

# IP Programming 3

---

Jongseok Lee([suk2080@kw.ac.kr](mailto:suk2080@kw.ac.kr))

Yong-Jo Ahn ([yjahn@digitalinsights.co.kr](mailto:yjahn@digitalinsights.co.kr))

2018-03-21

# Contents

---

- 3.1 실습 목적
- 3.2 실습 흐름도
- 3.3 실습 따라 하기
- 3.4 실습 과제

# **[실습 3.1] 실습 목적**

---

## 3.1 실습 목적

---

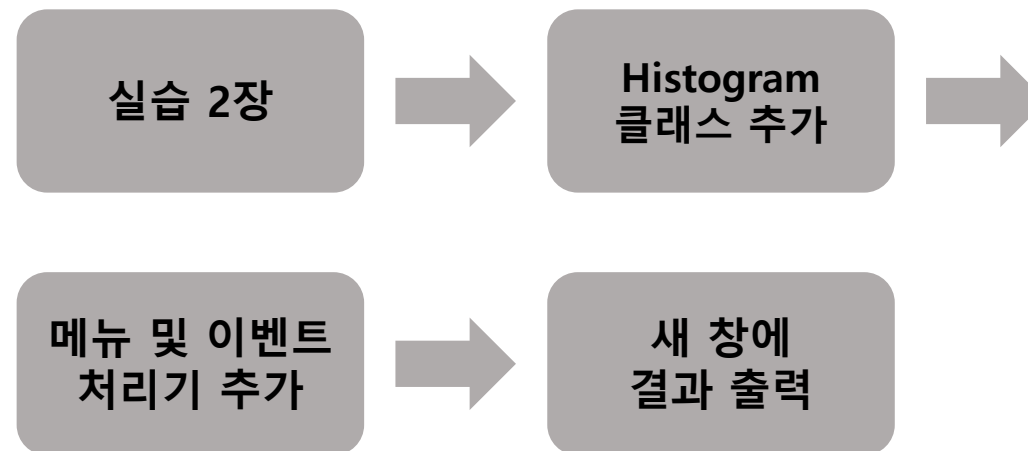
- 이번 장에서는 이미지 밝기 분포를 살펴보고 이를 균일하도록 수정하는 실습을 진행한다.
- 이 과정을 통해 영상의 특성을 파악하고 해당 특성에 따라 영상을 어떻게 개선하면 좋을지에 대해 생각해 보자.

# **[실습 3.2] 실습 흐름도**

---

## 3.2 실습 흐름도

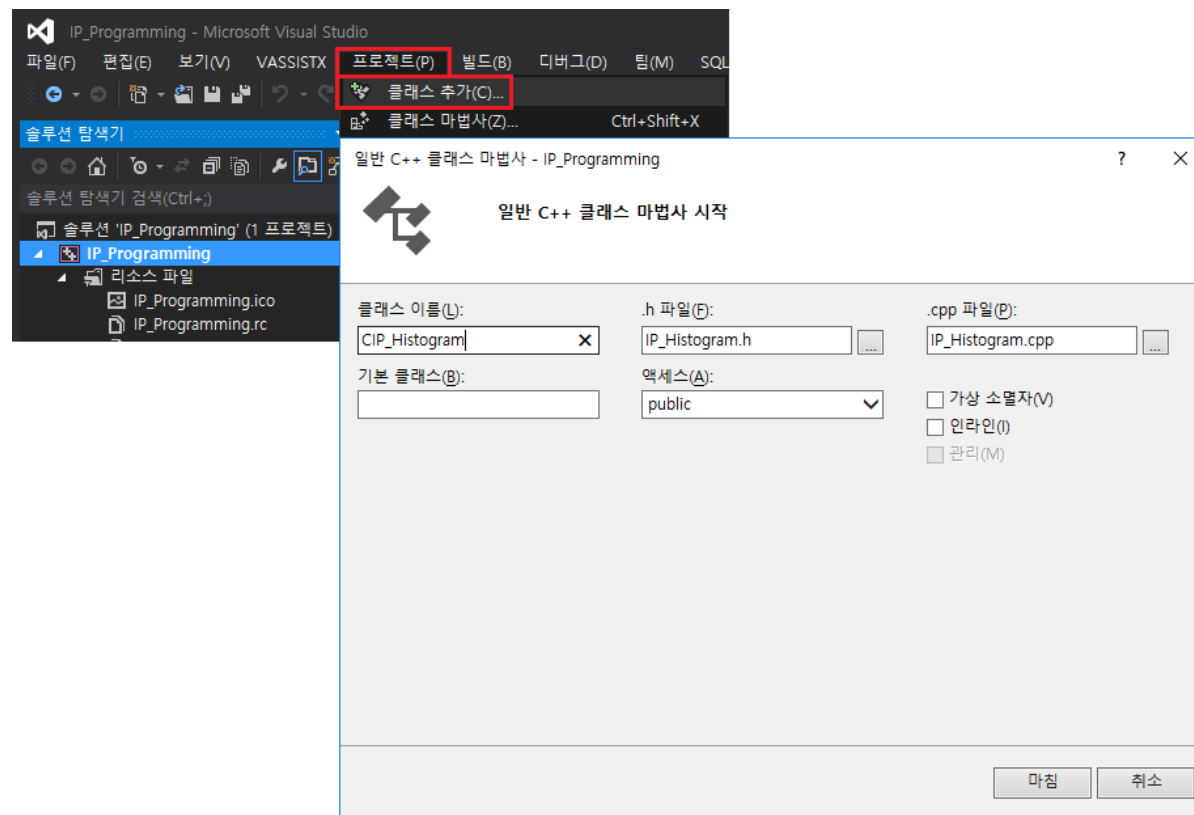
---



# [실습 3-3] 실습 따라 하기

## 3.3.1 Histogram 클래스 추가

- 단계 1 : Histogram 클래스 추가
  - [프로젝트] - [클래스 추가]





## 3.3.1 Histogram 클래스 추가

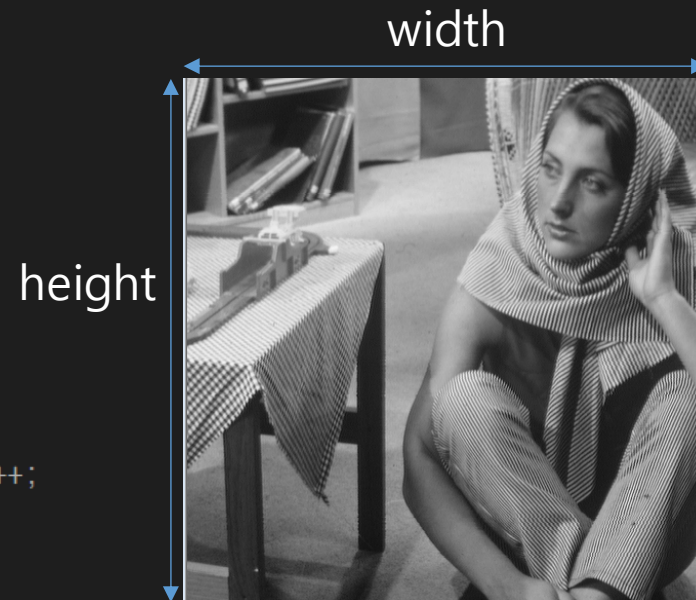
- 단계 2 : Histogram 클래스에 멤버변수와 멤버함수 추가

```
IP_Histogram.h  ➤ ✕
(전역 범위)
1  #pragma once
2  class CIP_Histogram
3  {
4  public:
5      UCHAR** m_pucHistEqualImgBuf; //히스토그램 평활화 이미지 저장 버퍼
6
7  public:
8      CIP_Histogram(void);
9      ~CIP_Histogram(void);
10
11     void Histogram_MakeHistogram(UCHAR** HistImgbuf, float HistArray[256], int width, int height); //이미지의 히스토그램 계산
12     void Histogram_MakeHistogramEqualization(UCHAR** imgbuf, int width, int height); //히스토그램 평활화
13     UCHAR** memory_alloc2D(int width, int height); //2D 메모리 할당
14 };
```

## 3.3.1 Histogram 클래스 추가

- 단계 3 : Histogram 클래스 멤버함수 작성
  - Histogram\_MakeHistogram

```
void CIP_ProgrammingMyClass::MyClass_MakeHistogram(UCHAR** HistImgbuf, float HistArray[256])  
{  
    int i,j;  
    int width = m_uiWidth;  
    int height = m_uiHeight;  
  
    //8-bit depth 임시 배열 생성  
    int Image_Histogram[256] = {0,};  
  
    //Histogram Calculation  
    for(i=0; i<height; i++)  
    {  
        for(j=0; j<width; j++)  
        {  
            Image_Histogram[HistImgbuf[i][j]]++;  
        }  
    }  
  
    //Histogram Normalization  
    float Image_Area = (float)(height*width);  
    for(i=0; i<256; i++)  
    {  
        HistArray[i] = Image_Histogram[i]/Image_Area;  
    }  
}
```



## 3.3.1 Histogram 클래스 추가

- 단계 3 : Histogram 클래스 멤버함수 작성
  - MakeHistogramEqualization

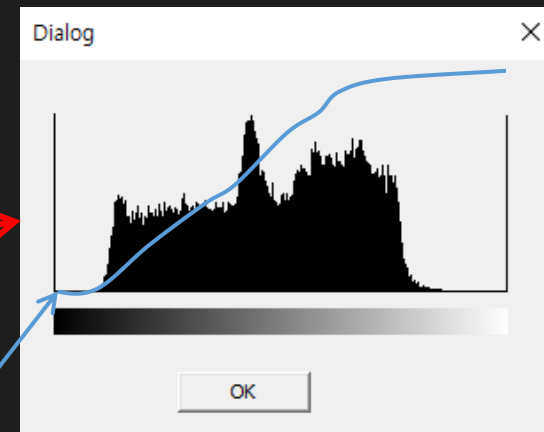
```
void CIP_ProgrammingMyClass::MyClass_MakeHistogramEqualization ()
```

```
{  
    int i,j;  
    int width = m_uiWidth;  
    int height = m_uiHeight;  
  
    //원본 영상의 Histogram 생성  
    float Image_Histogram_Equalization[256] = {0.};  
    MyClass_MakeHistogram(m_pucImgbuf, Image_Histogram_Equalization);
```

```
    //Histogram 누적 합 계산  
    double CDF[256] = {0.0, };  
    CDF[0] = Image_Histogram_Equalization[0];  
    for(i=1; i<256; i++)  
    {  
        CDF[i] = CDF[i-1] + Image_Histogram_Equalization[i];  
    }
```

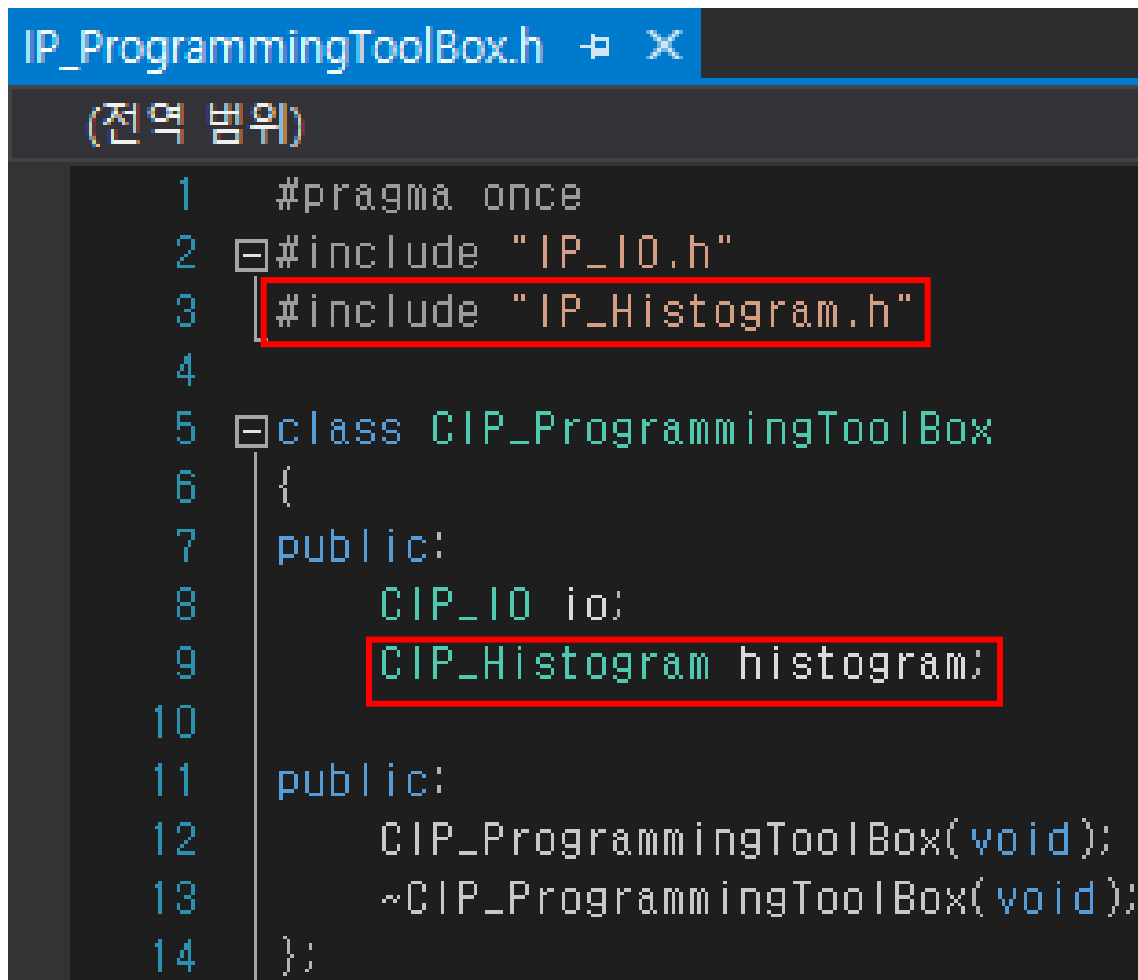
```
    //Histogram equalization 영상 메모리 할당  
    m_pucHistEqualImgBuf = memory_alloc2D(height,width);
```

```
    //Histogram equalization image 생성  
    for(i=0; i<height; i++)  
    {  
        for(j=0; j<width; j++)  
        {  
            m_pucHistEqualImgBuf[i][j] =  
                (UCHAR) (CDF[m_pucImgbuf[i][j]]*255);  
        }  
    }  
}
```



## 3.3.1 Histogram 클래스 추가

- 단계 4 : CIP\_ProgrammingToolBox 클래스에 멤버변수 추가

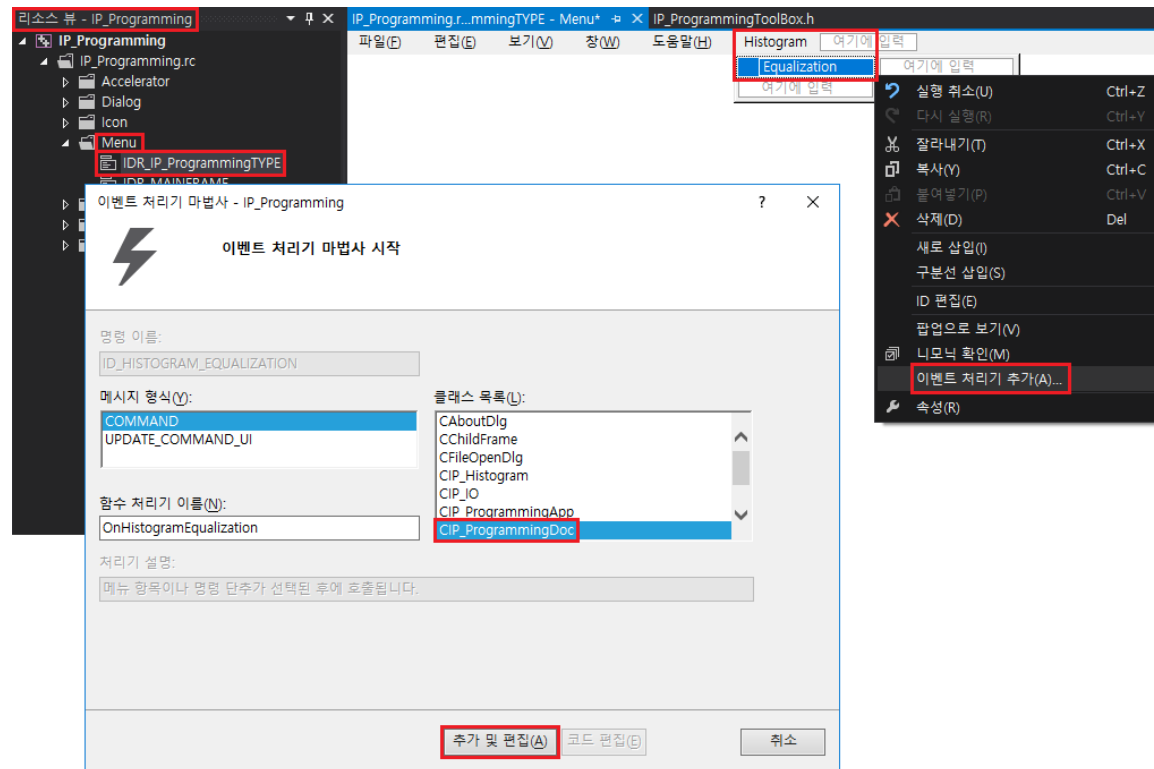


The screenshot shows a code editor window titled "IP\_ProgrammingToolBox.h". The code is in C++ and defines the CIP\_ProgrammingToolBox class. The following lines are highlighted with red boxes:

```
(전역 범위)
1  #pragma once
2  #include "IP_IO.h"
3  #include "IP_Histogram.h"
4
5  class CIP_ProgrammingToolBox
6  {
7  public:
8      CIP_IO io;
9      CIP_Histogram histogram;
10
11  public:
12      CIP_ProgrammingToolBox(void);
13      ~CIP_ProgrammingToolBox(void);
14  };
```

## 3.3.2 메뉴 및 이벤트 처리기 추가

- 단계 5 : Histogram 함수 메뉴 및 이벤트 처리기 추가



- 단계 6 : 이벤트 처리기 함수 작성  
– OnHistogramEqualization

### 3.3.3 새 창에 결과 출력

- 단계 7 : CIP\_Programming 클래스 멤버변수 추가

```

13 // CIP_ProgrammingApp:
14 // 이 클래스의 구현에 대해서는 IP_Programming.cpp를 참조하십시오.
15 //
16
17 class CIP_ProgrammingToolBox;
18 class CIP_ProgrammingApp : public CWinApp
19 {
20 public:
21     CIP_ProgrammingToolBox *toolbox;
22
23 public:
24     CIP_ProgrammingApp();

```

- 단계 8 : CIP\_ProgrammingDoc 멤버함수 수정

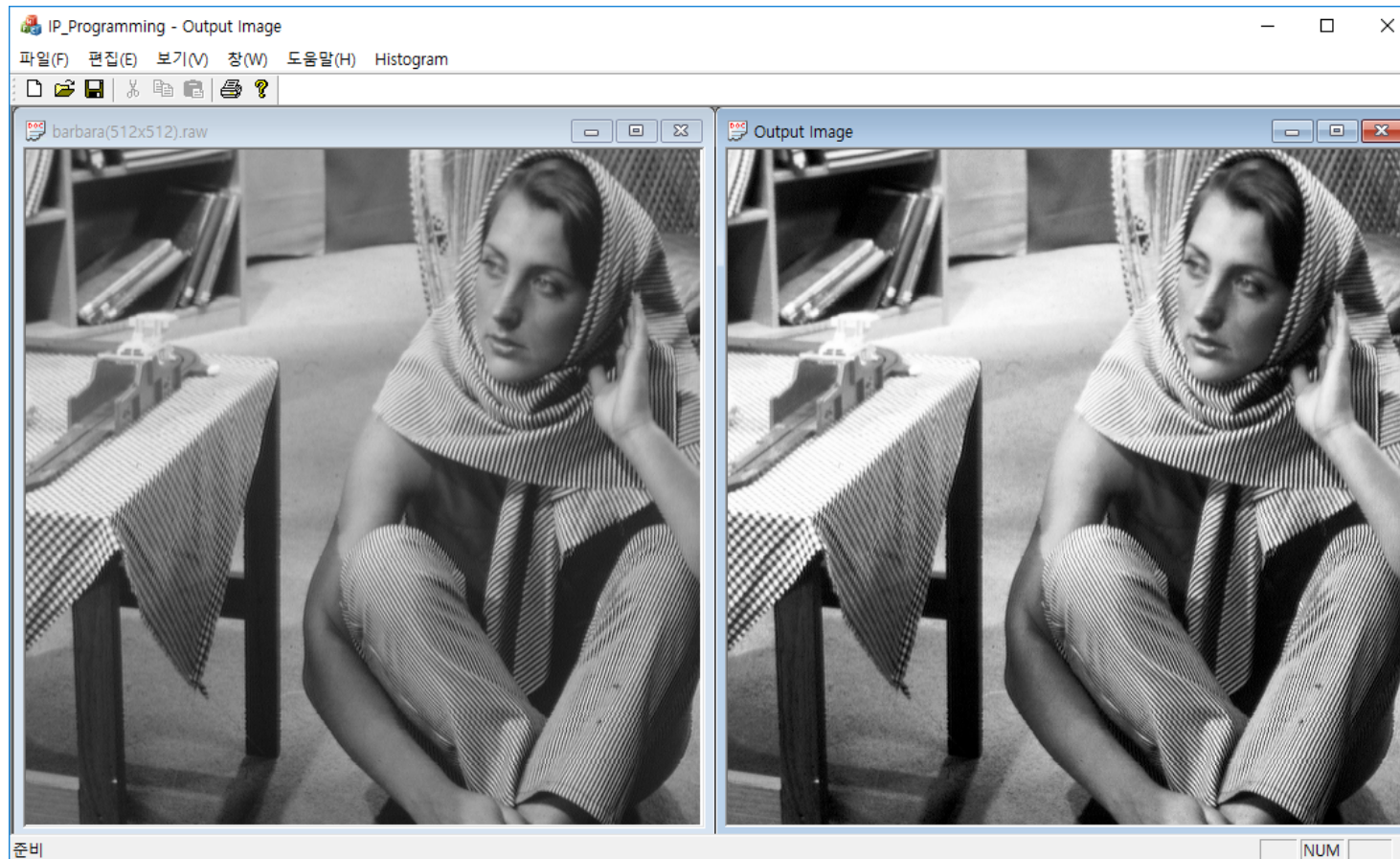
```

IP_ProgrammingDoc.cpp -> X
(전역 범위)
42 BOOL CIP_ProgrammingDoc::OnNewDocument()
43 {
44     if (!CDocument::OnNewDocument())
45         return FALSE;
46
47     // TODO: 여기에 재초기화 코드를 추가합니다.
48     // SDI 문서는 이 문서를 다시 사용합니다.
49     CIP_ProgrammingApp *pApp = (CIP_ProgrammingApp*)AfxGetApp();
50     if (pApp->toolbox != NULL){
51         if (pApp->toolbox->io.m_Outputbuf){
52             toolbox.io.m_Height = pApp->toolbox->io.m_Height;
53             toolbox.io.m_Width = pApp->toolbox->io.m_Width;
54             toolbox.io.IO_MakeGrayImageToBMP(pApp->toolbox->io.m_Outputbuf);
55
56             this->SetTitle("Output Image");
57             pApp->toolbox = NULL;
58         }
59     }
60
61     return TRUE;
62 }

```

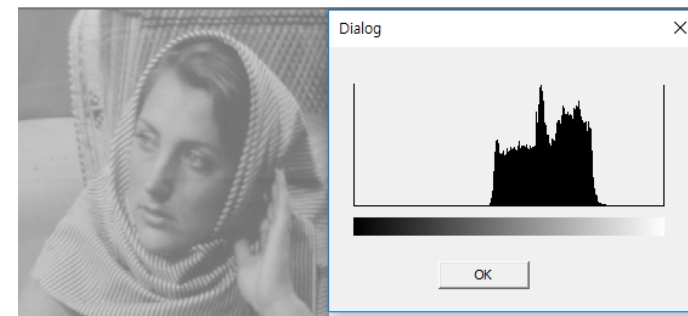
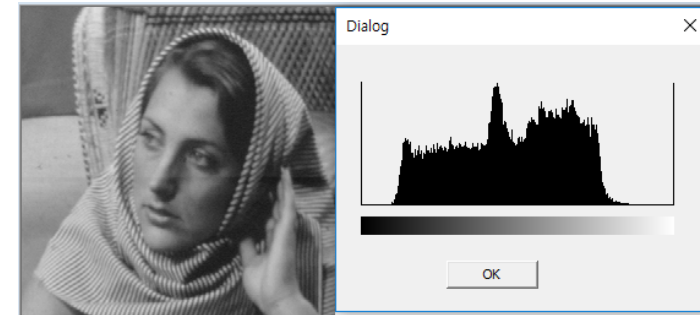
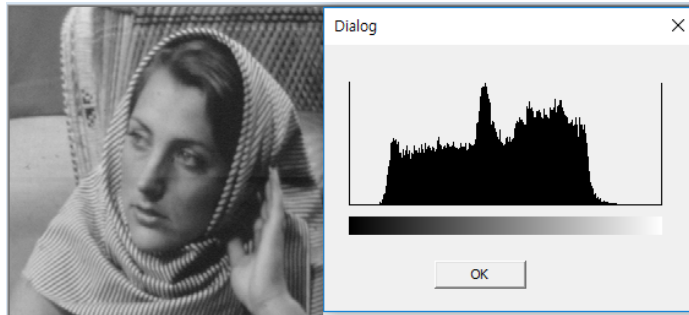
## 3.3.4 최종 출력 결과

### ■ 최종 출력 결과



### 3.3.4 최종 출력 결과

- 입력 받은 영상의 히스토그램의 분포를 좁거나 넓게 변환하여 영상의 대비 조절 가능

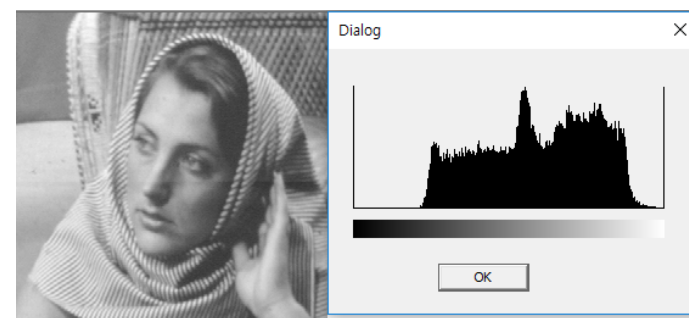
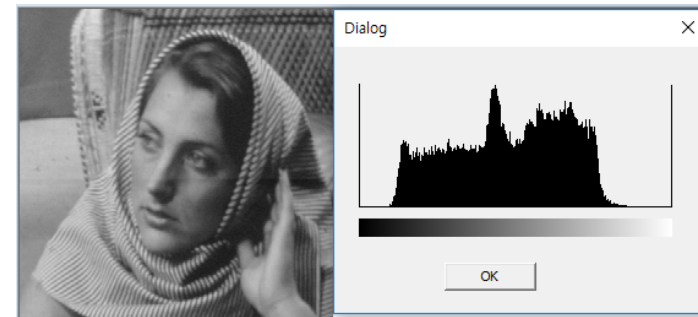
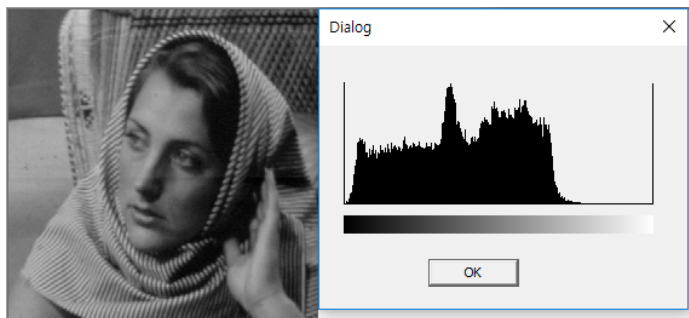
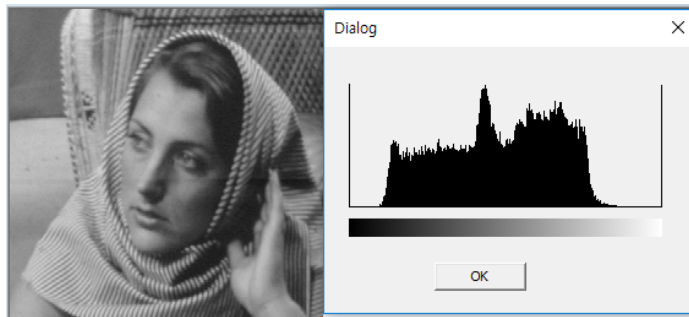




# [실습 3-4] 실습 프로젝트

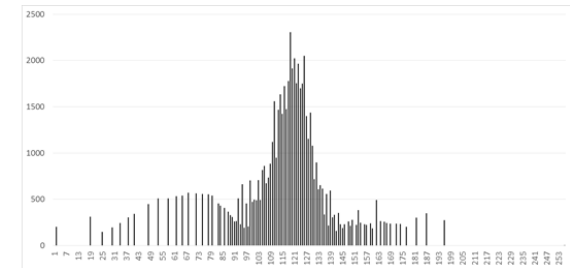
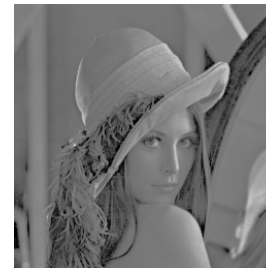
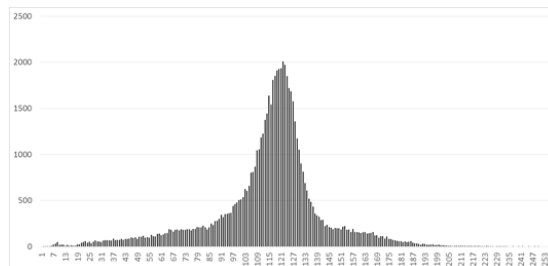
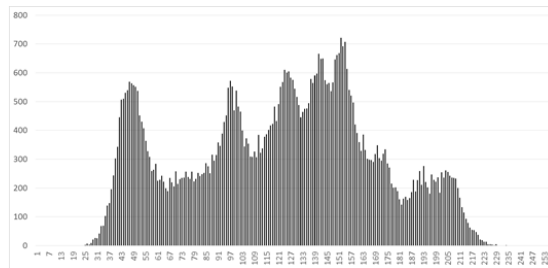
## 3.4.2 실습 과제 1

- [실습 과제 1] 영상 밝기 변환
  - 입력 받은 영상의 히스토그램을 왼쪽 혹은 오른쪽으로 shift시켜 영상의 밝기를 변환하도록 프로그램을 수정



## 3.4.3 실습 과제 2

- [실습 과제 1] 히스토그램 매칭
  - 영상을 두개 입력 받아, 첫 번째 영상의 히스토그램의 분포를 두 번째 영상의 히스토그램의 분포와 비슷한 모양을 갖도록 프로그램 수정



# END OF PRESENTATION

---

Q&A