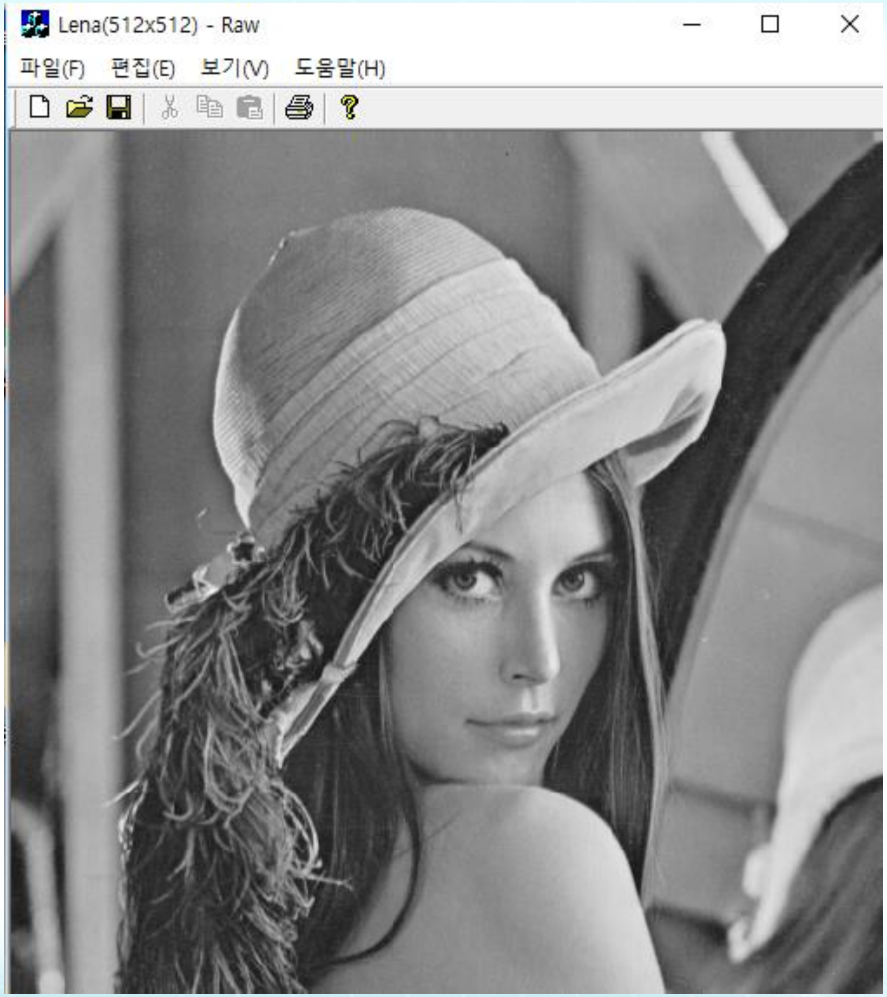
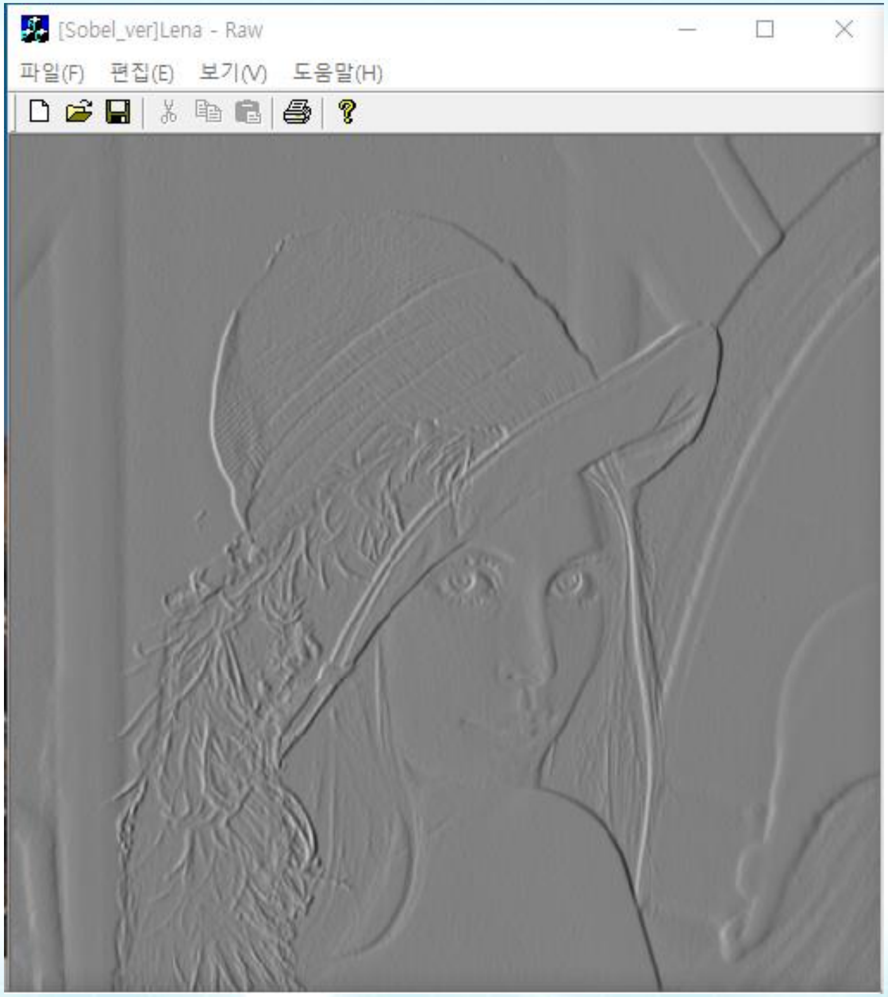
과제 설명 수정사항

- Assignment#3 : sharpening
  - Sharpening mask의 경우에 offset으로 128을 더하면 pixel value가 지나치게 shift되어 밝아지는 현상이 나타나므로 sharpening의 경우에만 128을 더하지 않고 바로 clipping을 해줍니다

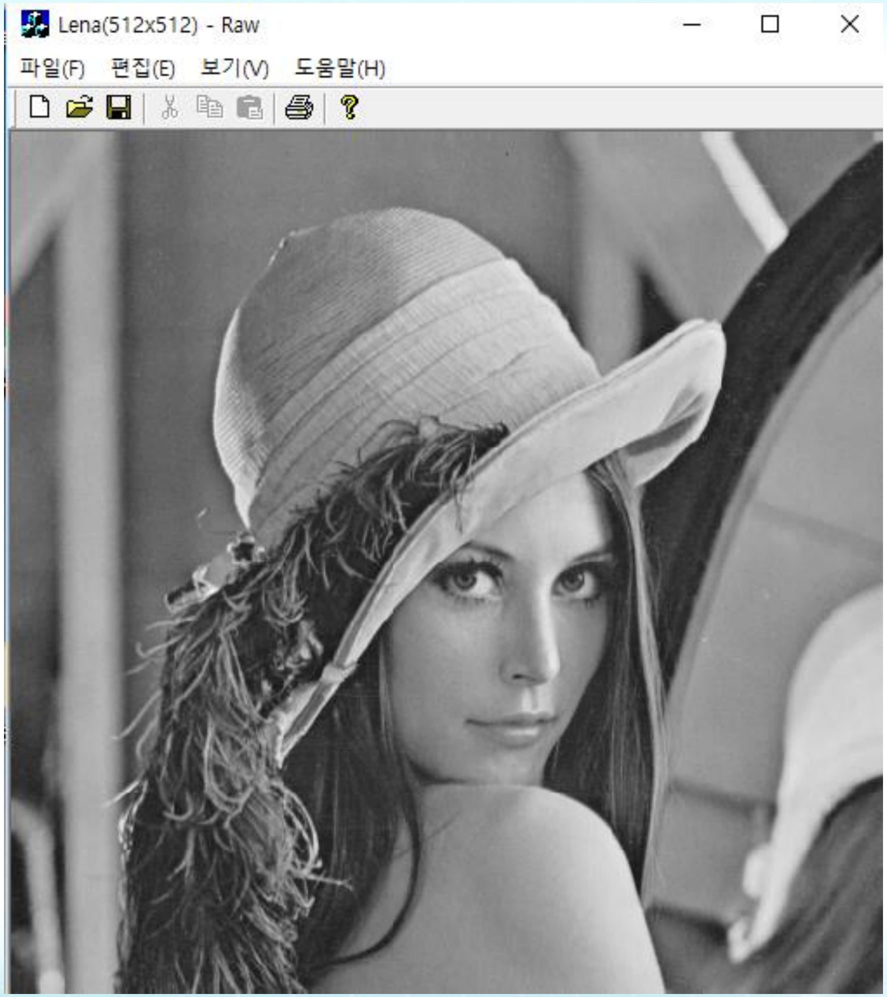
- 정리
  - Sobel , Laplacian, LoG
    - Convolution → + 128 → clipping
  - Sharpening
    - Convolution → clipping

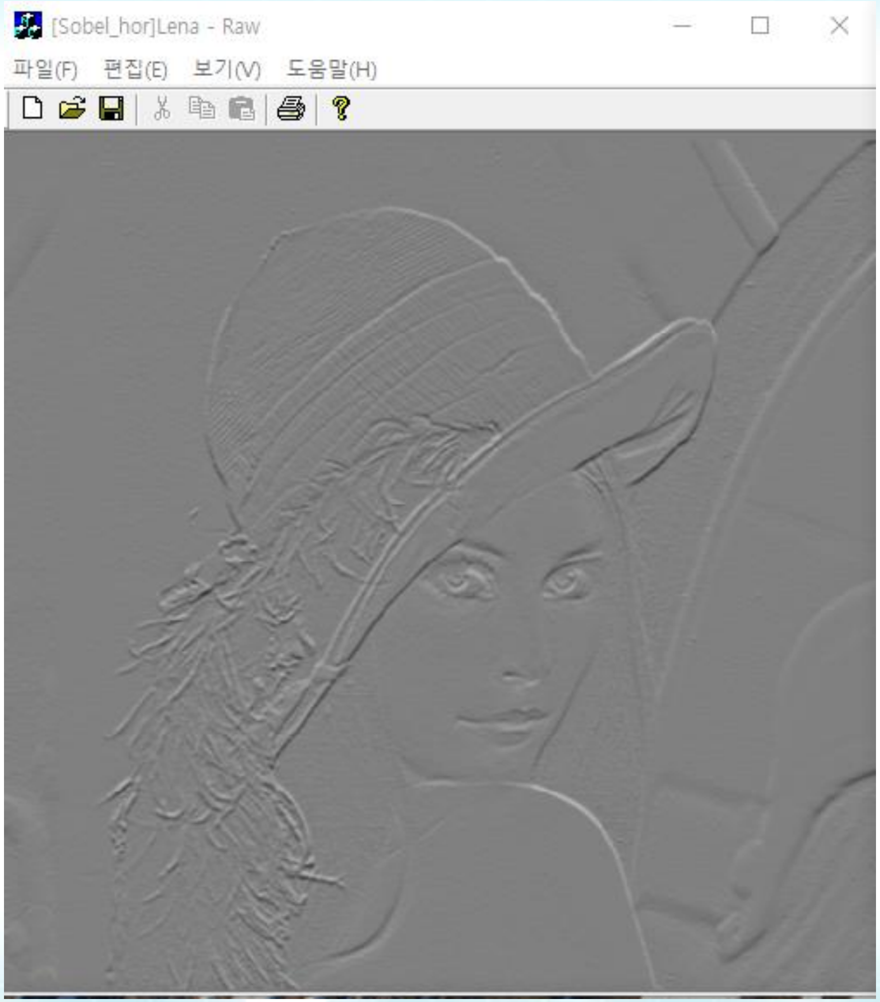
Assignment#2 Sobel mask vertical



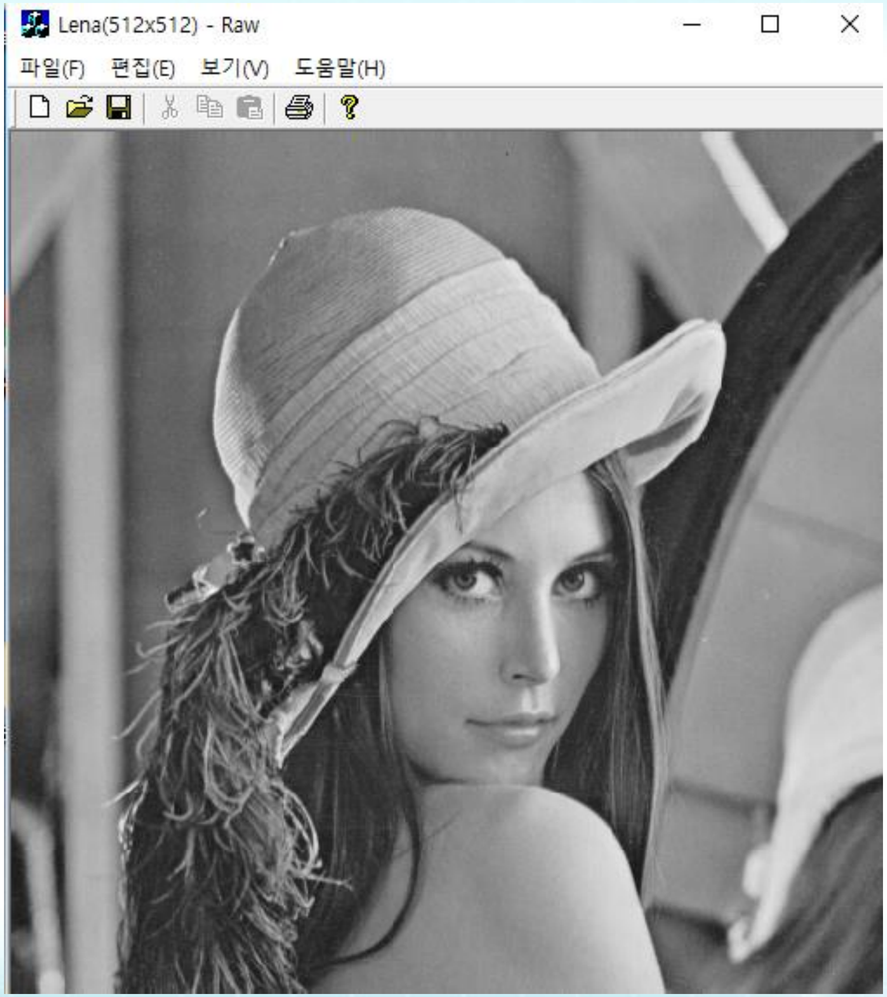


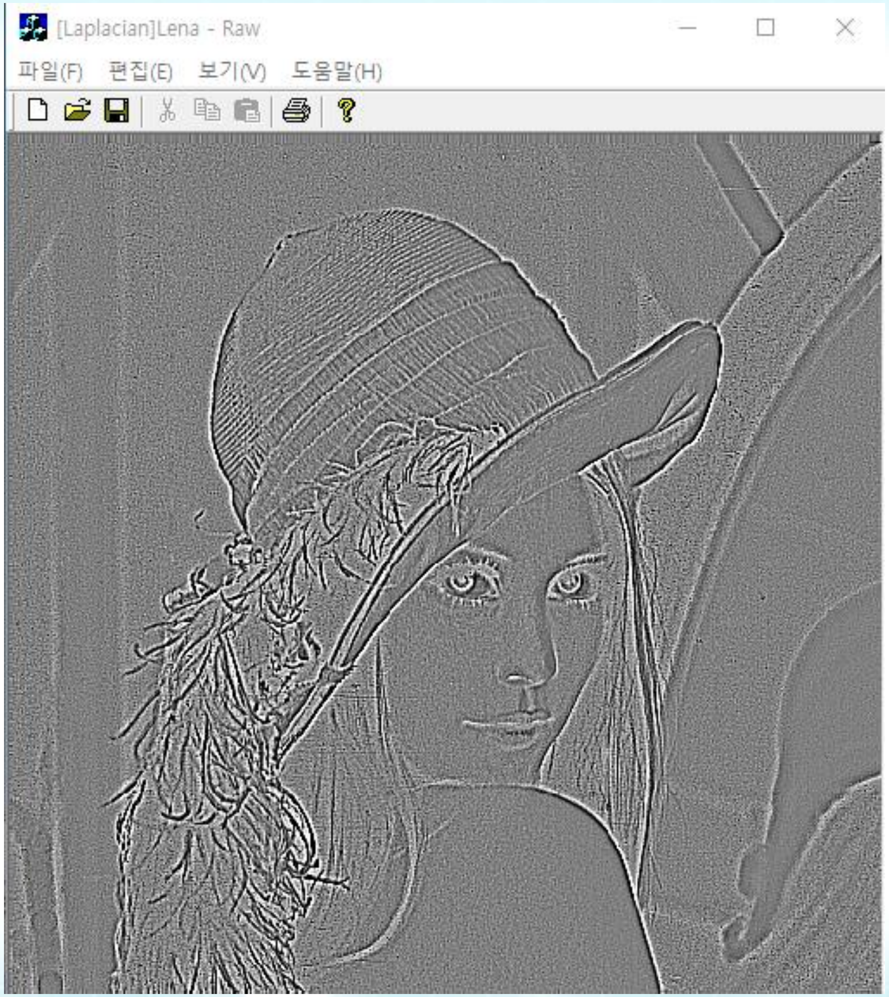
Assignment#2 Sobel mask horizontal



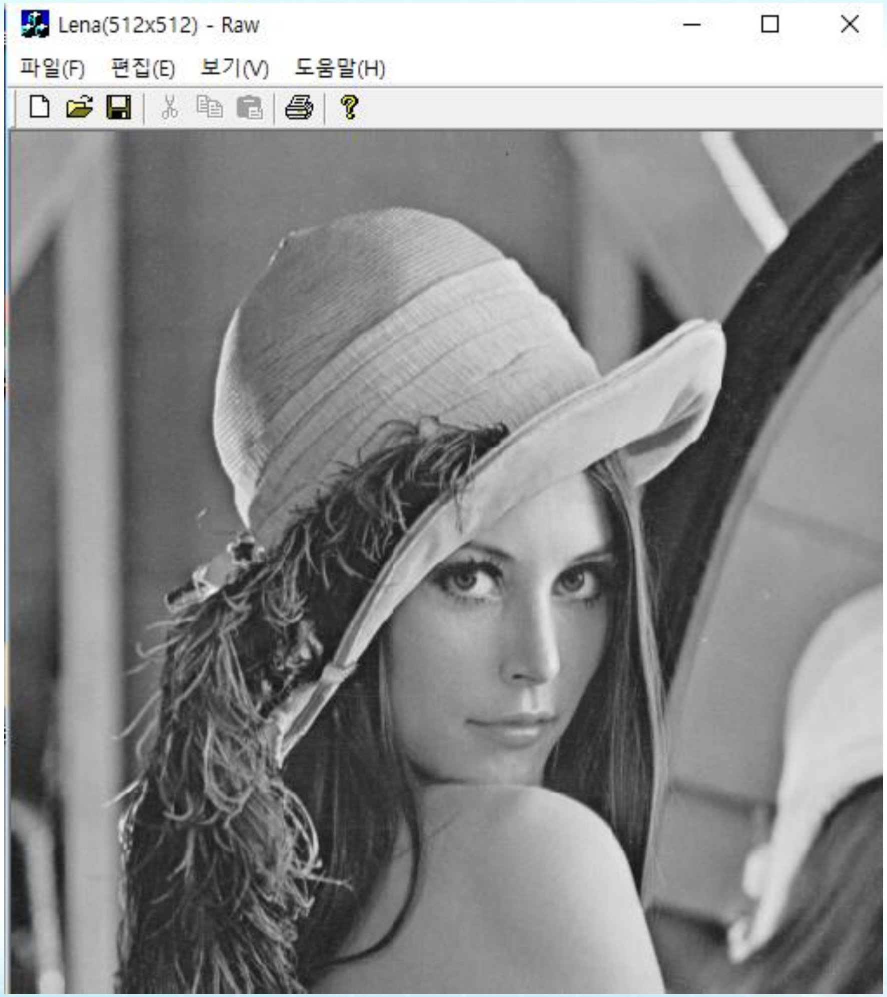


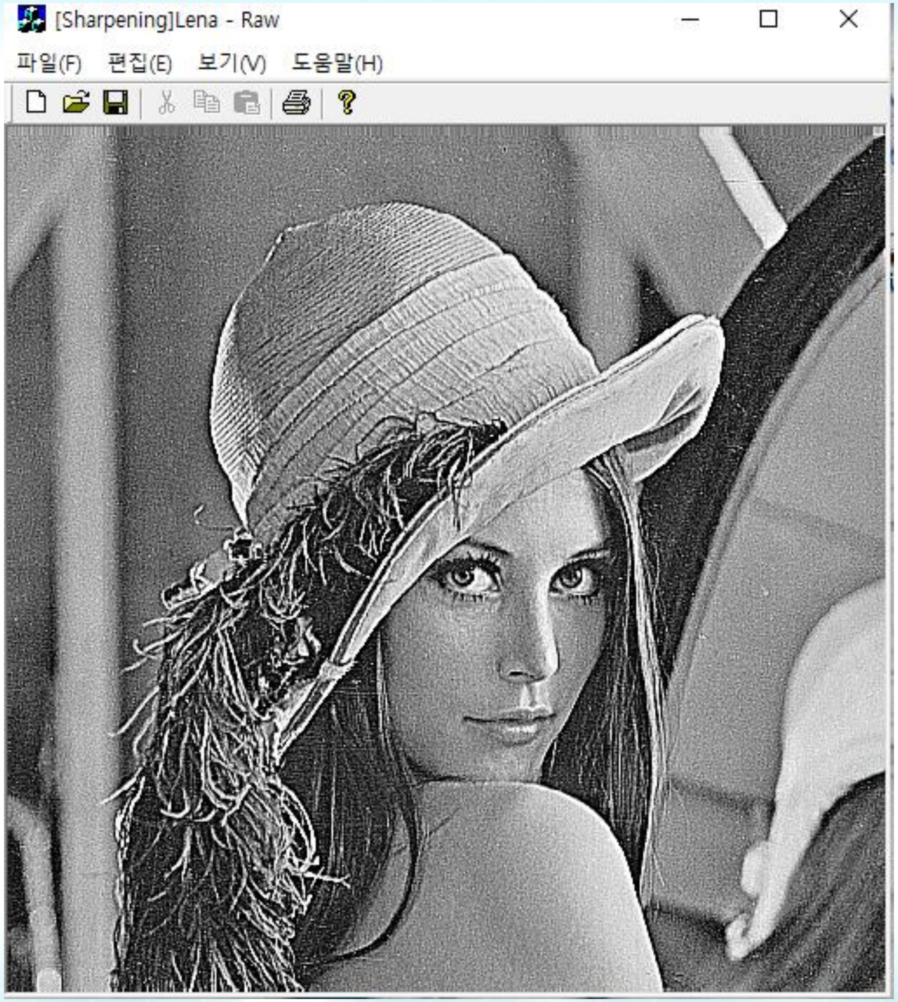
Assignment#3 Laplacian mask



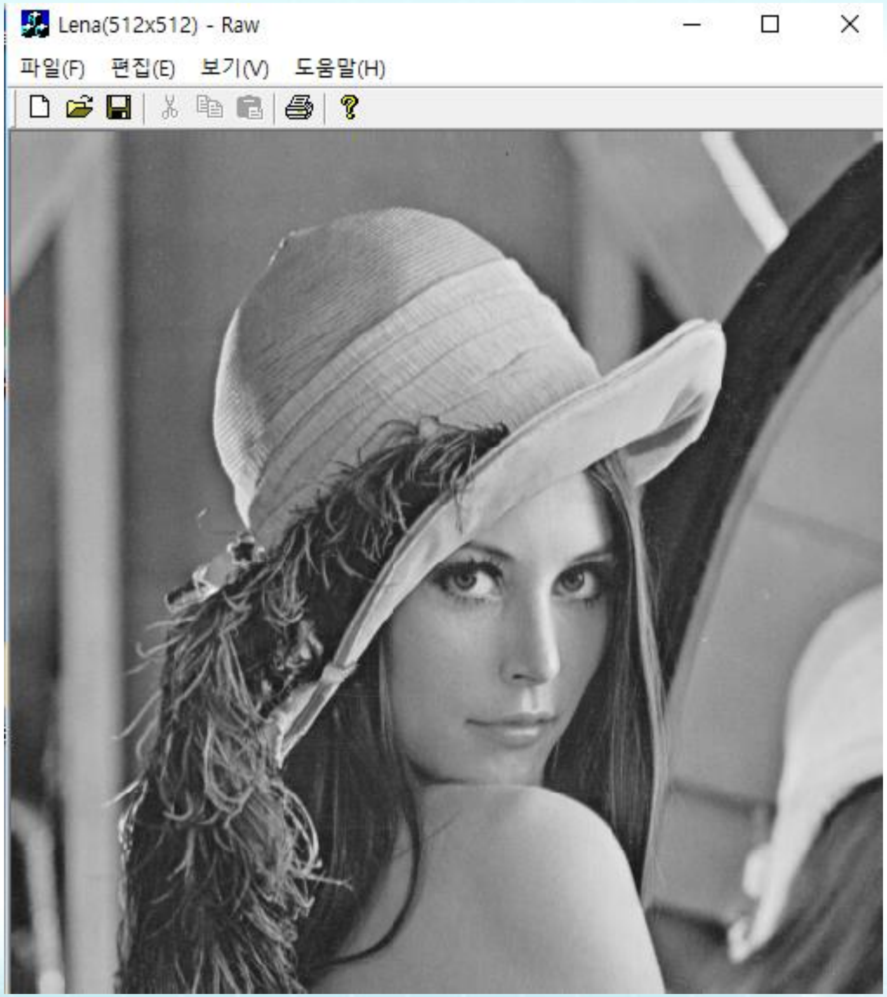


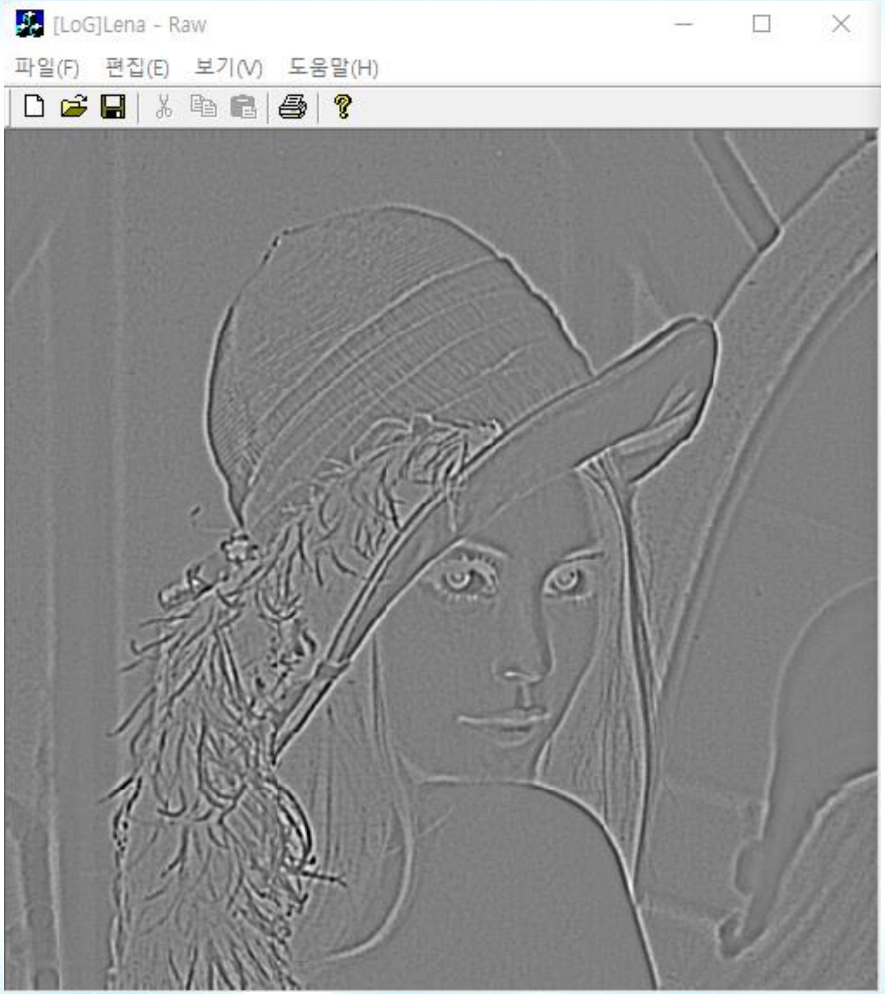
Assignment#3 sharpening mask





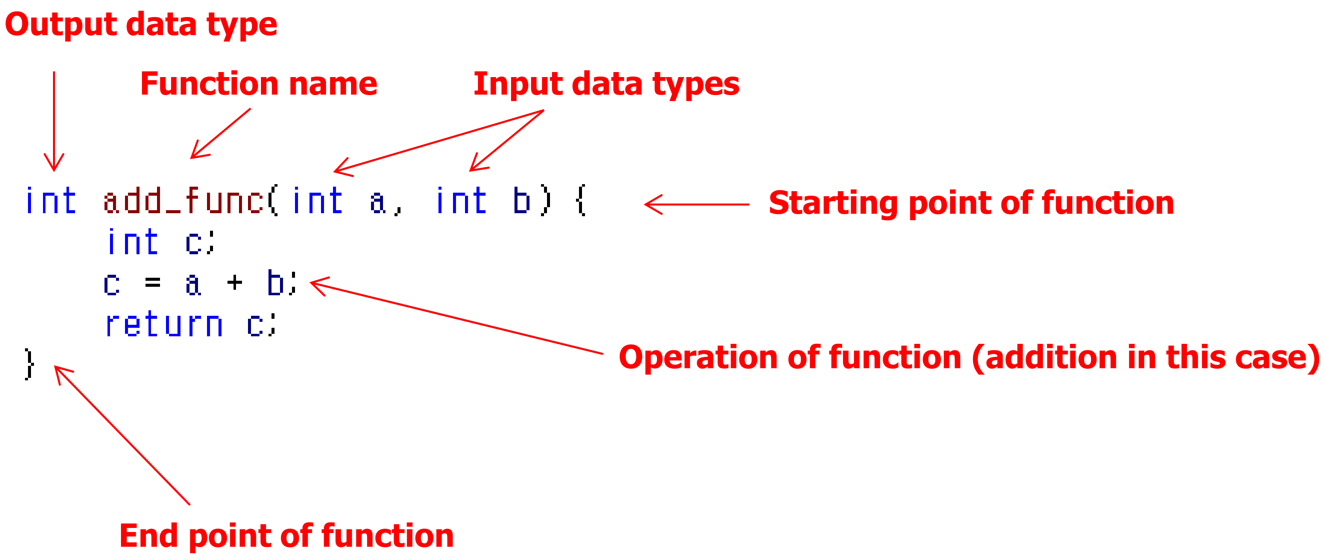
Assignment#4 LoG mask





P.S 함수 작성 요령

- 함수의 구성요소



```
int add_func( int a, int b) {
    int c;
    c = a + b;
    return c;
}
```

Output data type

Function name

Input data types

Starting point of function

Operation of function (addition in this case)

End point of function

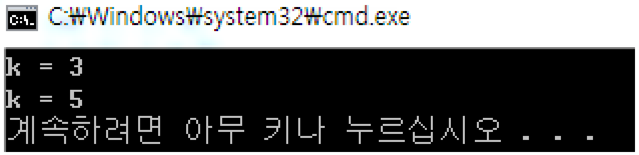
Function

❖ Example

```c
#include <stdio.h>

int add_func(int a, int b);

int main()
{
	int i, j, k;
	i = 1; j = 2;
	k = add_func(i,j);
	// k = 3;
	printf("k = %d₩n", k);
	k = add_func(2,3);
	// k = 5;
	printf("k = %d₩n", k);

	return 0;
}

int add_func(int a, int b) {
	int c;
	c = a + b;
	return c;
}
```

a = i　　b = j

```c
int add_func(int a, int b) {
	int c;
	c = a + b;
	return c;
}
```

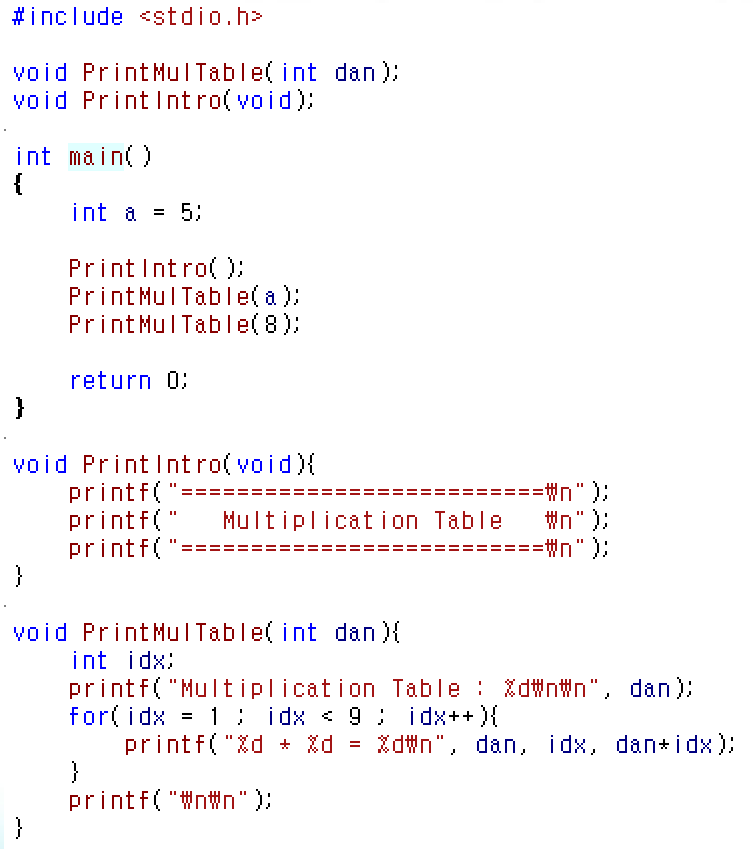


C:₩Windows₩system32₩cmd.exe

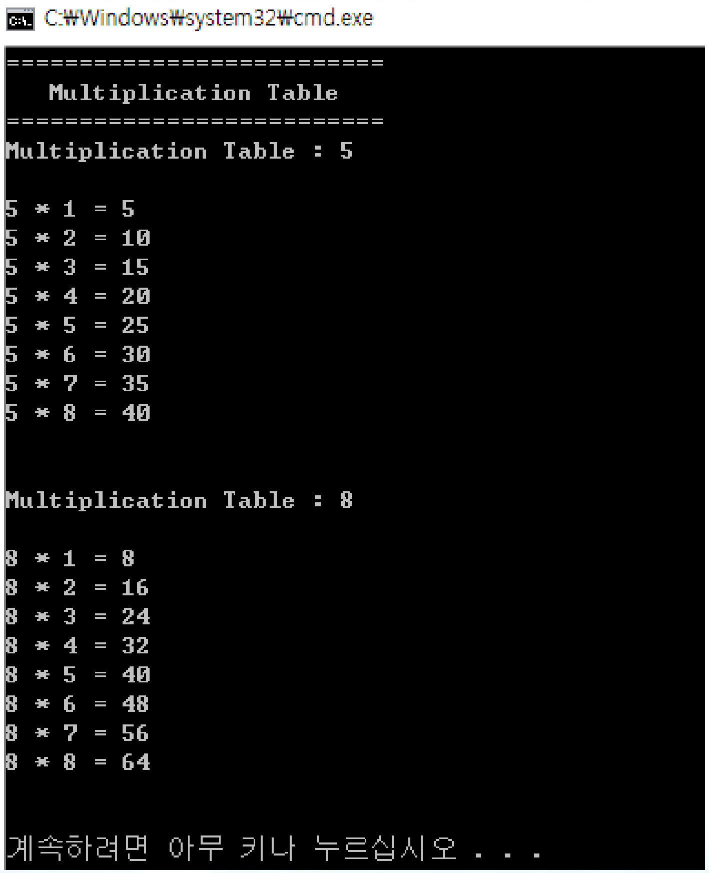
```
k = 3
k = 5
계속하려면 아무 키나 누르십시오 . . .
```

Function

- Example



```c
#include <stdio.h>

void PrintMulTable(int dan);
void PrintIntro(void);

int main()
{
	int a = 5;

	PrintIntro();
	PrintMulTable(a);
	PrintMulTable(8);

	return 0;
}

void PrintIntro(void){
	printf("==========================₩n");
	printf("   Multiplication Table   ₩n");
	printf("==========================₩n");
}

void PrintMulTable(int dan){
	int idx;
	printf("Multiplication Table : %d₩n₩n", dan);
	for(idx = 1 ; idx < 9 ; idx++){
		printf("%d * %d = %d₩n", dan, idx, dan*idx);
	}
	printf("₩n₩n");
}
```



C:₩Windows₩system32₩cmd.exe

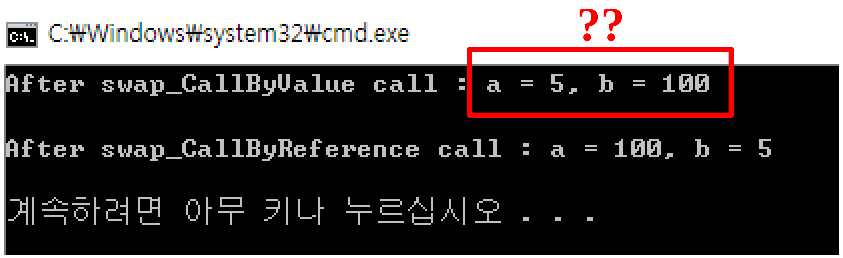
```
==========================
   Multiplication Table
==========================
Multiplication Table : 5

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40


Multiplication Table : 8

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64


계속하려면 아무 키나 누르십시오 . . .
```

Function

- Call by value vs. Call by reference

```c
#include <stdio.h>

void swap_CallByValue(int a, int b);
void swap_CallByReference(int *a, int *b);

int main()
{
    int a = 5;
    int b = 100;

    swap_CallByValue(a, b);
    printf("After swap_CallByValue call : a = %d, b = %d₩n₩n", a, b);

    swap_CallByReference(&a, &b);
    printf("After swap_CallByReference call : a = %d, b = %d₩n₩n", a, b);

    return 0;
}

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

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

??

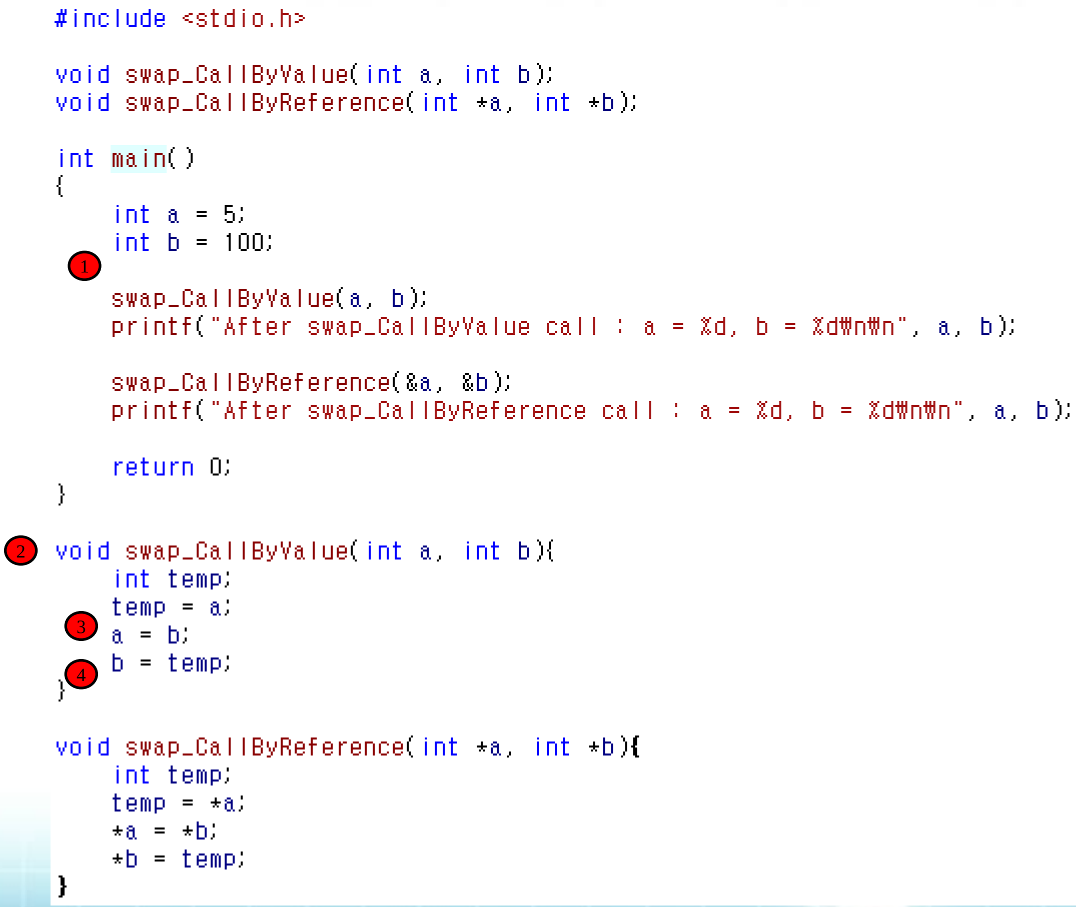


C:₩Windows₩system32₩cmd.exe

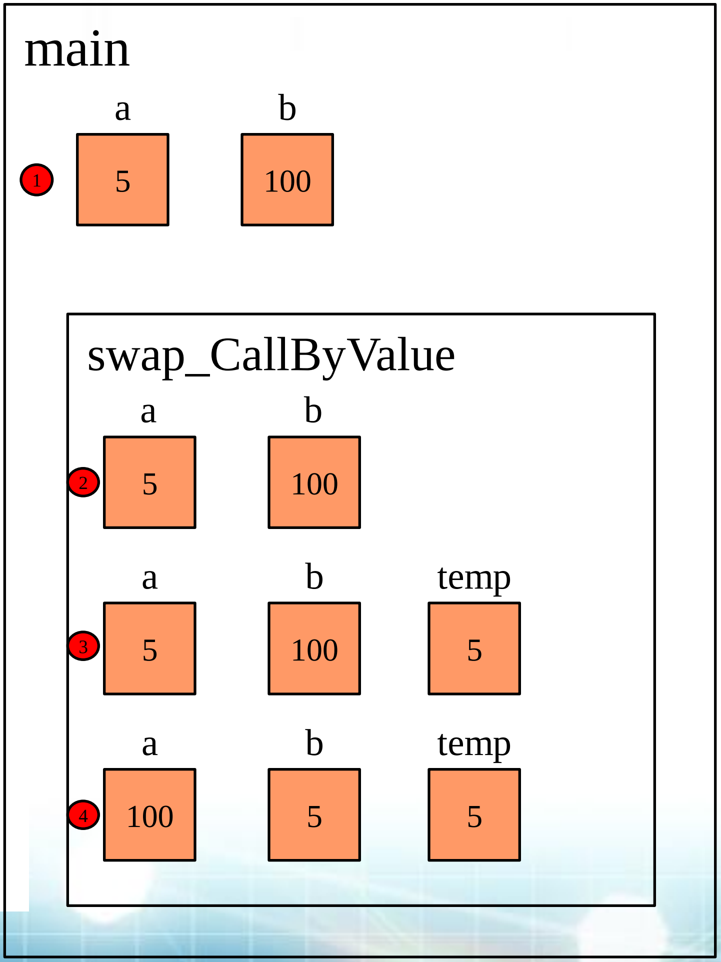
```
After swap_CallByValue call : a = 5, b = 100

After swap_CallByReference call : a = 100, b = 5

계속하려면 아무 키나 누르십시오 . . .
```

Function

- Call by value vs. Call by reference

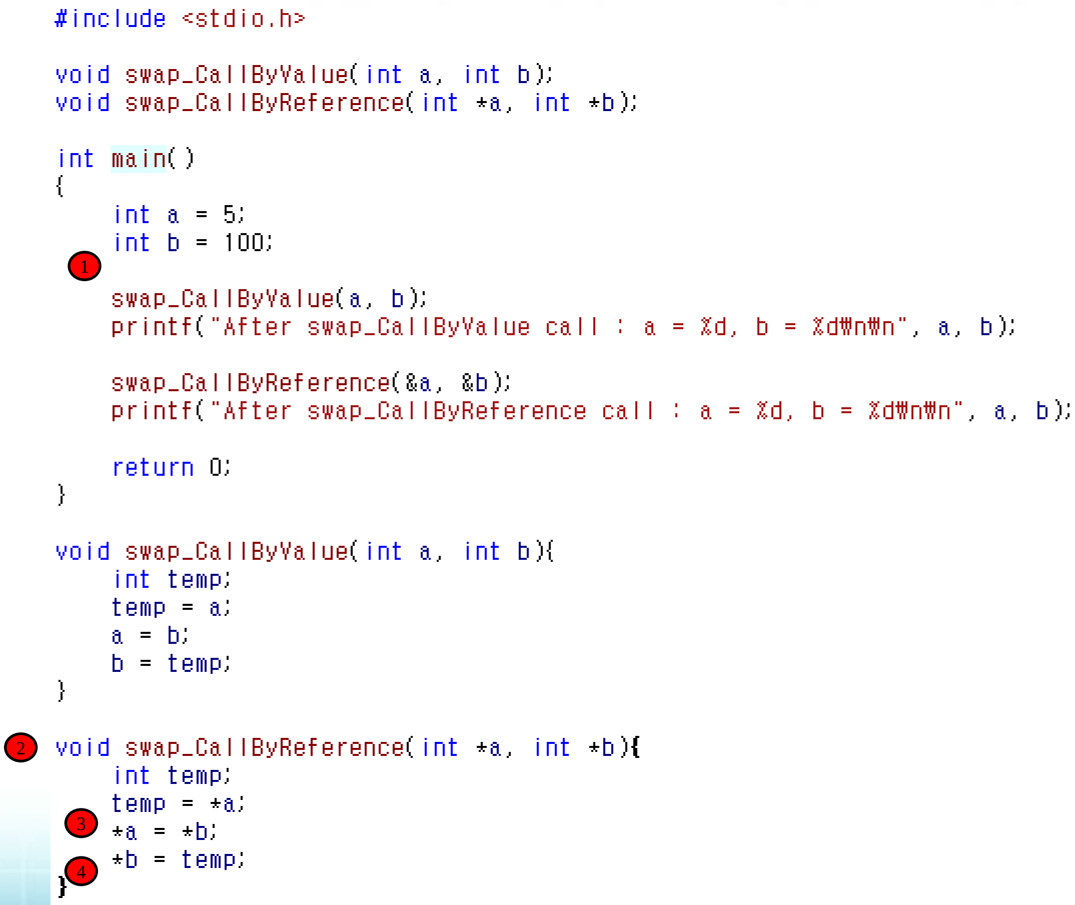


```
#include <stdio.h>

void swap_CallByValue(int a, int b);
void swap_CallByReference(int *a, int *b);

int main()
{
    int a = 5;
    int b = 100;

    swap_CallByValue(a, b);
    printf("After swap_CallByValue call : a = %d, b = %d₩n₩n", a, b);

    swap_CallByReference(&a, &b);
    printf("After swap_CallByReference call : a = %d, b = %d₩n₩n", a, b);

    return 0;
}

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

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



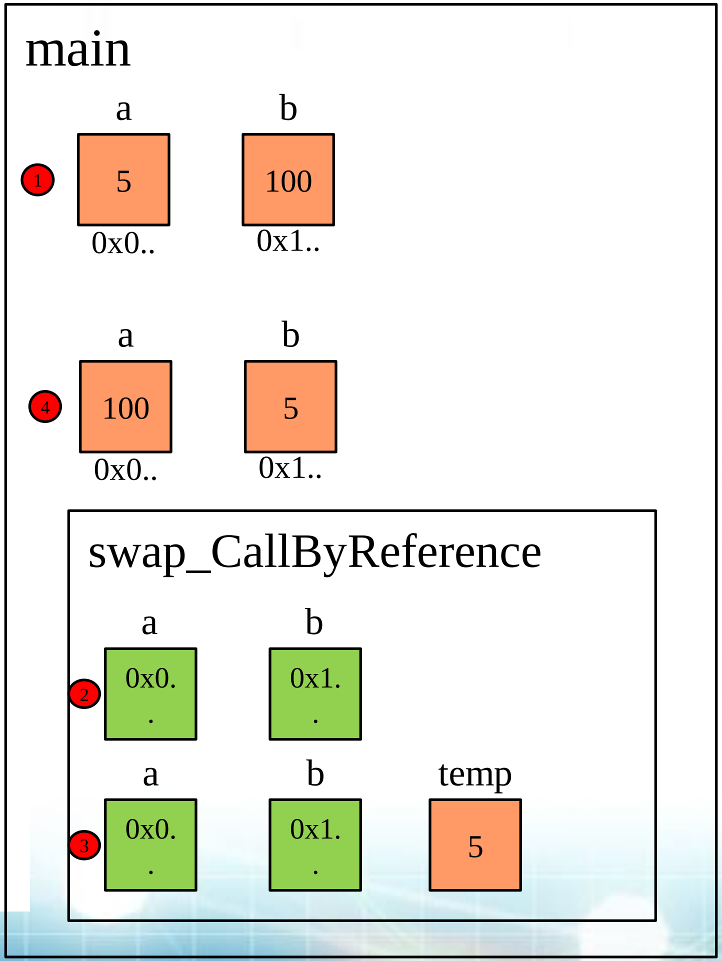
Function

❖Call by value vs. Call by reference



```
#include <stdio.h>

void swap_CallByValue(int a, int b);
void swap_CallByReference(int *a, int *b);

int main()
{
	int a = 5;
	int b = 100;

	swap_CallByValue(a, b);
	printf("After swap_CallByValue call : a = %d, b = %d₩n₩n", a, b);

	swap_CallByReference(&a, &b);
	printf("After swap_CallByReference call : a = %d, b = %d₩n₩n", a, b);

	return 0;
}

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

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

1
2
3
4



main
a
b
1
5
100
0x0..
0x1..
a
b
4
100
5
0x0..
0x1..
swap_CallByReference
a
b
2
0x0.
.
0x1.
.
a
b
temp
3
0x0.
.
0x1.
.
5