DIP(Digital Image Processing)
Program Exercise

김남욱, 김명준, 정지연, 김양우, 이영렬

Sejong University, DMS Lab.

Program Exercise 1

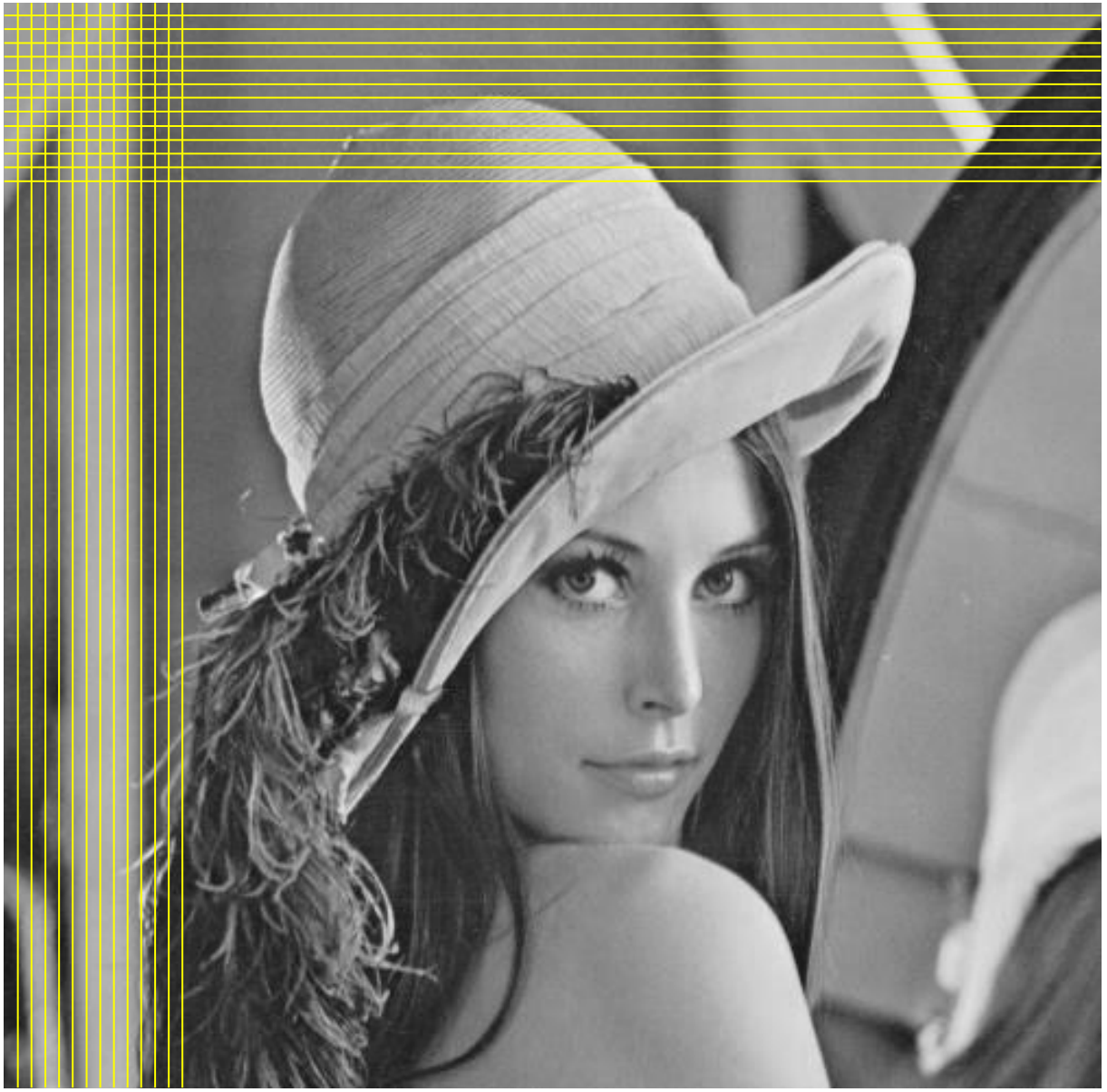
1. Zoom-out
2. Zoom-in
3. Low-pass filtering (average filtering)
4. High-pass filtering (Sobel operator)
5. Zoom-out and translation
6. Zoom-out and translation and rotation
7. histogram equalization(or Histogram slide-mapping & histogram stretch-mapping)

- 총 10점 만점
  - 1~4번 각 1점
  - 5~7번 각 2점

- 채점 및 검사
  - 장소 : 실습시간 중 또는 대양AI센터 817호
  - 조교 : 김남욱, 김명준, 정지연, 김양우

실습이미지 - **lena.img**

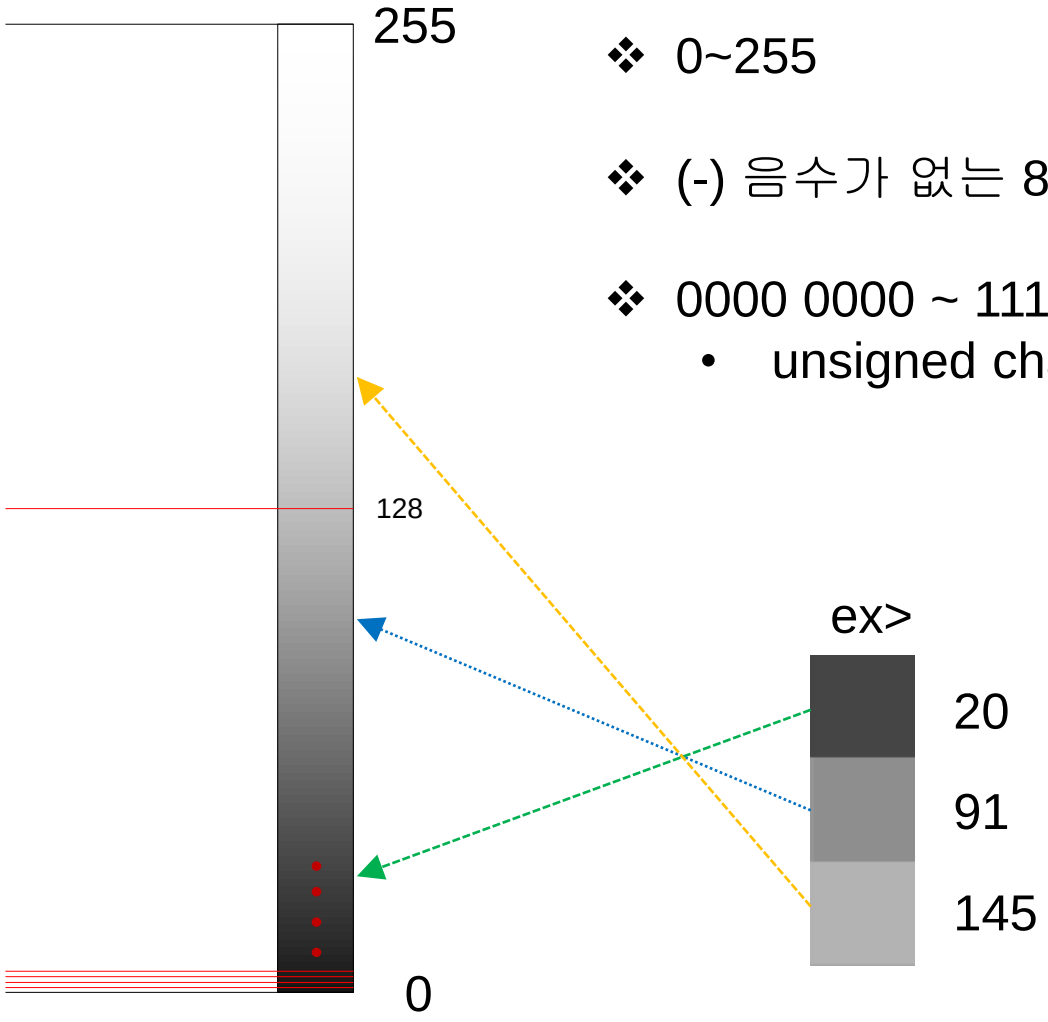
- 512 x 512 (pixel by pixel size)
- 밝기 값만 가지고 있는 이미지
- 8 bits range



512x512 = 26112개 화소
26112 Bytes

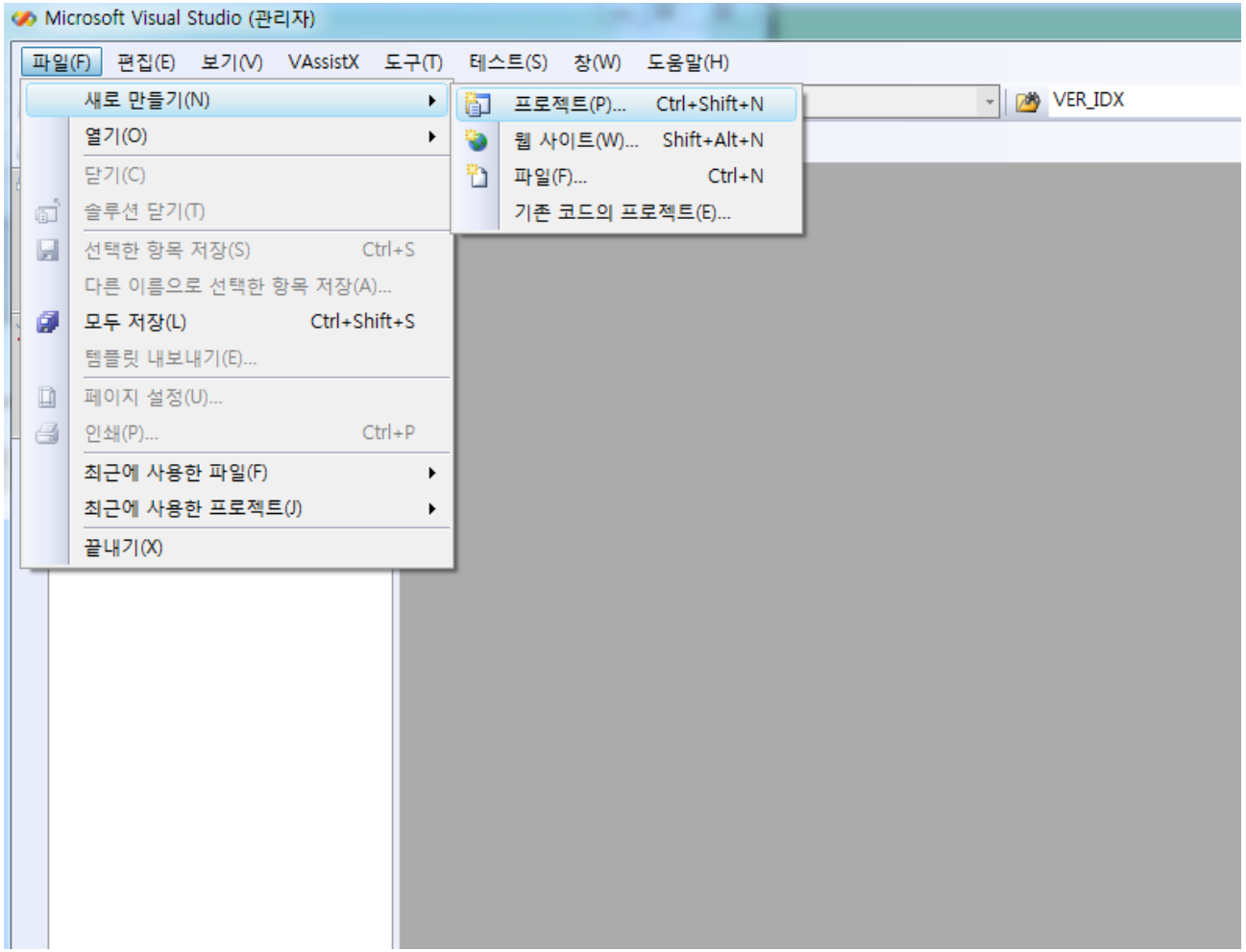
Lana 이미지 확대

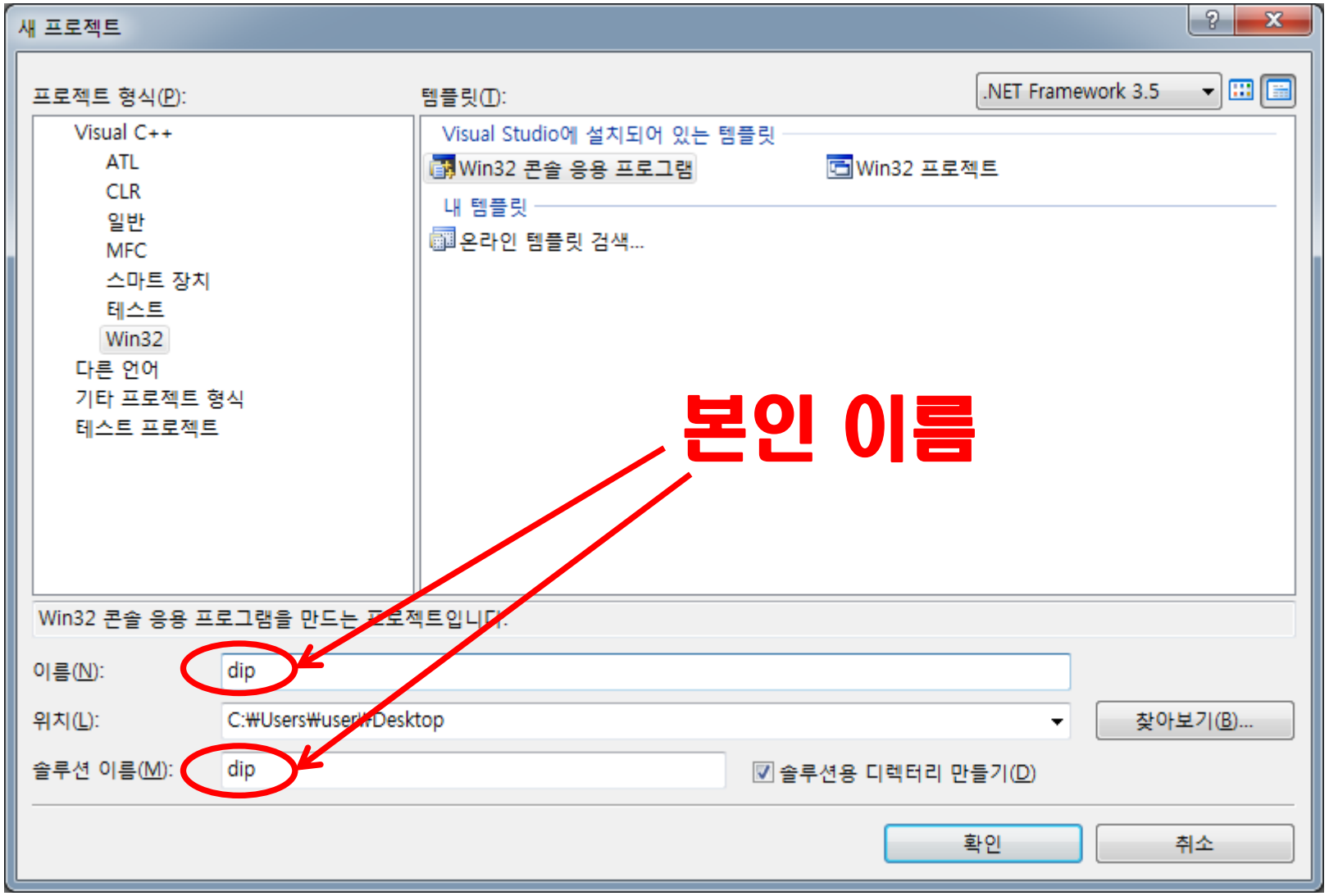
Lena 밝기 표현 방법

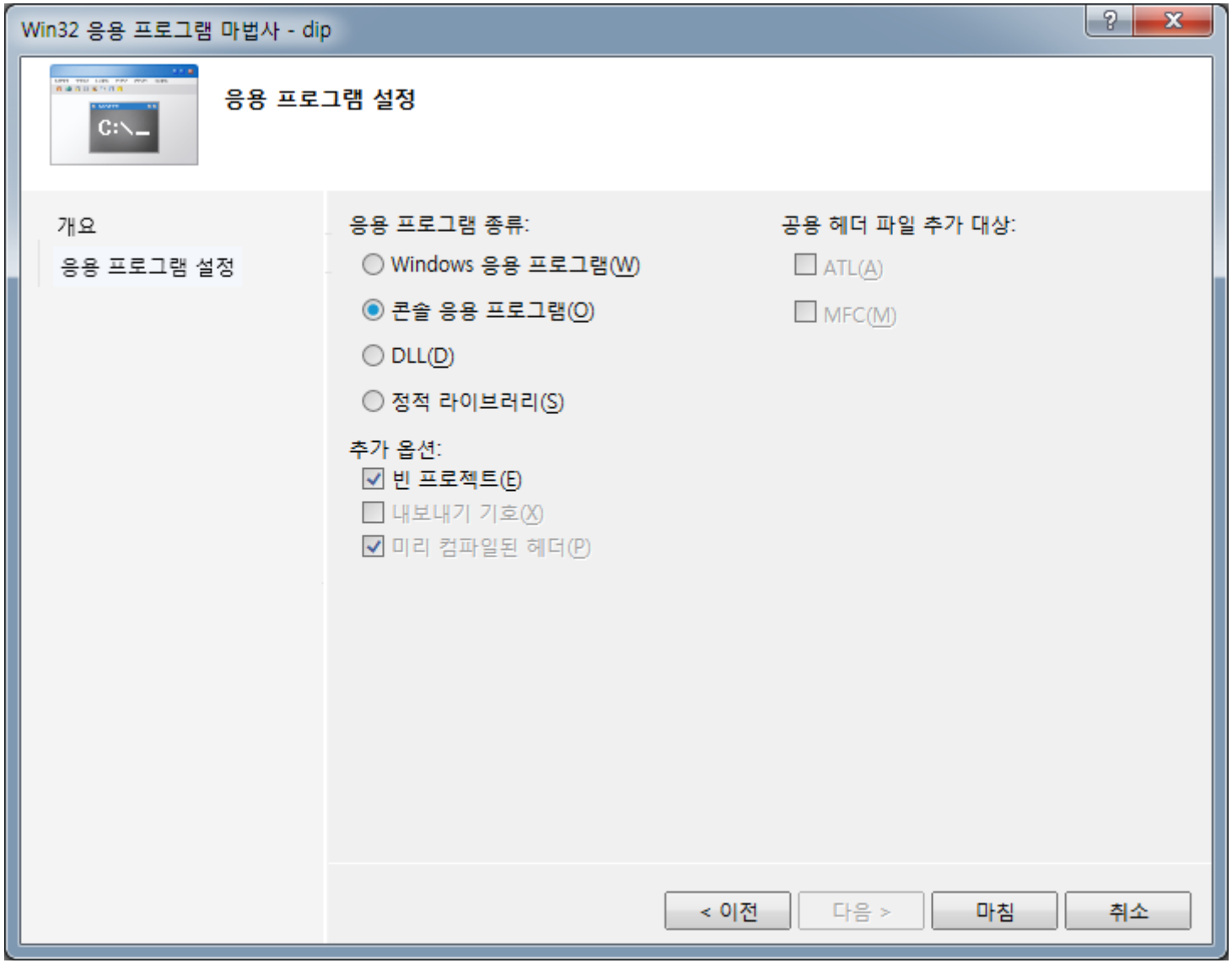


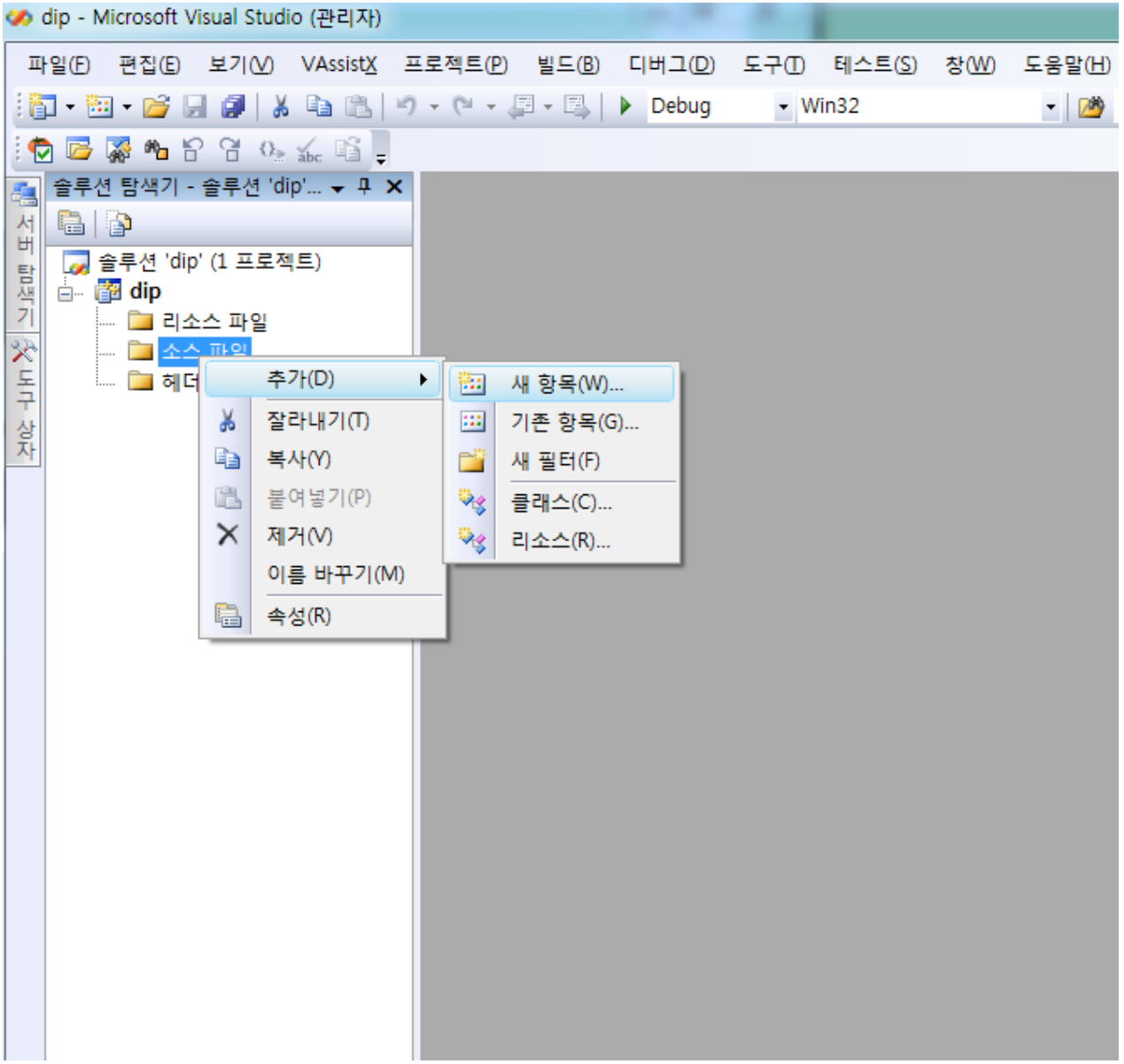
- 0~255
- (-) 음수가 없는 8bit 로 표현 가능
- 0000 0000 ~ 1111 1111
  - unsigned char 사용

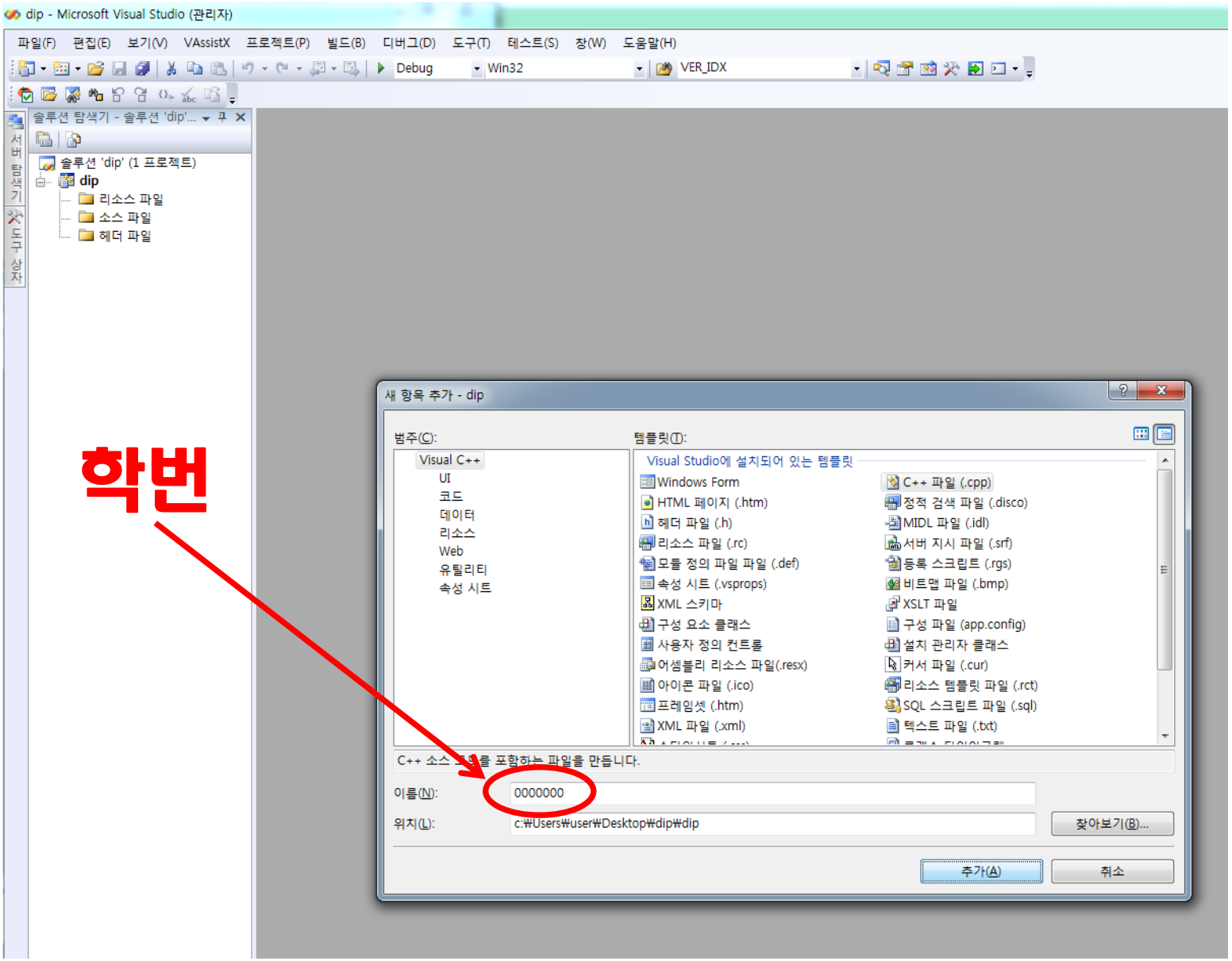
프로젝트 생성 과정 (2010 ver.)

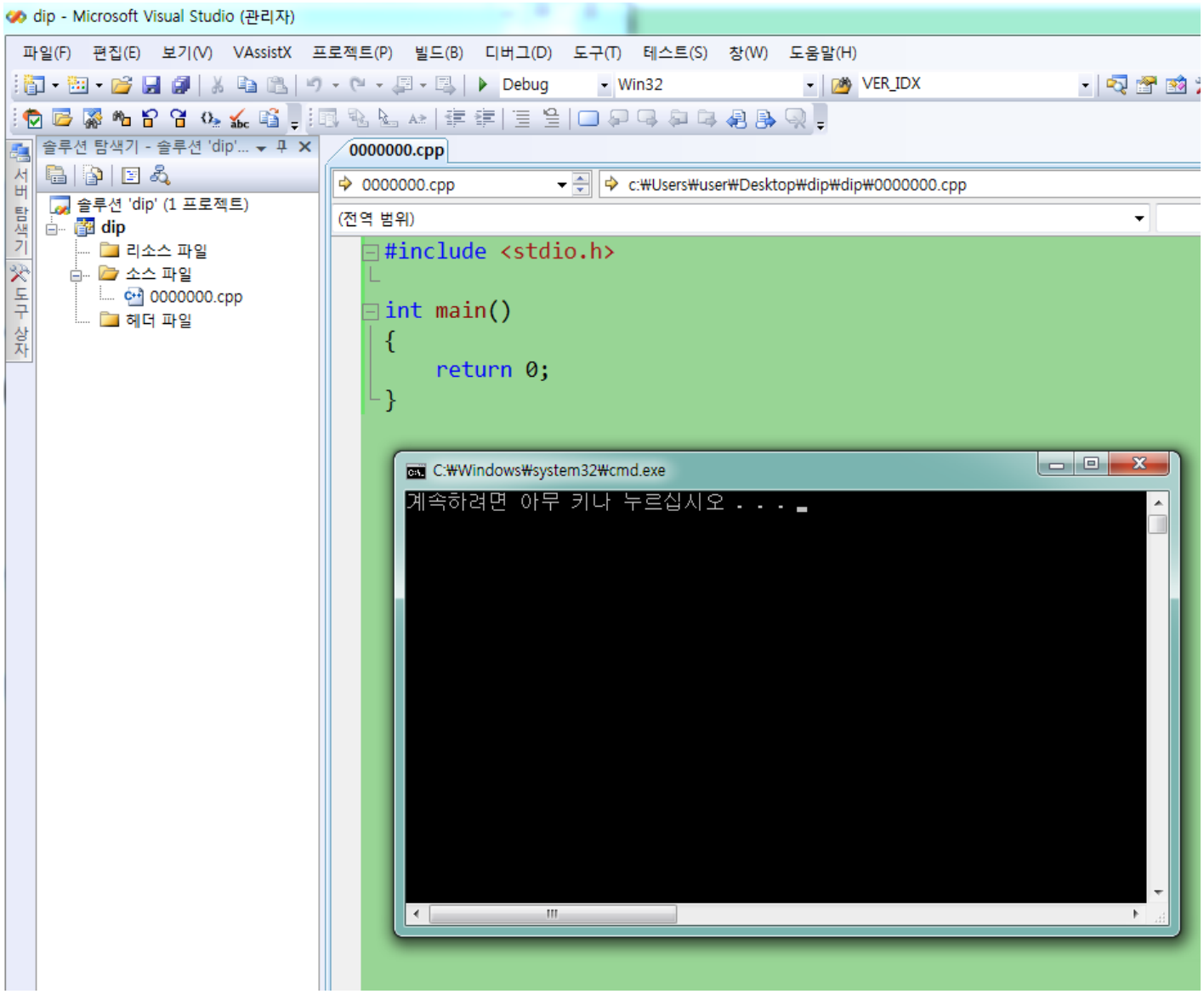



새 프로젝트
프로젝트 형식(P):
템플릿(T):
.NET Framework 3.5
Visual C++
ATL
CLR
일반
MFC
스마트 장치
테스트
Win32
다른 언어
기타 프로젝트 형식
테스트 프로젝트
Visual Studio에 설치되어 있는 템플릿
Win32 콘솔 응용 프로그램
Win32 프로젝트
내 템플릿
온라인 템플릿 검색...
본인 이름
Win32 콘솔 응용 프로그램을 만드는 프로젝트입니다.
이름(N):
dip
위치(L):
C:₩Users₩user₩Desktop
찾아보기(B)...
솔루션 이름(M):
dip
솔루션용 디렉터리 만들기(D)
확인
취소

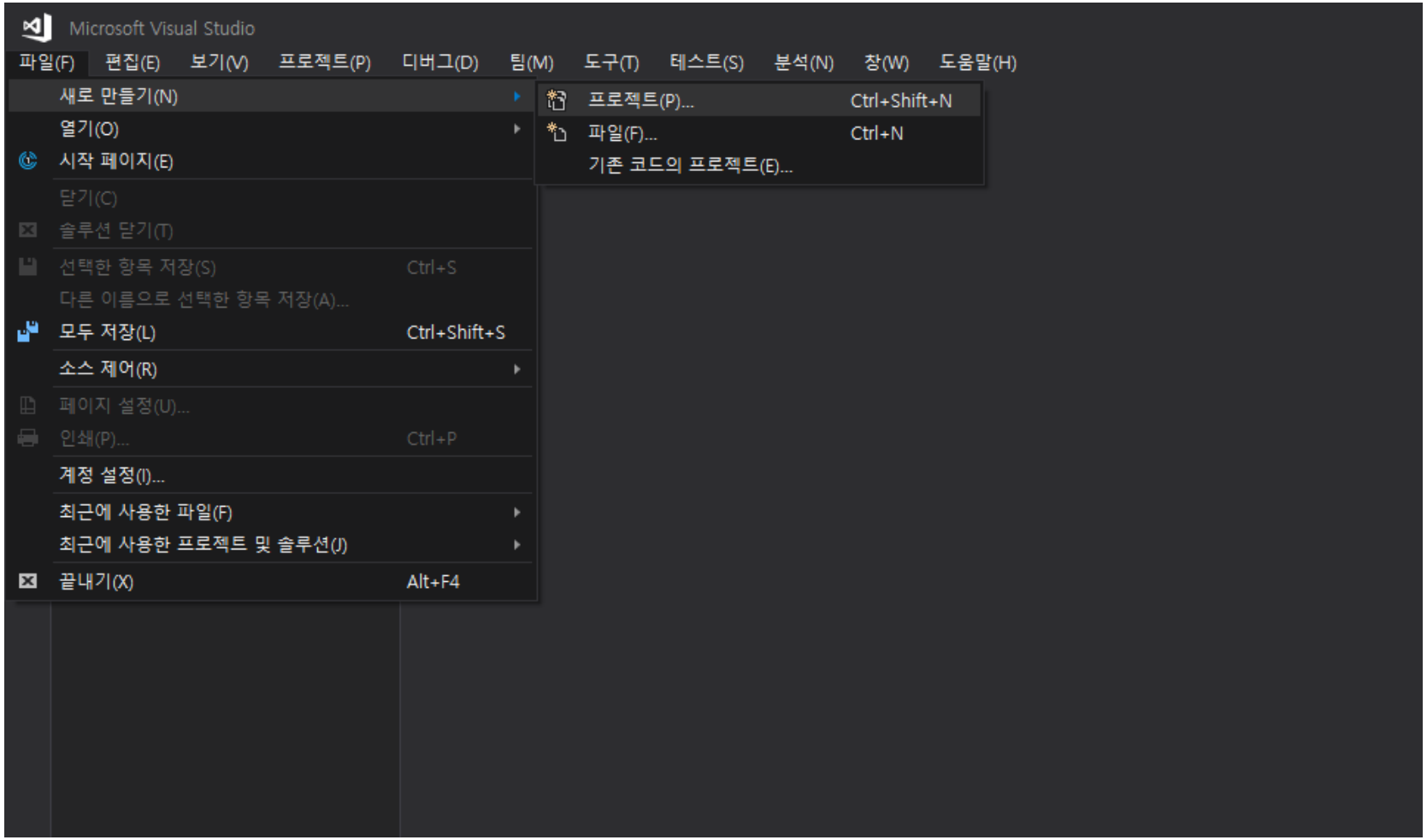

Win32 응용 프로그램 마법사 - dip
응용 프로그램 설정
개요
응용 프로그램 설정
응용 프로그램 종류:
Windows 응용 프로그램(W)
콘솔 응용 프로그램(O)
DLL(D)
정적 라이브러리(S)
추가 옵션:
빈 프로젝트(E)
내보내기 기호(X)
미리 컴파일된 헤더(P)
공용 헤더 파일 추가 대상:
ATL(A)
MFC(M)
< 이전
다음 >
마침
취소

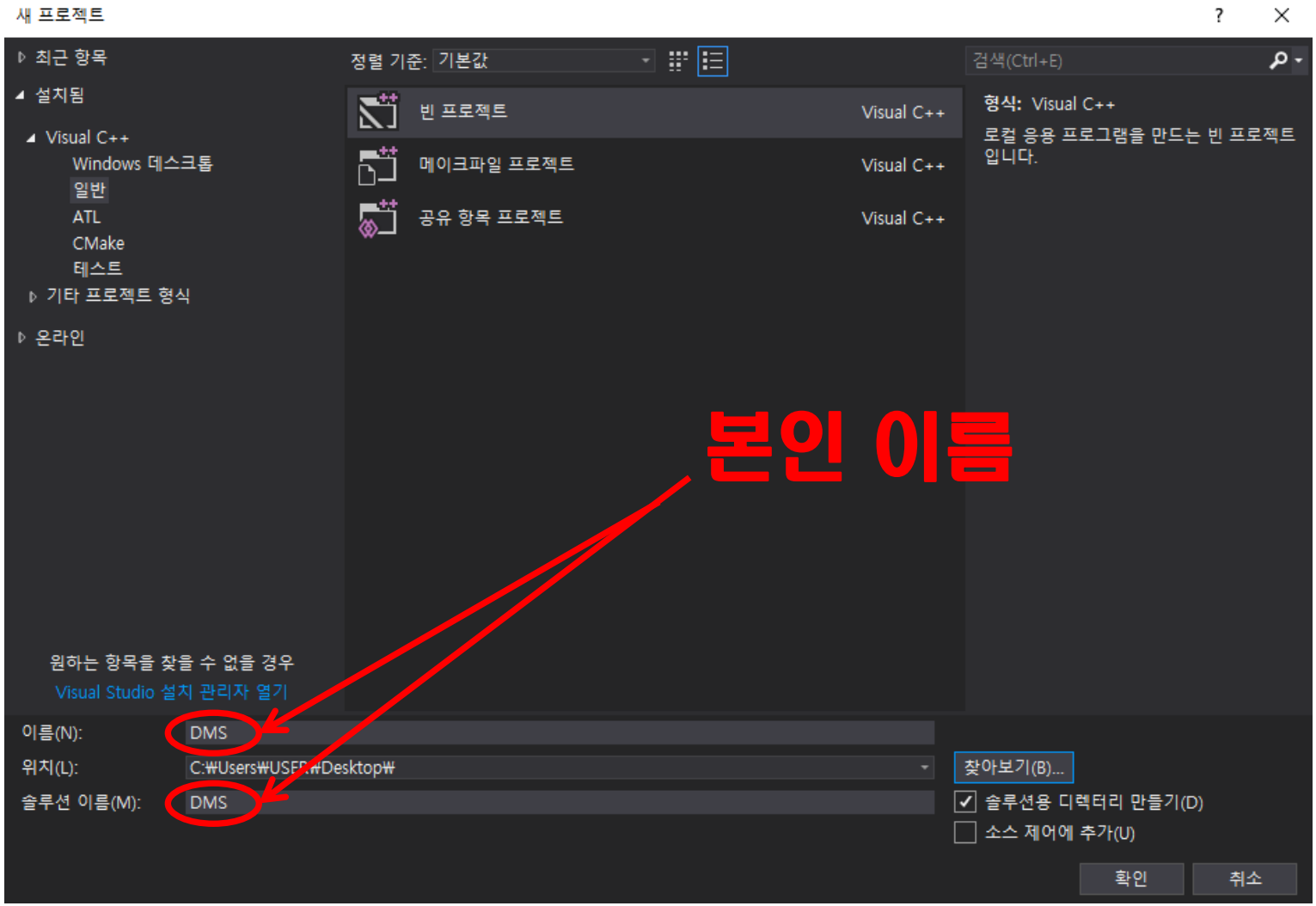

dip - Microsoft Visual Studio (관리자)
파일(F) 편집(E) 보기(V) VAssistX 프로젝트(P) 빌드(B) 디버그(D) 도구(T) 테스트(S) 창(W) 도움말(H)
Debug
Win32
솔루션 탐색기 - 솔루션 'dip'...
솔루션 'dip' (1 프로젝트)
dip
리소스 파일
추가(D)
잘라내기(T)
복사(Y)
붙여넣기(P)
제거(V)
이름 바꾸기(M)
속성(R)
새 항목(W)...
기존 항목(G)...
새 필터(F)
클래스(C)...
리소스(R)...
서버 탐색기
도구 상자


dip - Microsoft Visual Studio (관리자)
파일(F) 편집(E) 보기(V) VAssistX 프로젝트(P) 빌드(B) 디버그(D) 도구(T) 테스트(S) 창(W) 도움말(H)
Debug
Win32
VER_IDX
솔루션 탐색기 - 솔루션 'dip'...
솔루션 'dip' (1 프로젝트)
dip
리소스 파일
소스 파일
헤더 파일
학번
새 항목 추가 - dip
범주(C):
템플릿(T):
Visual C++
UI
코드
데이터
리소스
Web
유틸리티
속성 시트
Visual Studio에 설치되어 있는 템플릿
Windows Form
HTML 페이지 (.htm)
헤더 파일 (.h)
리소스 파일 (.rc)
모듈 정의 파일 파일 (.def)
속성 시트 (.vsprops)
XML 스키마
구성 요소 클래스
사용자 정의 컨트롤
어셈블리 리소스 파일(.resx)
아이콘 파일 (.ico)
프레임셋 (.htm)
XML 파일 (.xml)
C++ 파일 (.cpp)
정적 검색 파일 (.disco)
MIDL 파일 (.idl)
서버 지시 파일 (.srf)
등록 스크립트 (.rgs)
비트맵 파일 (.bmp)
XSLT 파일
구성 파일 (app.config)
설치 관리자 클래스
커서 파일 (.cur)
리소스 템플릿 파일 (.rct)
SQL 스크립트 파일 (.sql)
텍스트 파일 (.txt)
C++ 소스 코드를 포함하는 파일을 만듭니다.
이름(N):
0000000
위치(L):
c:₩Users₩user₩Desktop₩dip₩dip
찾아보기(B)...
추가(A)
취소

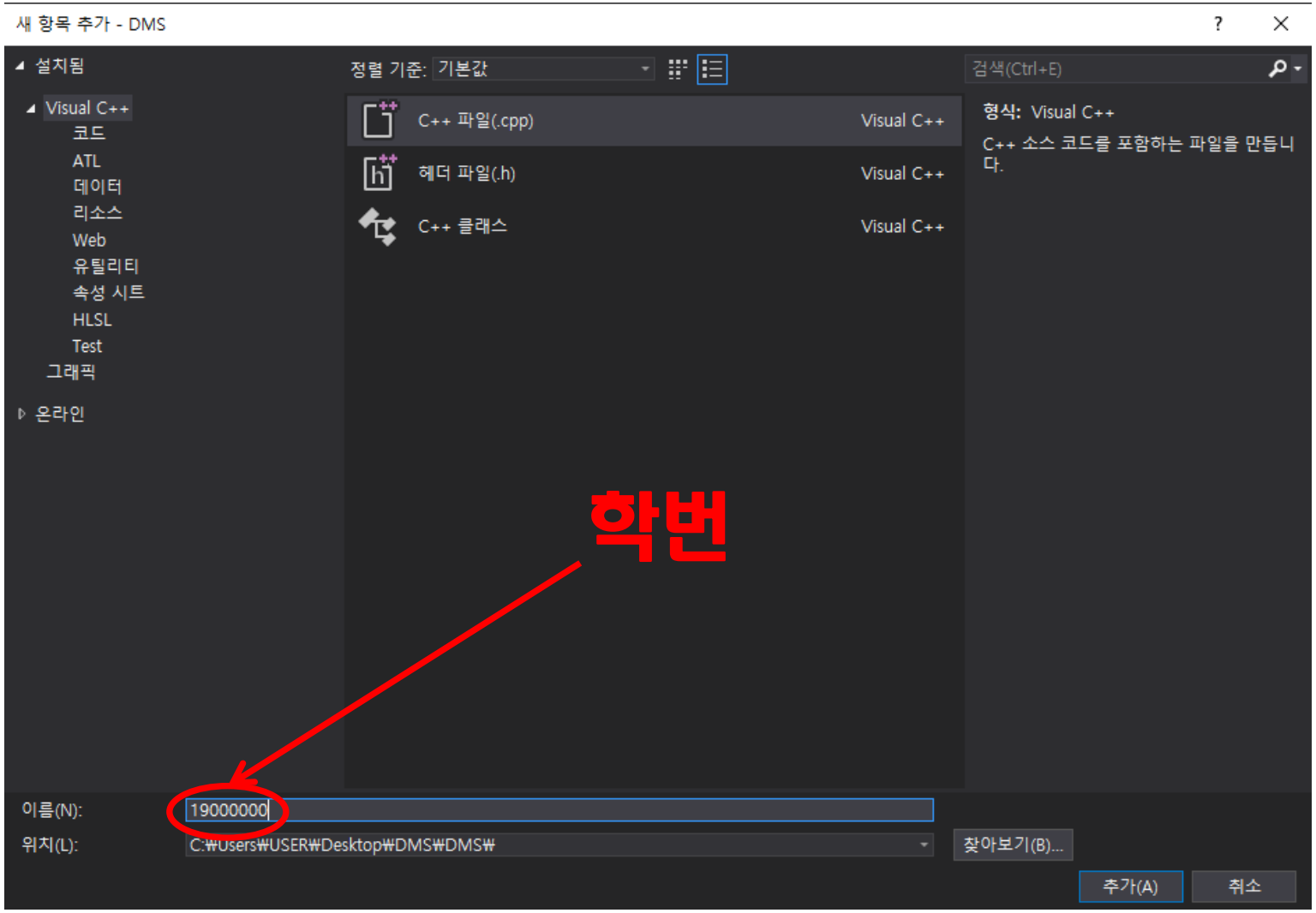

dip - Microsoft Visual Studio (관리자)
파일(F) 편집(E) 보기(V) VAssistX 프로젝트(P) 빌드(B) 디버그(D) 도구(T) 테스트(S) 창(W) 도움말(H)
Debug
Win32
VER_IDX
솔루션 탐색기 - 솔루션 'dip'...
솔루션 'dip' (1 프로젝트)
dip
리소스 파일
소스 파일
0000000.cpp
헤더 파일
서버 탐색기
도구 상자
0000000.cpp
c:₩Users₩user₩Desktop₩dip₩dip₩0000000.cpp
(전역 범위)
#include <stdio.h>
int main()
{
return 0;
}
C:₩Windows₩system32₩cmd.exe
계속하려면 아무 키나 누르십시오 . . .

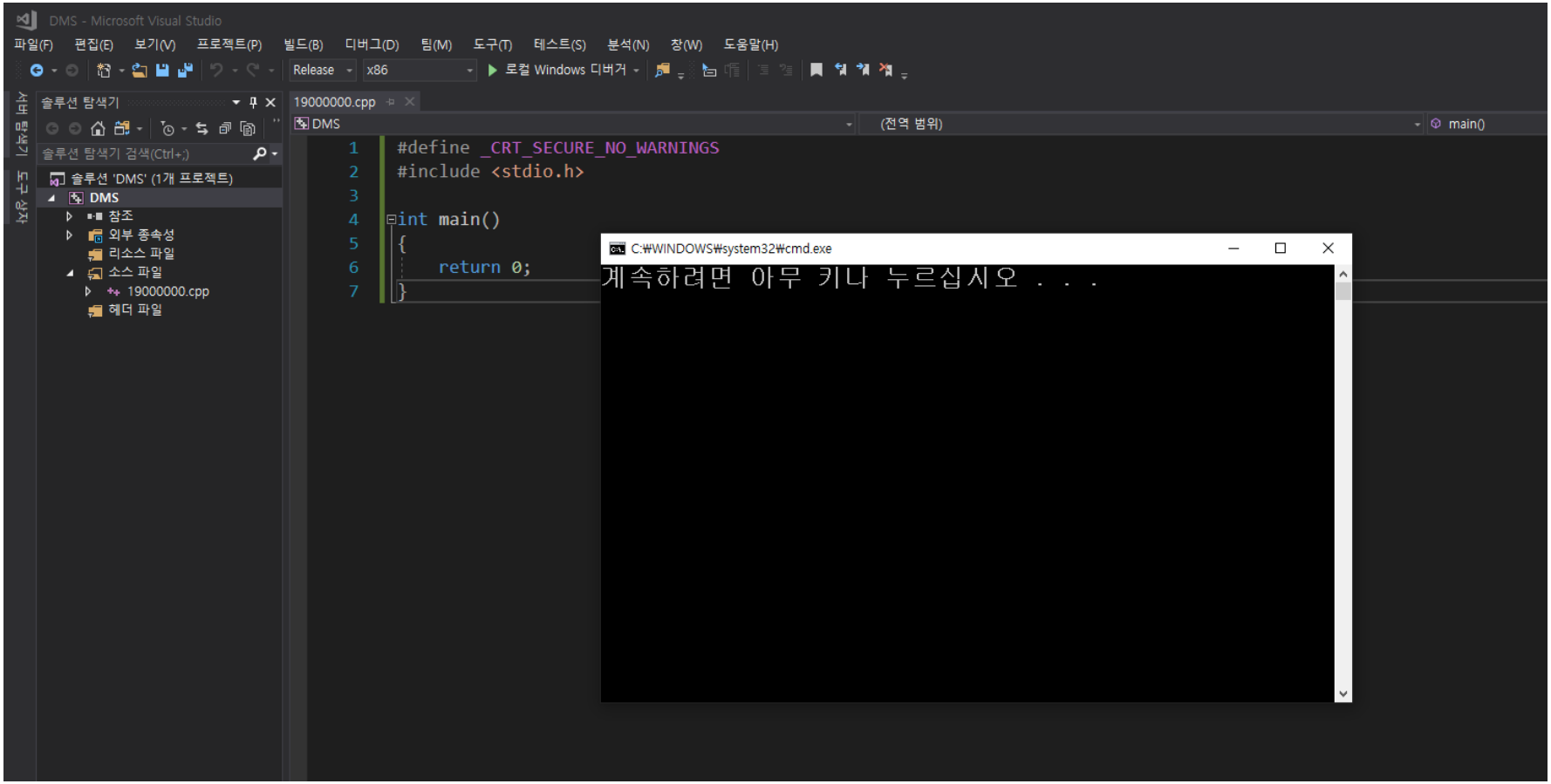
프로젝트 생성 과정 (2017 ver.)




새 프로젝트
최근 항목
설치됨
Visual C++
Windows 데스크톱
일반
ATL
CMake
테스트
기타 프로젝트 형식
온라인
정렬 기준: 기본값
빈 프로젝트 Visual C++
메이크파일 프로젝트 Visual C++
공유 항목 프로젝트 Visual C++
검색(Ctrl+E)
형식: Visual C++
로컬 응용 프로그램을 만드는 빈 프로젝트입니다.
본인 이름
원하는 항목을 찾을 수 없을 경우
Visual Studio 설치 관리자 열기
이름(N): DMS
위치(L): C:₩Users₩USER₩Desktop₩
솔루션 이름(M): DMS
찾아보기(B)...
솔루션용 디렉터리 만들기(D)
소스 제어에 추가(U)
확인
취소


DMS - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
Debug
x86
로컬 Windows 디버거
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'DMS' (1개 프로젝트)
DMS
참조
외부 종속성
리소스 파일
소스 파일
헤더 파일
추가(D)
클래스 마법사(Z)... Ctrl+Shift+X
여기까지 범위 지정(S)
새 솔루션 탐색기 뷰(N)
잘라내기(T) Ctrl+X
복사(Y) Ctrl+C
붙여넣기(P) Ctrl+V
삭제(D) Del
이름 바꾸기(M)
속성(R) Alt+Enter
새 항목(W)... Ctrl+Shift+A
기존 항목(G)... Shift+Alt+A
새 필터(F)
클래스(C)...
리소스(R)...


새 항목 추가 - DMS
설치됨
Visual C++
코드
ATL
데이터
리소스
Web
유틸리티
속성 시트
HLSL
Test
그래픽
온라인
정렬 기준: 기본값
C++ 파일(.cpp) Visual C++
헤더 파일(.h) Visual C++
C++ 클래스 Visual C++
검색(Ctrl+E)
형식: Visual C++
C++ 소스 코드를 포함하는 파일을 만듭니다.
학번
이름(N): 19000000
위치(L): C:₩Users₩USER₩Desktop₩DMS₩DMS₩
찾아보기(B)...
추가(A)
취소


DMS - Microsoft Visual Studio
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 팀(M) 도구(T) 테스트(S) 분석(N) 창(W) 도움말(H)
Release
x86
로컬 Windows 디버거
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'DMS' (1개 프로젝트)
DMS
참조
외부 종속성
리소스 파일
소스 파일
19000000.cpp
헤더 파일
19000000.cpp
DMS
(전역 범위)
main()
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
int main()
{
return 0;
}
C:₩WINDOWS₩system32₩cmd.exe
계속하려면 아무 키나 누르십시오 . . .

Function

- 메모리 동적 할당 및 해제
  - C : malloc(), free();
  - C++ : new, delete;

- 파일 입출력
  - 파일 열기/생성 : fopen()
  - 파일 닫기 : fclose()
  - 파일 읽기 : fread()
  - 파일에 데이터 쓰기 : fwrite()

File I/O function (ANSI C)

- FILE *fopen(const char *filename, const char *mode);

  ex) #include<stdio.h>

  ```
  FILE *fp;
  fp=fopen("lena.img","rb");
  ..............
  fclose(fp);

  fp=fopen("out.img","wb");
  ..............
  fclose(fp);
  ```

- fread(void *ptr, size_t size, size_t n, FILE *stream);

  ex) fread(a,sizeof(unsigned char),512*512,fp);

- fwrite(const void *ptr, size_t size, size_t n, FILE *stream);

  ex) fwrite(b,sizeof(unsigned char),512*512,fp);

1-D Memory Allocation (C)

```
#include<stdlib.h>
void *malloc(size_t size);
free(void * block);
```

Ex)
```
unsigned char *a;
a=(unsigned char*)malloc(sizeof(unsigned char)*512*512);
.....
.....
free(a);
```

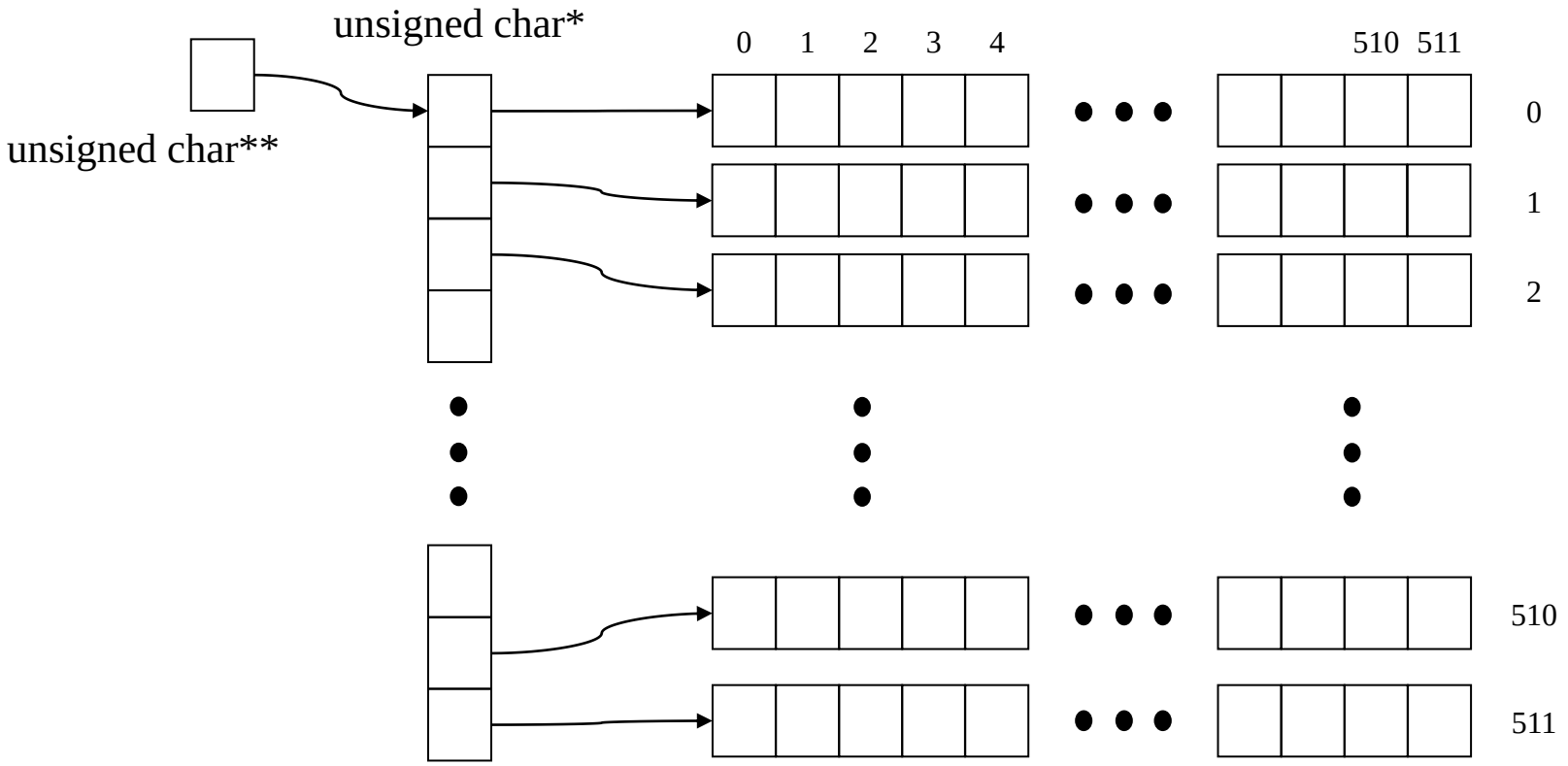
1-D memory allocation (C++)

- 예제코드

```
#include <stdio.h>
int main(void)
{
    // 이미지 읽기 및 메모리(pLenaImg)에 저장
    FILE* hLena = fopen("lena.img","rb");
    unsigned char *pLenaImg = new unsigned char[512*512];

    fread(pLenaImg, 1, 512*512, hLena);
    fclose(hLena);

    // 메모리에 저장된 이미지를 이용하여 영상처리 코드 구현
    unsigned char *pOutputImg = new unsigned char[512*512];
    ...

    // 파일에 결과물 출력, 동적할당 해제
    FILE* hOutput = fopen("output.img","wb");
    fwrite(pOutputImg, 1, 512*512, hOutput);
    fclose(hOutput);
    delete[] pLenaImg;
    delete[] pOutputImg;

    return 0;
}
```

Basic 2-D Memory allocation

a[512][512]



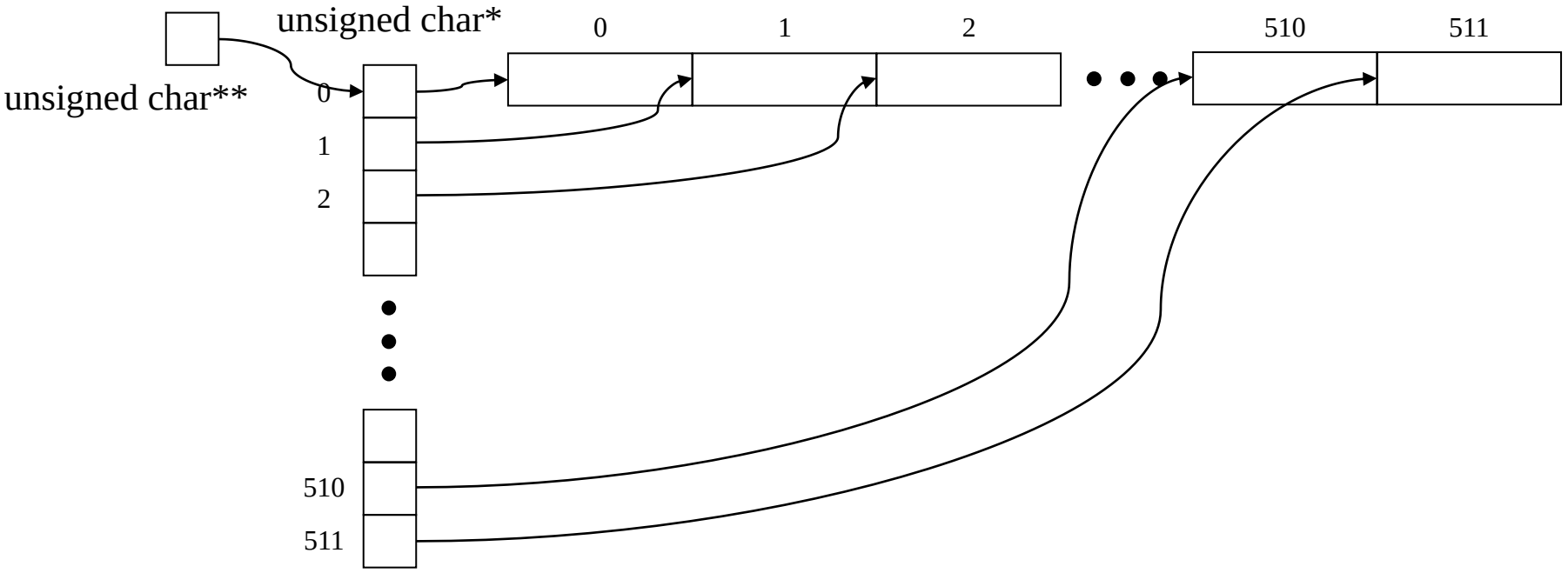
Basic 2D Memory Allocation in C

2-Dimensional malloc() and image read

```
#include<stdio.h>
#include<math.h>
ex)     unsigned char **a; //since image is 2D data
        a=(unsigned char**)malloc(sizeof(unsigned char*)*512); //Height
        for(i=0;i<512;i++)
            a[i]=(unsigned char*)malloc(sizeof(unsigned char)*512);

        fp=fopen("lena.img","rb");
        for(i=0;i<512;i++)
            fread(a[i],sizeof(unsigned char),512,fp);

        // Do operation like Convolution, DFT,.....
      ..........................................................................

        for(i=0;i<512;i++)
            free(a[i]);
        free(a);
```

C

Advanced 2-D memory allocation in C



Advanced 2-D memory allocation

- 2차원 배열 할당 예제 (http://codeng.tistory.com/8 참조)

C++

```
unsigned char** _2dAlloc(int width, int height)
{
    int i;
    unsigned char** ppA = new unsigned char*[height];
    ppA[0] = new unsigned char[width*height];
    for(i=1;i<height;i++) ppA[i] = ppA[i-1]+width;

    return ppA;
}
```

typedef uint8 unsigned char
typedef uint32 unsigned int

C

```
unsigned char** memory_alloc2D(int height, int width)
{
 unsigned char**   ppMem2D = 0;
 int       j, i;

  // array of pointer
  ppMem2D = (unsigned char **)malloc(sizeof(unsigned char *)* height);

  *ppMem2D = (unsigned char *)malloc(sizeof(unsigned char)*(width*height));

  for (j=1; j< height; j++ )
  {
    ppMem2D[j] = ppMem2D[j-1] + width;
  }
  return ppMem2D;
}
```

Memory Allocation

- 2차원 배열을 이용한 파일 입출력 예제

```
#include <stdio.h>
int main(void)
{
    FILE* hLena = fopen("lena.img", "rb");
    unsigned char** ppLena = _2dAlloc(512,512);
    fread(ppLena[0], 1, 512*512, hLena);

    ...

    fclose(hLena);
    delete[] ppLena[0];
    delete[] ppLena;
    return 0;
}
```

C++

example

typedef uint8 unsigned char
typedef uint32 unsigned int

```
uint8** memory_alloc2D(uint32 height, uint32 width)
{
  uint8**   ppMem2D = 0;
  uint32    j, i;

  // array of pointer
  ppMem2D = (uint8**)calloc(sizeof(uint8*), height);
  if ( ppMem2D == 0 )
  {
    return 0;
  }

  *ppMem2D = (uint8*)calloc(sizeof(uint8), width * height );
  if( (*ppMem2D) == 0 )
  {
    // free the memory of array of pointer
    free( ppMem2D );
    return 0;
  }

  for (j=1; j< height; j++ )
  {
    ppMem2D[j] = ppMem2D[j-1] + width;
  }

  return ppMem2D;
}
```

C

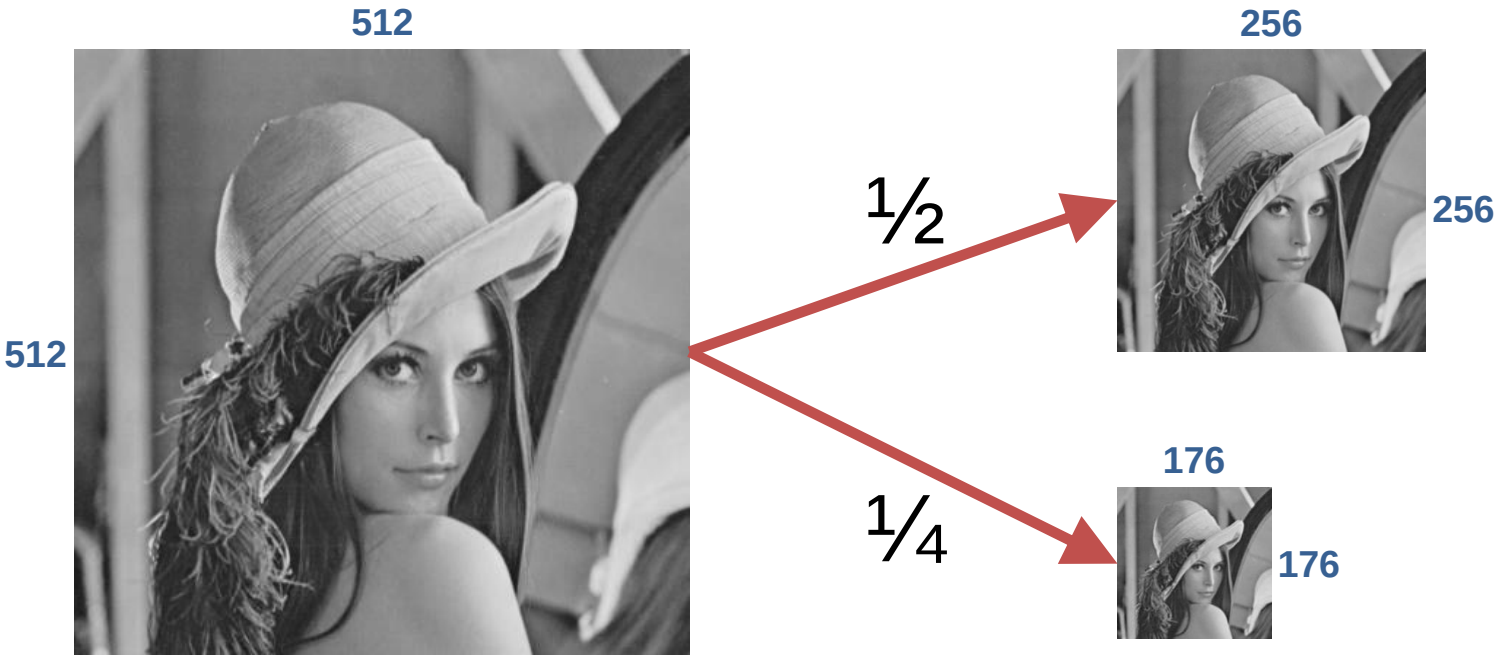
```
int memory_free2D(uint8** ppMemAllocated)
{
  if (ppMemAllocated == 0 )
  {
    return -1;
  }

  free( ppMemAllocated[0] );
  free( ppMemAllocated );

  return 0;
```

Image read/write fuction (ANSI C)

```
int main(void)
{
  FILE*     fpInputImage = 0;
  FILE*     fpOutputImage = 0;
  uint8**   ppInputImageBuffer= 0;

  // input file open
  fpInputImage = fopen(IMG_NAME, "rb");

  // memory allocaiton
  ppInputImageBuffer = memory_alloc2D(IMG_HEIGHT, IMG_WIDTH);

  // input file read to memory from the file
  fread( &ppInputImageBuffer[0][0], sizeof(uint8), IMG_WIDTH*IMG_HEIGHT, fpInputImage);

  // output fileopen
  fpOutputImage = fopen("result.raw", "wb");

  // write the file
  fwrite( &ppInputImageBuffer[0][0], sizeof(uint8), IMG_WIDTH*IMG_HEIGHT, fpOutputImage);

  memory_free2D(ppInputImageBuffer);
  fclose(fpInputImage);
  fclose(fpOutputImage);

  return 0;
}
```

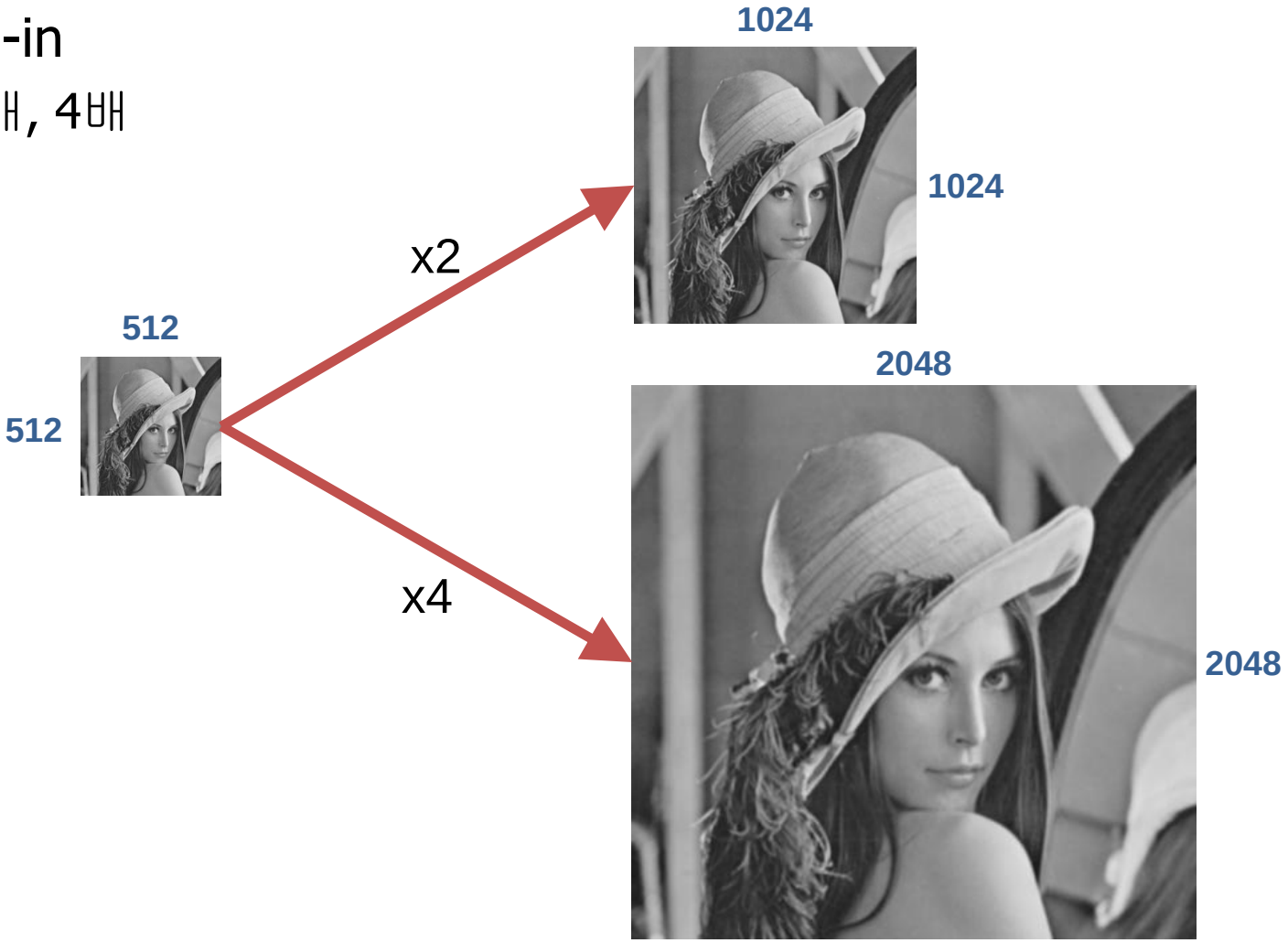
1. Zoom-out

- Zoom-out
  - 1/2배, 1/4배



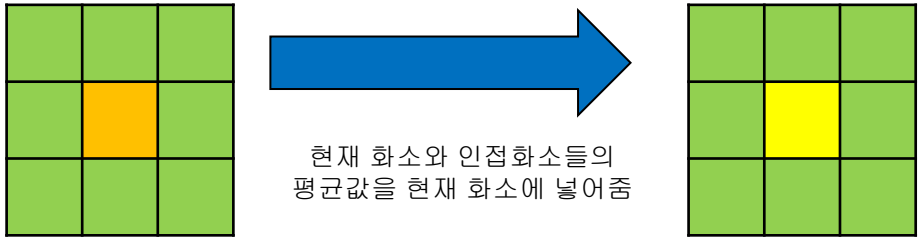
2. Zoom-in

- Zoom-in
  - 2배, 4배



3. Low-pass filtering

- Average filter 이용



: 현재 화소 : 인접 화소 : 필터링 된 화소

4. High-pass filtering

- Sobel operator 이용

+1	+2	+1
0	0	0
-1	-2	-1

< 수 평 >

+1	0	-1
+2	0	-2
+1	0	-1

< 수 직 >

여러 가지 **High-pass filtering**의 예

Roberts operator

-1	0	0
0	1	0
0	0	0

0	0	-1
0	1	0
0	0	0

Prewitt operator

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Sobel operator

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Frei-chen operator

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

5. Zoom-out and translation

- Zoom-out and translation
  - Lena 영상 크기 256 x 256
  - 검정 배경 크기 512x512

$$x' = x + dx$$
$$y' = y + dy$$

(50,50)

(100,80)

6. Zoom-out and translation and rotation

- Translation - (200,20)

$$x' = x + dx$$
$$y' = y + dy$$

- Rotation - (θ = 0.2, 0.5): radian

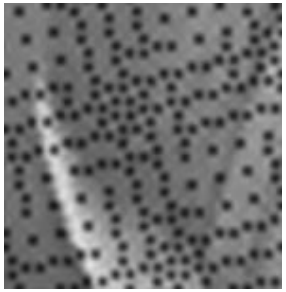
$$\mathrm{x}'' = \mathrm{x}' * \cos(\theta) - y' * sin(\theta)$$
$$\mathrm{y}'' = \mathrm{x}' * sin(\theta) + y' * cos(\theta)$$

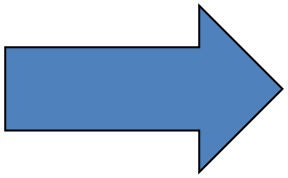
6. Zoom-out and translation and rotation

$(200,20)$
$\theta = 0.2$

$(200,20)$
$\theta = 0.5$

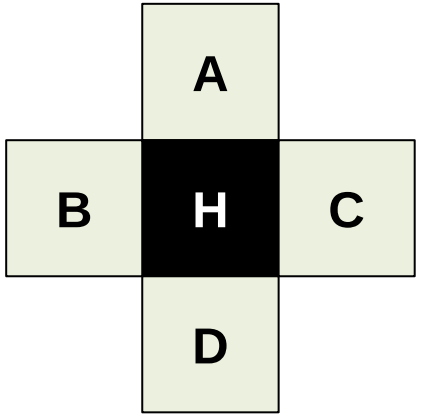
Hole 채우기





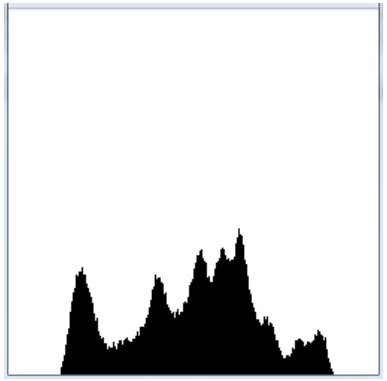
ex) 홀(Hole) 채우기

$$H = (A+B+C+D)/4$$

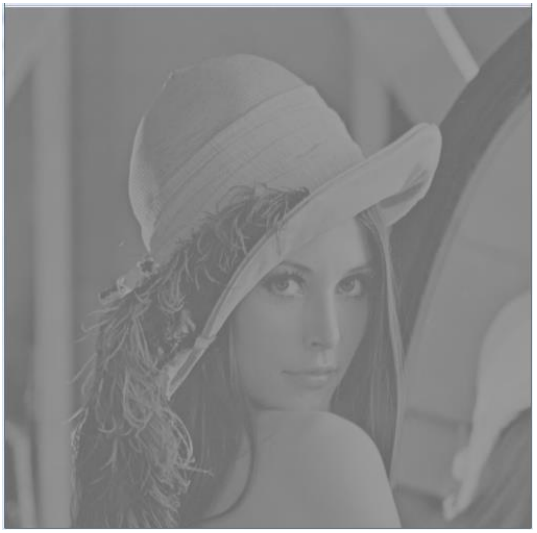


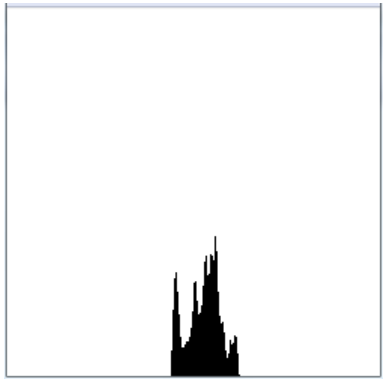
7. Histogram Slide-mapping & Stretch-mapping

Lena

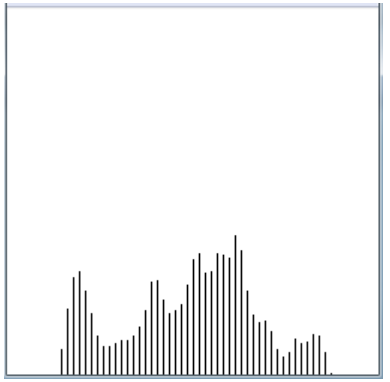


Slide mapping



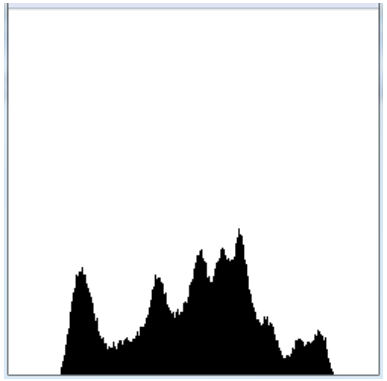


Stretch mapping



7. Histogram Equalization

Lena



주의사항

- Image Boundary
  - Don't touch
  - Image filtering

- 이미지 파일 출력 시 pixel range 를 0~255로 맞춰줘야 함.
  - 절대값
    - ✓ DPCM, Closing-Opening 등
  - clipping
    - ✓ Image filtering 등

```
if(img[i][j] < 0)
  clip = 0;
else if(img[i][j] > 255)
  clip = 255;
else
  clip = img[i][j];
```