

C Programming

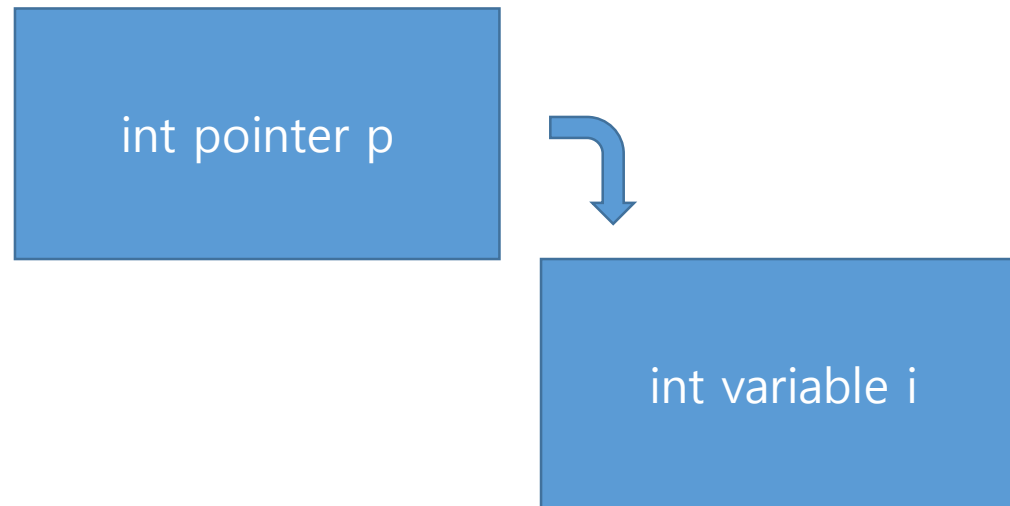
Practice 8

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-value

```
#include<stdio.h>
```

```
void swap(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

Result?

x = 5, y = 10

```
int main()
{
    int x = 5, y = 10;
    swap(x, y);
    printf("x = %d, y = %d\n", x, y);
}
```

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;
}
```

→ 3 7

→ 7 3

Array

- Define array

```
int grade [3];
```

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

```
char characters[3];
```

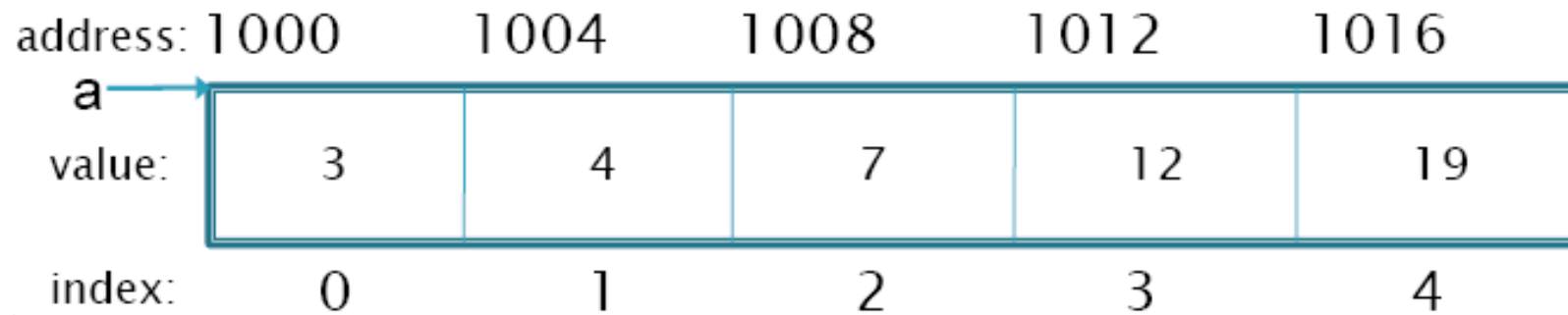


```
int a[] = {3, 4, 5};
```

Array

- The indexing of array elements always starts at 0.

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



Array & Pointer

- $a[i] \iff *(a + i)$
- $p[i] \iff *(p + i)$
- Pointer variable can have different values, but array name is not
- $p = a + i ;$
- $a = q ; /* \text{error} */$

Two-dimensional array

- Declaration
 - `int a[3][5];`
- In fact, the elements are stored sequentially

	1 Column	2 Column	3 Column	4 Column	5 Column
1 Row	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>	<code>a[0][4]</code>
2 Row	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>	<code>a[1][4]</code>
3 Row	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>	<code>a[2][3]</code>	<code>a[2][4]</code>

Two-dimensional array

- The same expressions as $a[i][j]$

1. $*(a[i] + j)$

2. $((*(a + i)) [j]$

3. $((*(a + i)) + j)$

Homework 16 – Bubble 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
- Print process 0~17

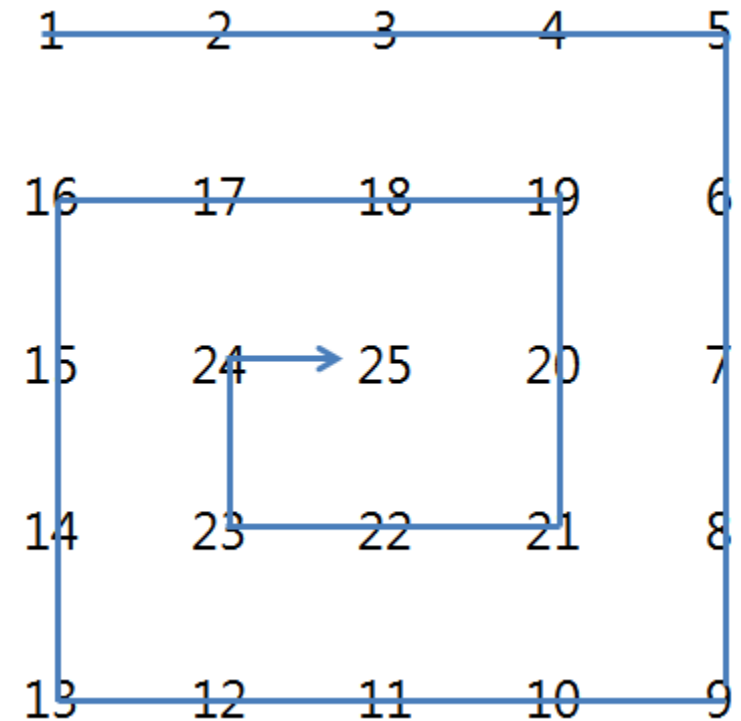
```
C:\Windows\system32\cmd.exe
initial values: 0 1 5 4 2 5 7 8 3 4 5 1 1 2 3 6 7 8
process 0:      0 1 4 2 5 5 7 3 4 5 1 1 2 3 6 7 8 8
process 1:      0 1 2 4 5 5 3 4 5 1 1 2 3 6 7 7 8 8
process 2:      0 1 2 4 5 3 4 5 1 1 2 3 5 6 7 7 8 8
process 3:      0 1 2 4 3 4 5 1 1 2 3 5 5 6 7 7 8 8
process 4:      0 1 2 3 4 4 1 1 2 3 5 5 5 6 7 7 8 8
process 5:      0 1 2 3 4 1 1 2 3 4 5 5 5 6 7 7 8 8
process 6:      0 1 2 3 1 1 2 3 4 4 5 5 5 6 7 7 8 8
process 7:      0 1 2 1 1 2 3 3 4 4 5 5 5 6 7 7 8 8
process 8:      0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 9:      0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 10:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 11:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 12:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 13:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 14:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 15:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 16:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 8
process 17:     0 1 1 1 2 2 3 3 4 4 5 5 5 6 7 7 8 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 17 – NxN land snail array

- Input array size N (1~20)
- Using Two-Dimensional array

```
Input a number : 3
  1   2   3
  8   9   4
  7   6   5
```

```
Input a number : 5
  1   2   3   4   5
 16  17  18  19   6
 15  24  25  20   7
 14  23  22  21   8
 13  12  11  10   9
```



Homework form

- Homework submission e-mail:

hizorro99@naver.com

- E-mail title: day(Thursday or Friday)_name_#week
 - Ex) Friday_james_week10
 - Ex) 목요일반_장원철_10주차
- File title: student id_name_#.c
 - Ex) 2014123456_james_16.c (or .cpp)
 - Ex) 2014123456_james_17.c (or .cpp)