Image Format

이진영

SEJONG UNIVERSITY

Image File Format

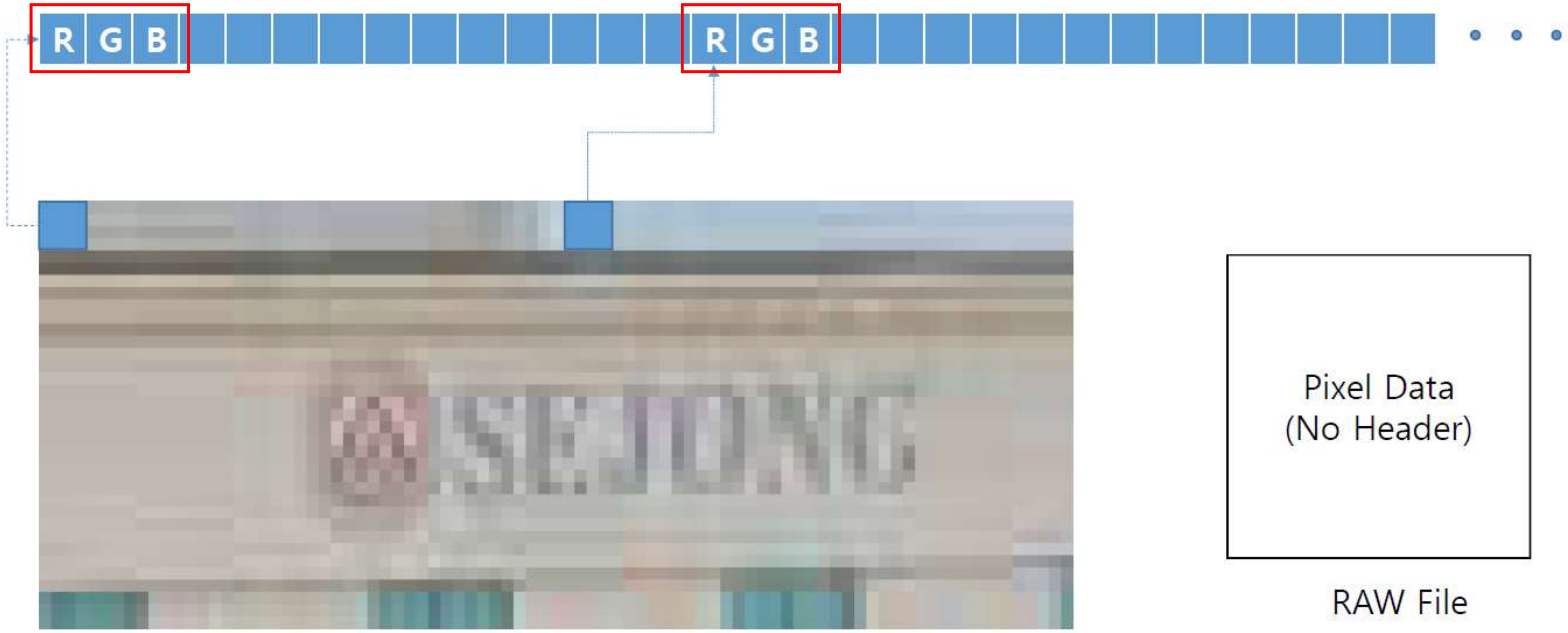
- RAW – Original
- JPG – Lossy compression
- BMP
- TIFF
- PNG
- …

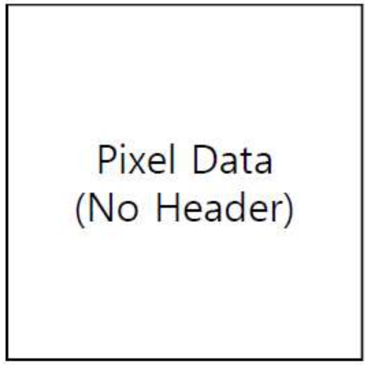
BMP, TIFF, PNG: Lossless compression

※ What type of a file for saving?

- Perfect copy of an original image
- Samll file size or Web service

RAW File Format



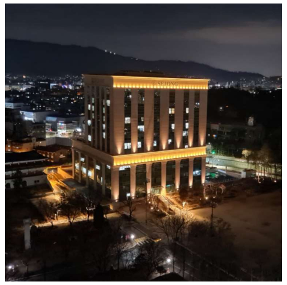


RAW File

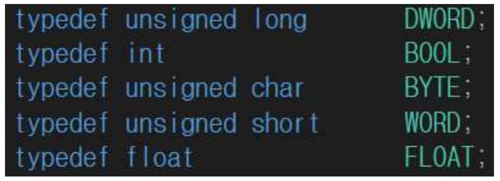
BMP I/O

- AICenter.bmp
  - Download from 집현캠퍼스 (Class Files)
  - 512×512

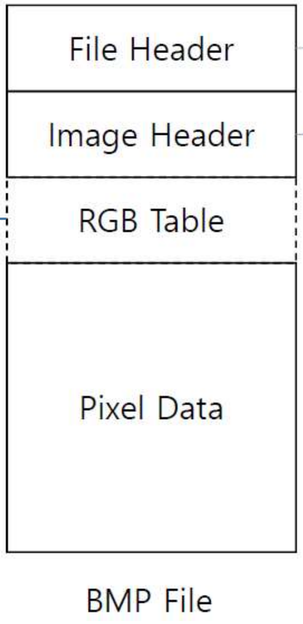
- 그림판 – 다른이름으로저장 - *.bmp



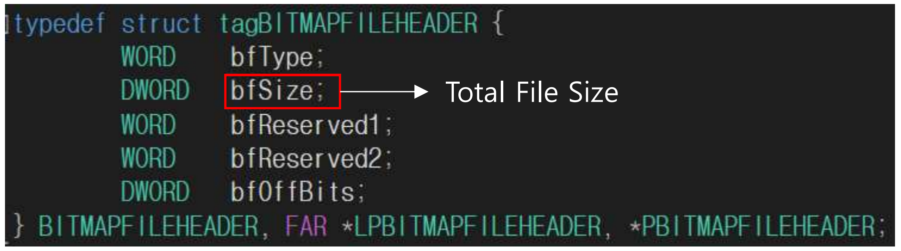
BMP File Format



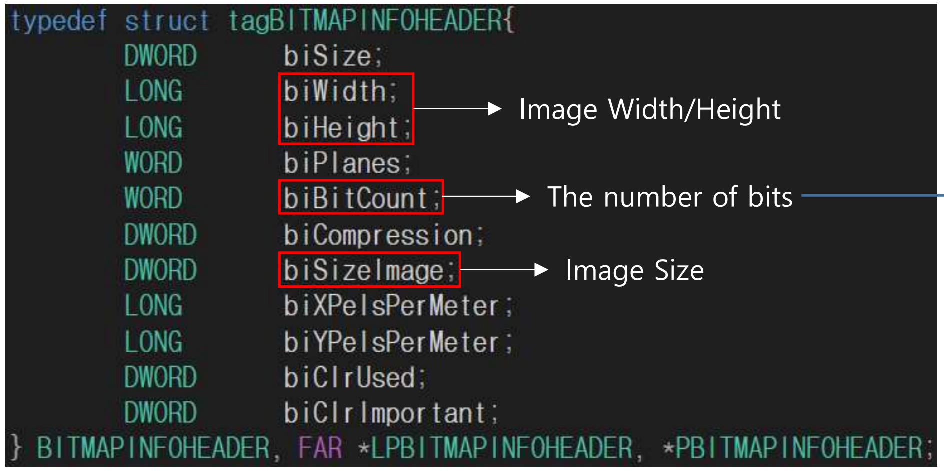
```
typedef unsigned long        DWORD;
typedef int                  BOOL;
typedef unsigned char        BYTE;
typedef unsigned short       WORD;
typedef float                FLOAT;
```



BMP File



```
typedef struct tagBITMAPFILEHEADER {
        WORD    bfType;
        DWORD   bfSize;
        WORD    bfReserved1;
        WORD    bfReserved2;
        DWORD   bfOffBits;
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER, *PBITMAPFILEHEADER;
```



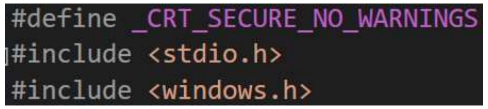
```
typedef struct tagBITMAPINFOHEADER{
        DWORD      biSize;
        LONG       biWidth;
        LONG       biHeight;
        WORD       biPlanes;
        WORD       biBitCount;
        DWORD      biCompression;
        DWORD      biSizeImage;
        LONG       biXPelsPerMeter;
        LONG       biYPelsPerMeter;
        DWORD      biClrUsed;
        DWORD      biClrImportant;
} BITMAPINFOHEADER, FAR *LPBITMAPINFOHEADER, *PBITMAPINFOHEADER;
```

RGB Table: Not used in our experiments

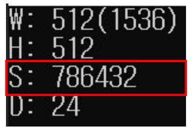
Bit-depth =24 in our experiments
(RGB × 8 bit-depth = 24 bits)

SEJONG UNIVERSITY

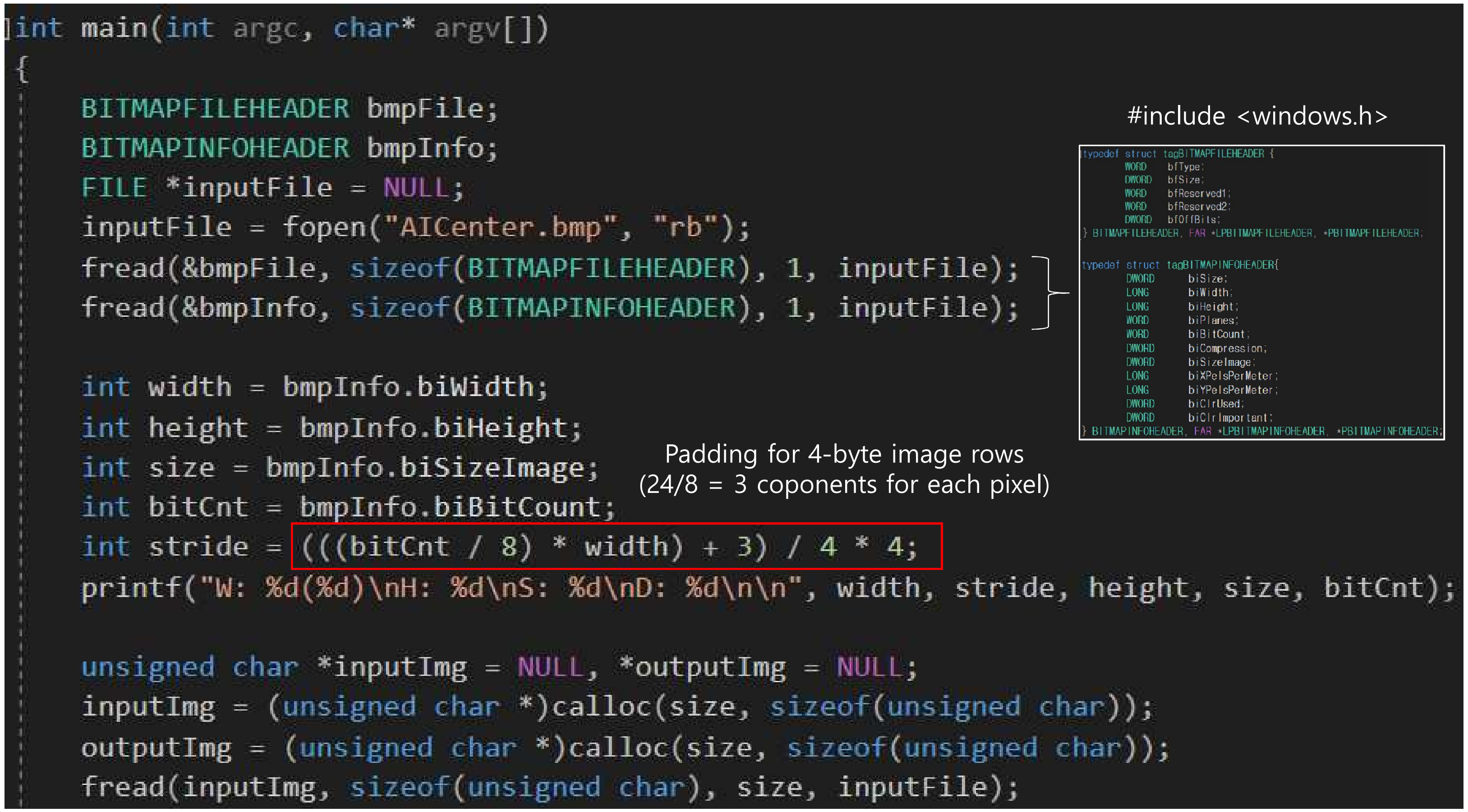
BMP File Input



```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <windows.h>
```



```
W: 512(1536)
H: 512
S: 786432
D: 24
```



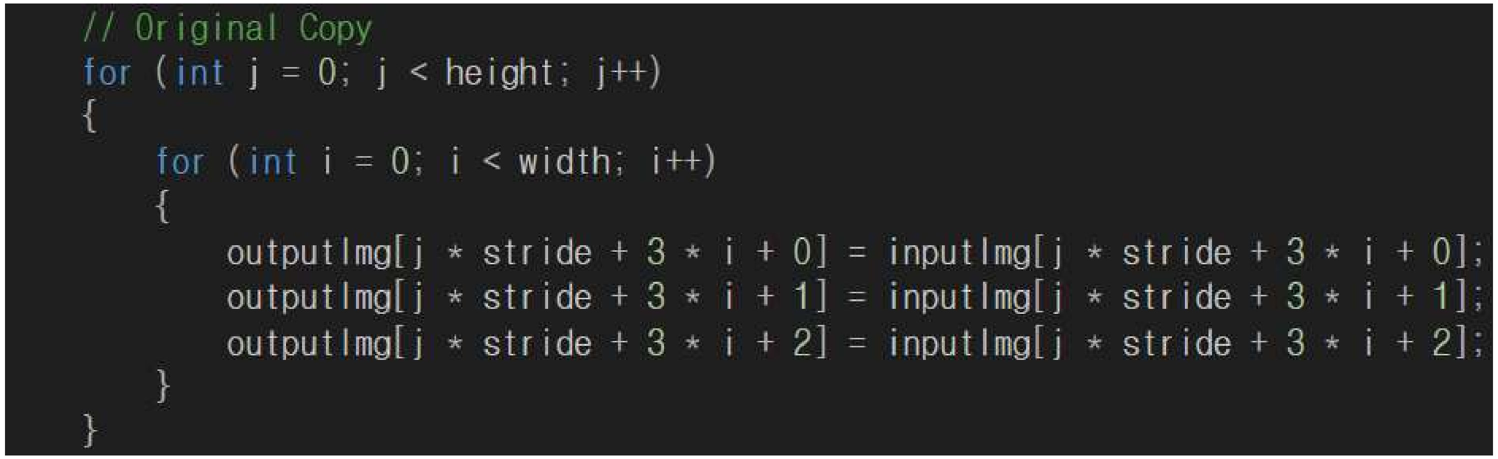
```
int main(int argc, char* argv[])
{
    BITMAPFILEHEADER bmpFile;
    BITMAPINFOHEADER bmpInfo;
    FILE *inputFile = NULL;
    inputFile = fopen("AICenter.bmp", "rb");
    fread(&bmpFile, sizeof(BITMAPFILEHEADER), 1, inputFile);
    fread(&bmpInfo, sizeof(BITMAPINFOHEADER), 1, inputFile);

    int width = bmpInfo.biWidth;
    int height = bmpInfo.biHeight;
    int size = bmpInfo.biSizeImage;
    int bitCnt = bmpInfo.biBitCount;
    int stride = (((bitCnt / 8) * width) + 3) / 4 * 4;
    printf("W: %d(%d)\nH: %d\nS: %d\nD: %d\n\n", width, stride, height, size, bitCnt);

    unsigned char *inputImg = NULL, *outputImg = NULL;
    inputImg = (unsigned char *)calloc(size, sizeof(unsigned char));
    outputImg = (unsigned char *)calloc(size, sizeof(unsigned char));
    fread(inputImg, sizeof(unsigned char), size, inputFile);
```

#include <windows.h>

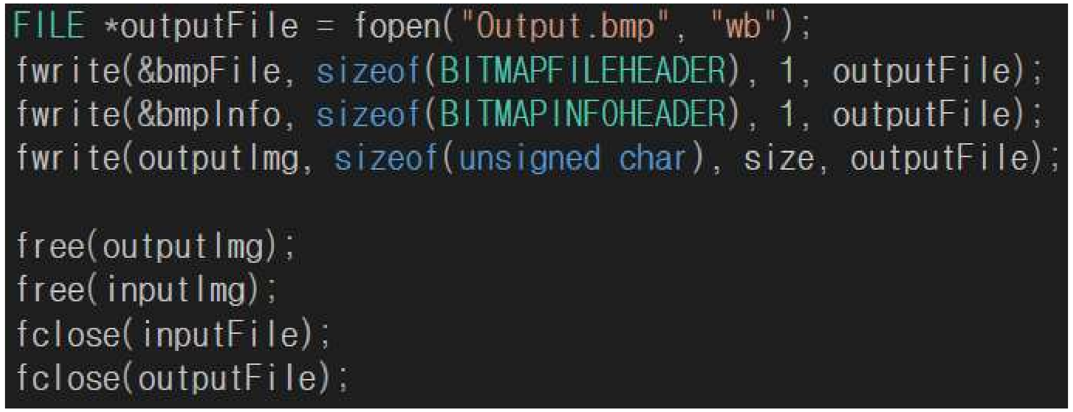
```
typedef struct tagBITMAPFILEHEADER {
        WORD    bfType;
        DWORD   bfSize;
        WORD    bfReserved1;
        WORD    bfReserved2;
        DWORD   bfOffBits;
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER, *PBITMAPFILEHEADER;

typedef struct tagBITMAPINFOHEADER{
        DWORD      biSize;
        LONG       biWidth;
        LONG       biHeight;
        WORD       biPlanes;
        WORD       biBitCount;
        DWORD      biCompression;
        DWORD      biSizeImage;
        LONG       biXPelsPerMeter;
        LONG       biYPelsPerMeter;
        DWORD      biClrUsed;
        DWORD      biClrImportant;
} BITMAPINFOHEADER, FAR *LPBITMAPINFOHEADER, *PBITMAPINFOHEADER;
```

Padding for 4-byte image rows
(24/8 = 3 coponents for each pixel)

BMP File Output

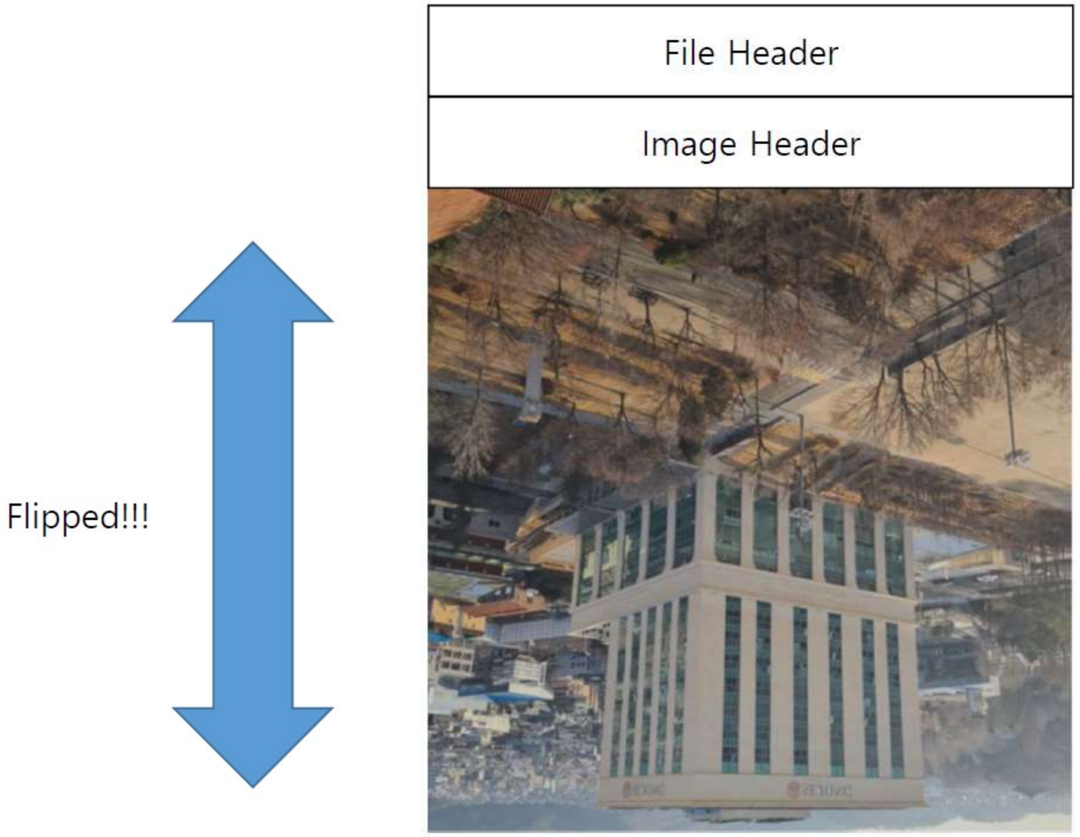


```
    // Original Copy
    for (int j = 0; j < height; j++)
    {
        for (int i = 0; i < width; i++)
        {
            outputImg[j * stride + 3 * i + 0] = inputImg[j * stride + 3 * i + 0];
            outputImg[j * stride + 3 * i + 1] = inputImg[j * stride + 3 * i + 1];
            outputImg[j * stride + 3 * i + 2] = inputImg[j * stride + 3 * i + 2];
        }
    }
```

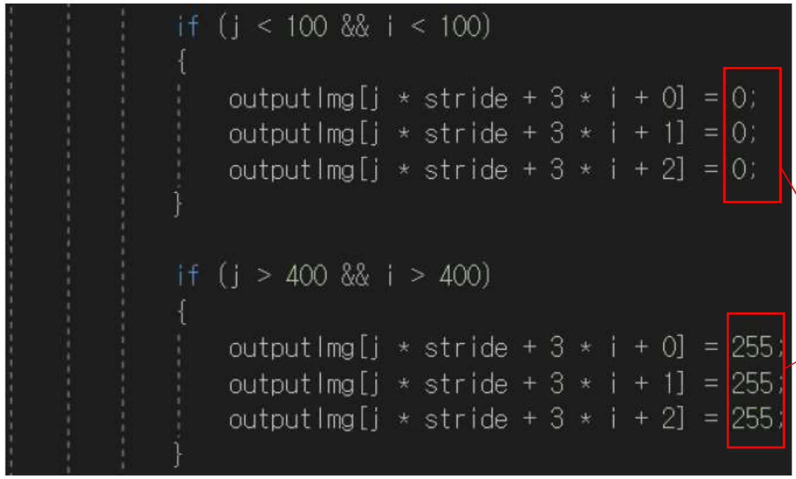


```
FILE *outputFile = fopen("Output.bmp", "wb");
fwrite(&bmpFile, sizeof(BITMAPFILEHEADER), 1, outputFile);
fwrite(&bmpInfo, sizeof(BITMAPINFOHEADER), 1, outputFile);
fwrite(outputImg, sizeof(unsigned char), size, outputFile);

free(outputImg);
free(inputImg);
fclose(inputFile);
fclose(outputFile);
```

BMP Pixel Data



Experiment



```
if (j < 100 && i < 100)
{
    outputImg[j * stride + 3 * i + 0] = 0;
    outputImg[j * stride + 3 * i + 1] = 0;
    outputImg[j * stride + 3 * i + 2] = 0;
}

if (j > 400 && i > 400)
{
    outputImg[j * stride + 3 * i + 0] = 255;
    outputImg[j * stride + 3 * i + 1] = 255;
    outputImg[j * stride + 3 * i + 2] = 255;
}
```