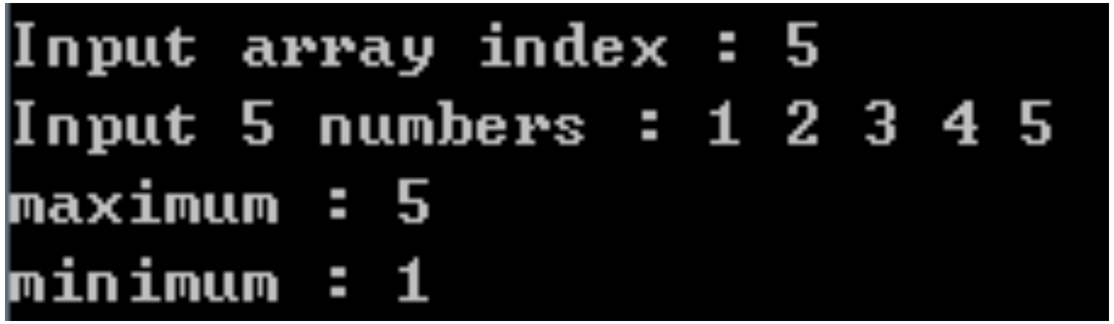
Two-Dimensional array

int a[3][5];

	col 1	col 2	col 3	col 4	col 5
row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]
row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]
row 3	a[2][0]	a[2][1]	a[2][2]	a[2][3]	a[2][4]

Homework 14 - Maximum number

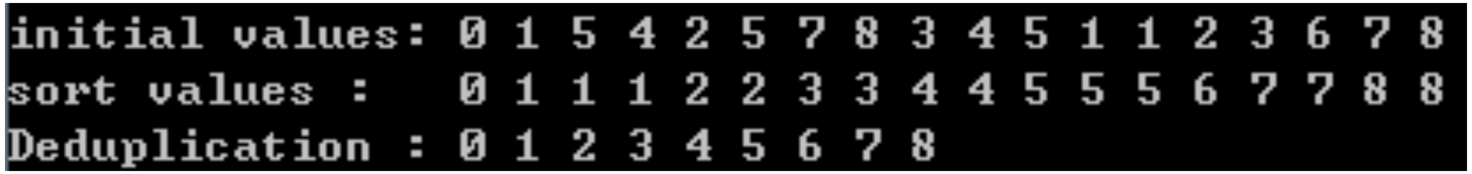
- Input array index N
- Input the number of N
- Print the maximum value and the minimum value



```
Input array index : 5
Input 5 numbers : 1 2 3 4 5
maximum : 5
minimum : 1
```

Homework 15 - Sorting

- Define the array : {0, 1, 5, 4, 2, 5, 7, 8, 3, 4, 5, 1, 1, 2, 3, 6, 7, 8}
- Sort by ascending
- Deduplication



```
initial values: 0 1 5 4 2 5 7 8 3 4 5 1 1 2 3 6 7 8
sort values :   0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
Deduplication : 0 1 2 3 4 5 6 7 8
```

Homework form

- Homework submission e-mail:

hizorro99@naver.com

- E-mail title: day(Thursday or Friday)_name_#week
  - Ex) Friday_james_week8
  - Ex) 목요일반_장원철_8주차
- File title: student id_name_#.c
  - Ex) 2014123456_james_14.c (or .cpp)
  - Ex) 2014123456_james_15.c (or .cpp)