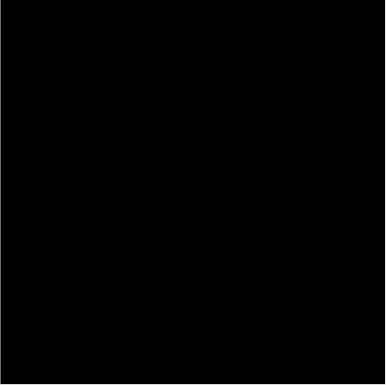
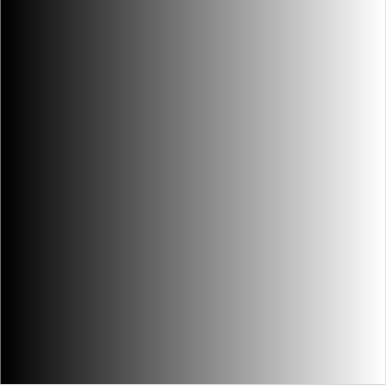
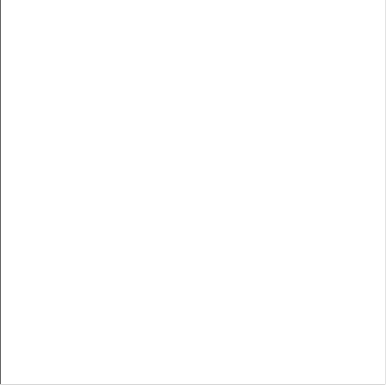
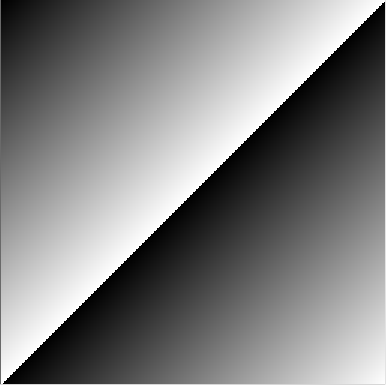
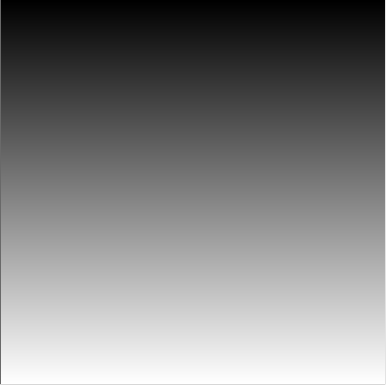
**영상처리 실습 #2**

16010980 이우석

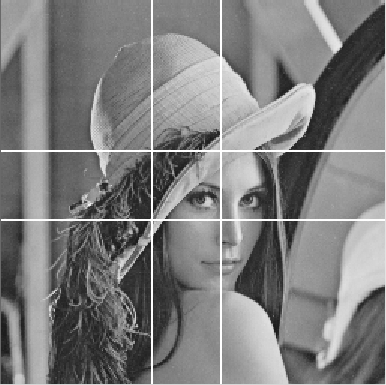
1. **실습 A**



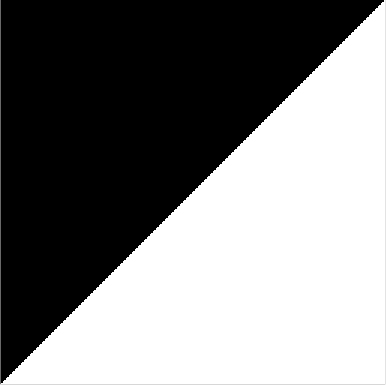


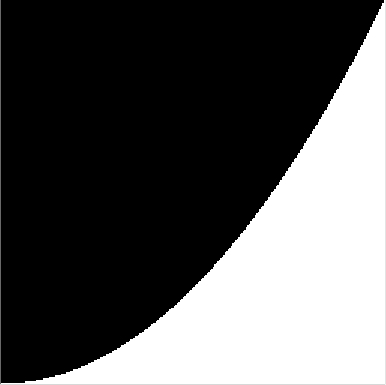


1. **실습 B**

****

1. **실습 C**





1. **< 코드 내용 >**

/\* initialize Look-up table \*/

for (i = 0; i < 256; i++)

{

/\* preserve an original image or y is equal to x \*/

// temp = i;

/\* operation \*/

// temp = operation(i); // temp 임시 변수에 i \* 1.9 값을 저장. temp 는 정수형이기 때문에 소수점 이하의 값을 버림.

/\* plus \*/

// temp = plus(i); // + 50

/\* multiply \*/

// temp = multiply(i); // \* 1.5

/\* complex operation \*/

// temp = complexOper(i); // \* 1.5 + 40

/\* convert 0 \*/

// temp = convert\_0(i); // 120 < i < 128 => temp

/\* gamma \*/

// temp = gamma(i); // gamma 2

CLIP(temp, 0, 255); // 맵핑된 후의 밝기 레벨(L) 값이 0 보다 작거나, 255 보다 크면 안되기 때문에.

LUT[i] = temp; // LUT[i] 에 temp 값을 초기화.

}

/\* process image via the Look-up table \*/

for (i = 0; i < number\_of\_pixels; i++) { // 모든 픽셀에 대해 Look-up Table 을 이용하여 밝기 레벨(L)을 새롭게 맵핑.

/\* 실습 A - make 0 \*/

// buffer[i] = 0;

/\* 실습 A - make 255 \*/

// buffer[i] = 255;

/\* 실습 A - cols \*/

// buffer[i] = i % cols;

/\* 실습 A - rows \*/

// buffer[i] = i / cols;

/\* 실습 A - rows + cols \*/

// buffer[i] = ((i / cols) + (i % cols)) % cols;

/\* 실습 B \*/

/\*

if (i % cols == 100 || i % cols == 146 || i / cols == 100 || i / cols == 146)

buffer[i] = 255;

\*/

/\* 실습 C \*/

/\*

if ((255 - i / cols) > LUT[i % cols])

buffer[i] = 0;

else

buffer[i] = 255;

\*/

// buffer[i] = LUT[buffer[i]];