



# 图像信息处理实验报告

**Digital Image Processing (Experiments)**

|  |  |
| --- | --- |
| 姓 名 |  |
| 指导老师 |  |
| 学 号 |  |
| 专业班级 |  |

**二〇一九年**

**秋冬学期**

## 实验一

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称：** | 作业1 | **指导老师：** |  | **成绩：** |  |

### 一、实验目的和要求（必填）

1、读入一张真彩色bmp图（Read a color bmp）

2、将bmp图片中的rgb数据转换为yuv数据（RGB->YUV）

3、在yuv色域中，将y值赋予给灰度值（Color to gray:gray=Y in YUV color space）

4、将灰度值在0-255范围内重新分布（Rearrange gray intensity to lie between[0,255]）

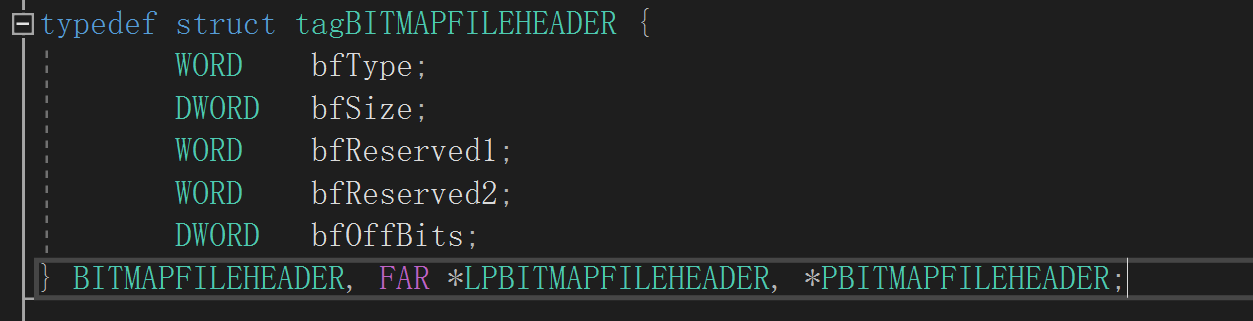
5、输出一张灰度图（Write a grayscale bmp）

6、改变Y值（Change the luminance value Y）

7、将改变后的yuv转换为rgb（YUV->RGB）

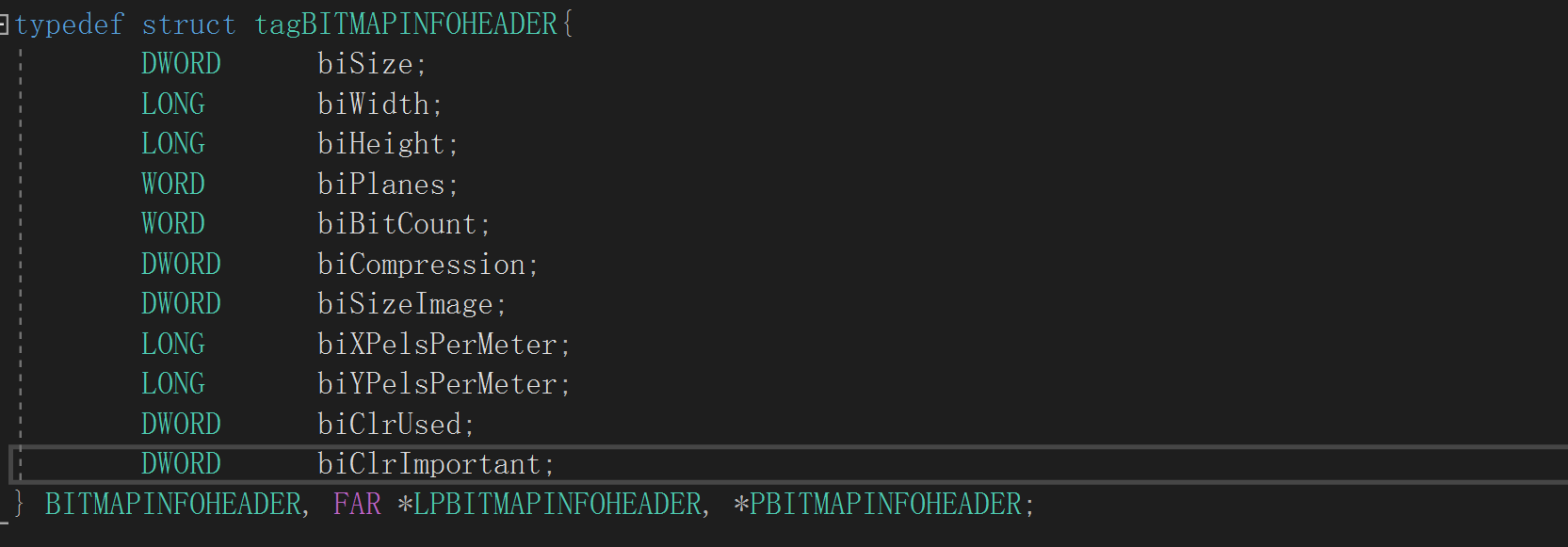
8、输出一张真彩图（Write a color bmp）

### 二、实验内容和原理（必填）

1、位图图像BMP（24bit）的组成

1）位图文件头BITMAPFILEHEADER

包含 BMP图像文件的类型、显示内容等信息

 2）位图信息头BITMAPINFOHEADER

包含有BMP图像的宽、高压缩方法，以及定义颜色等信息

3）调色板Palette（24位真彩色图可不用）

4）实际的位图数据ImageData

这部分的内容根据位图使用的位数不同而不同，在24位图中直接使用RGB

2、YUV与RGB之间的转换

Y = 0.299\*R + 0.587\*G + 0.114\*B;

U = -0.147\*R - 0.289\*G + 0.436\*B;

V = 0.615\*R - 0.515\*G - 0.100\*B;

R = Y + 1.14\*V;

G = Y - 0.39\*U - 0.58\*V;

B = Y + 2.03\*U;

### 三、源代码与分析

1、项目框架

main.c：主函数

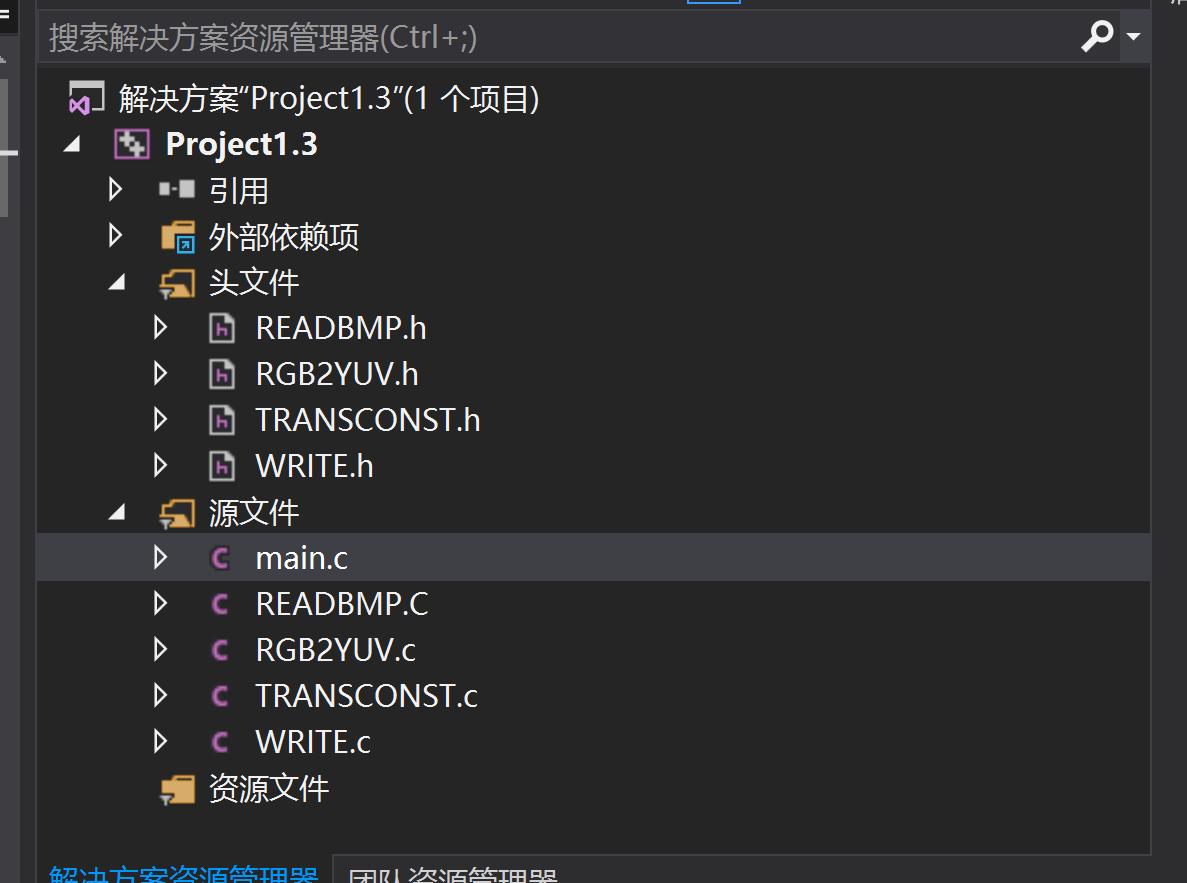
READBMP. c：读入bmp图像的rgb值

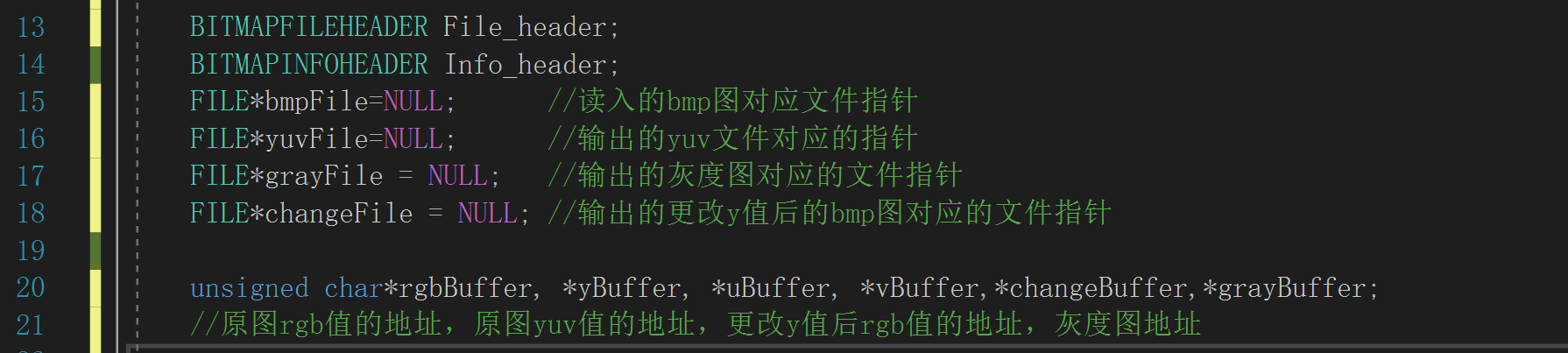
RGB2YUV.c：实现rgb->yuv转换

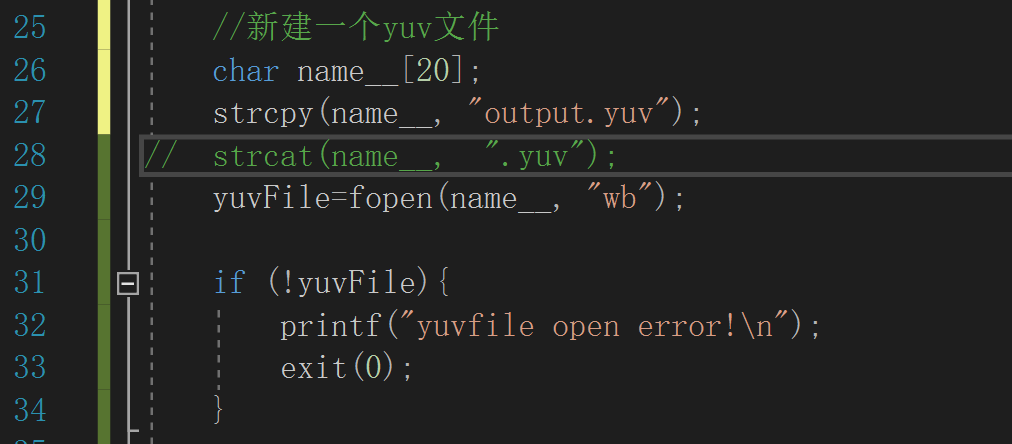
TRANSCONST.c：rgb->yuv过程中存在的常数参数

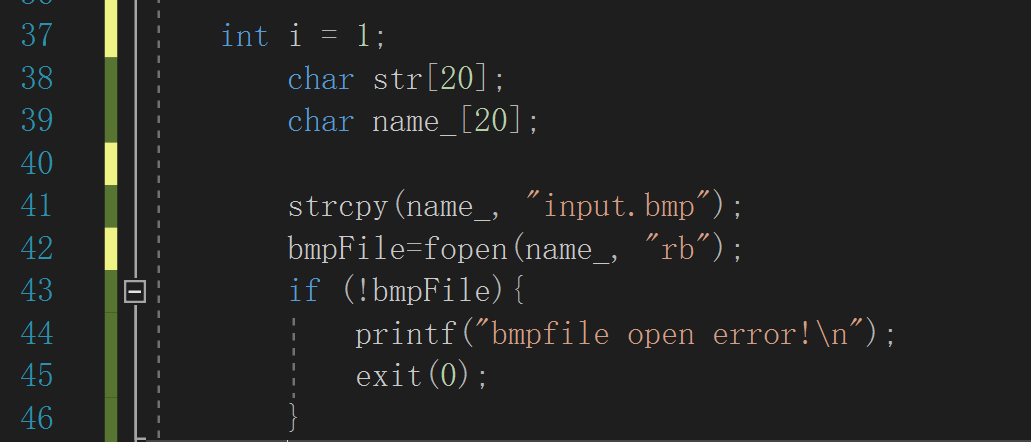
WRITE.c：将rgb值输出，产生新的灰度图与bmp图

对应存在各个.h头文件，进行对函数的声明

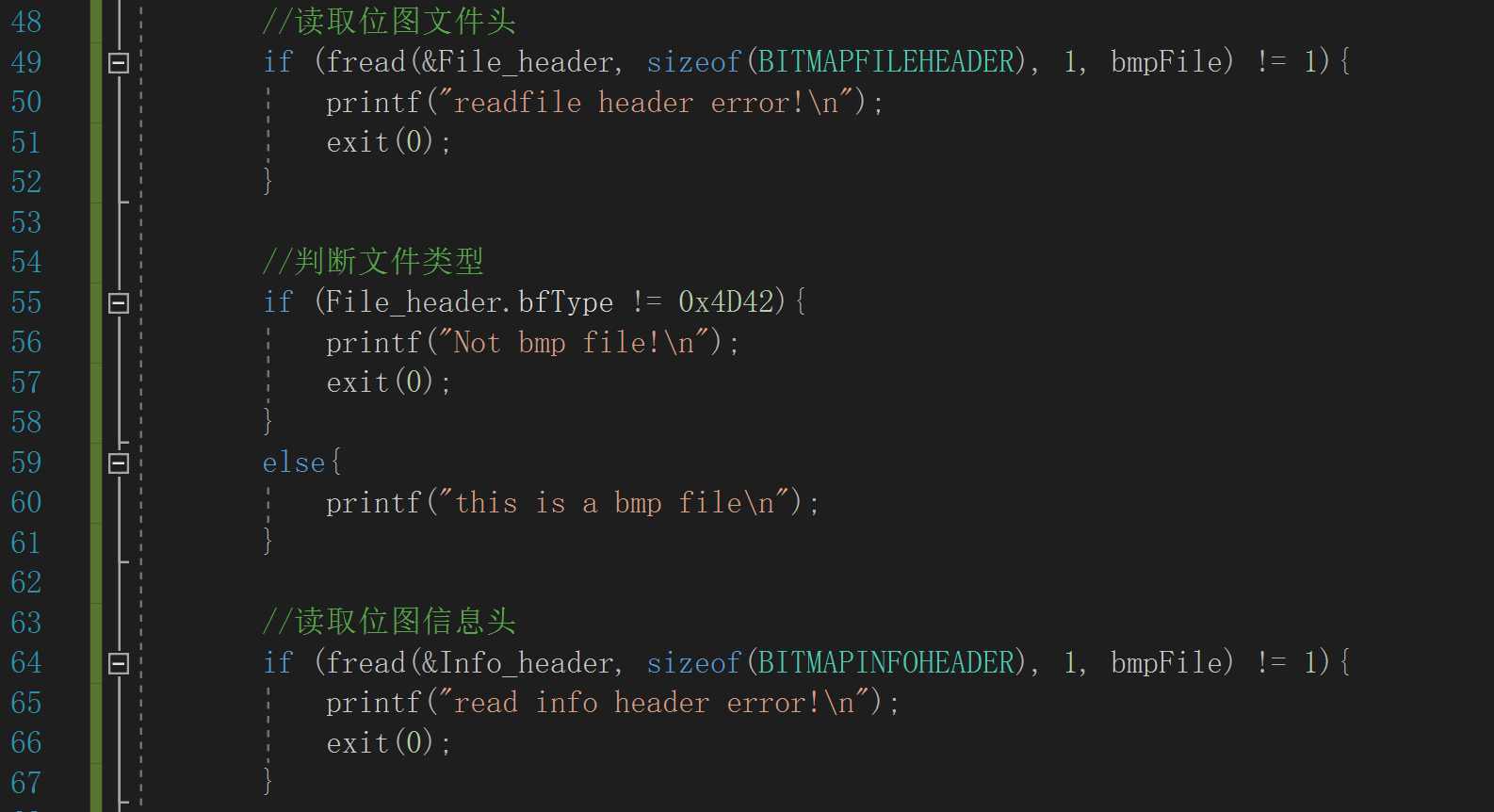
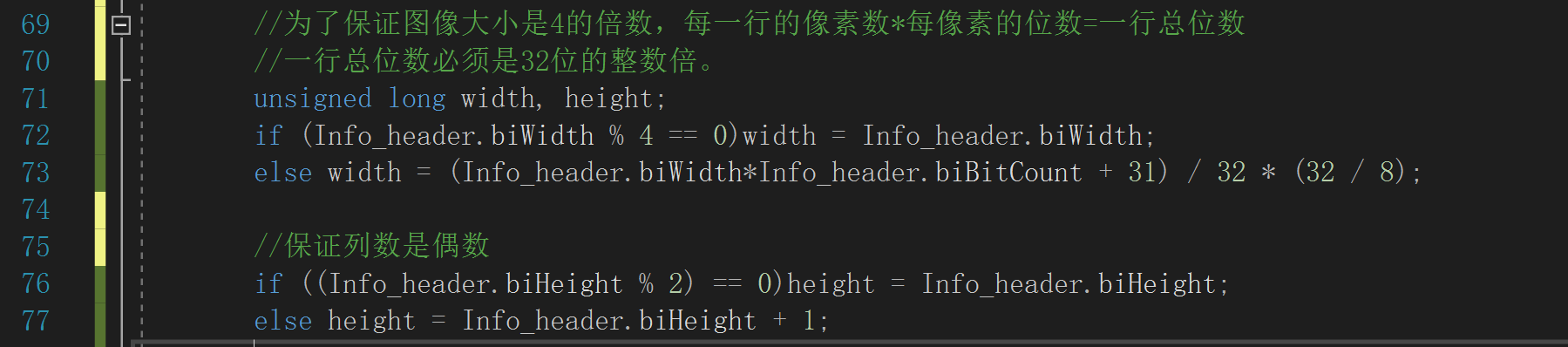
2、主函数代码分析

 1）变量定义

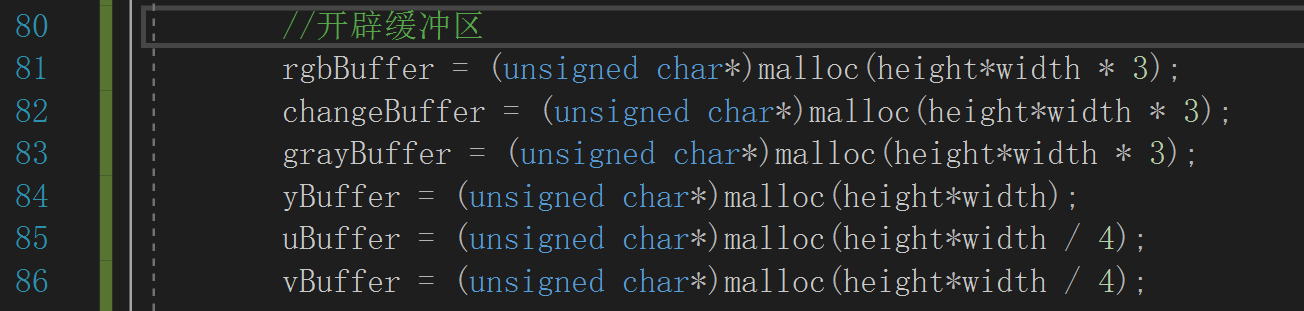
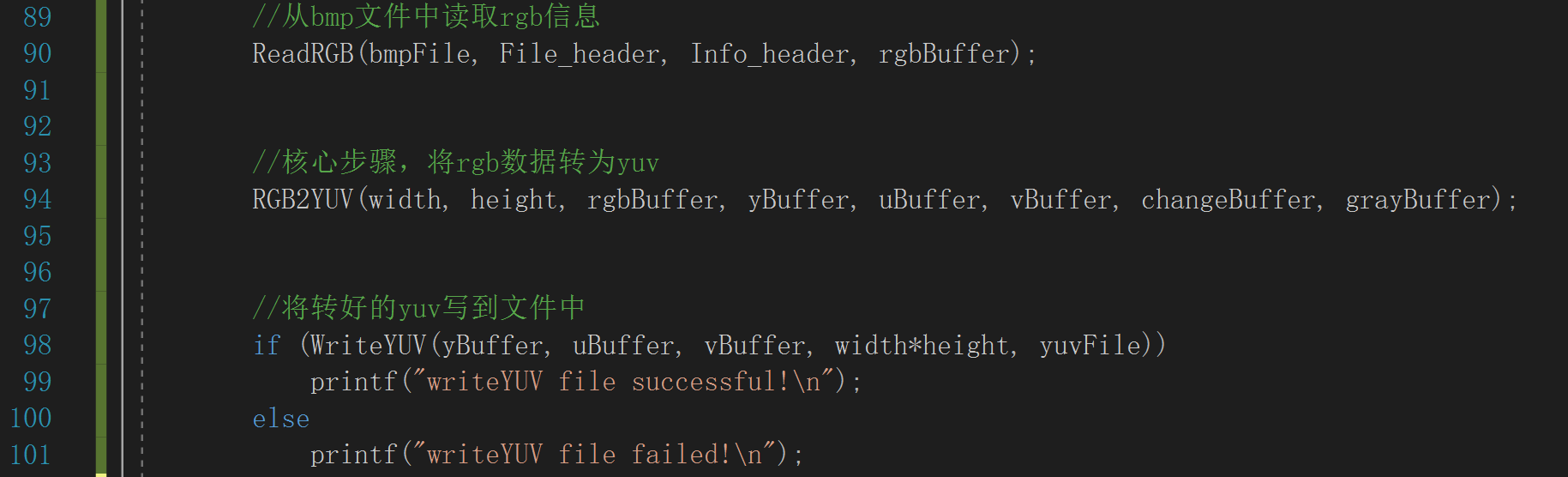
 2）新建input.yuv文件

 3）打开input.bmp文件

4）读取bmp文件文件头与信息头

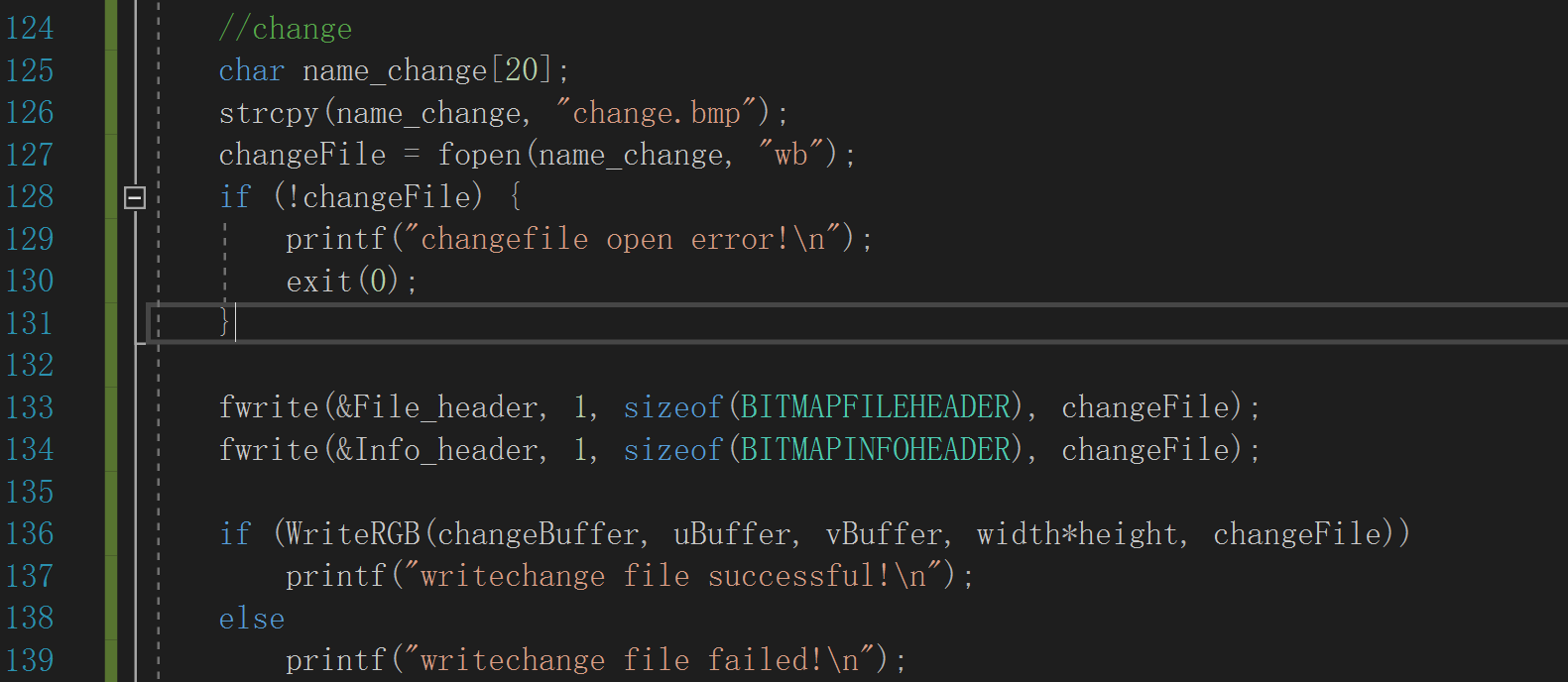
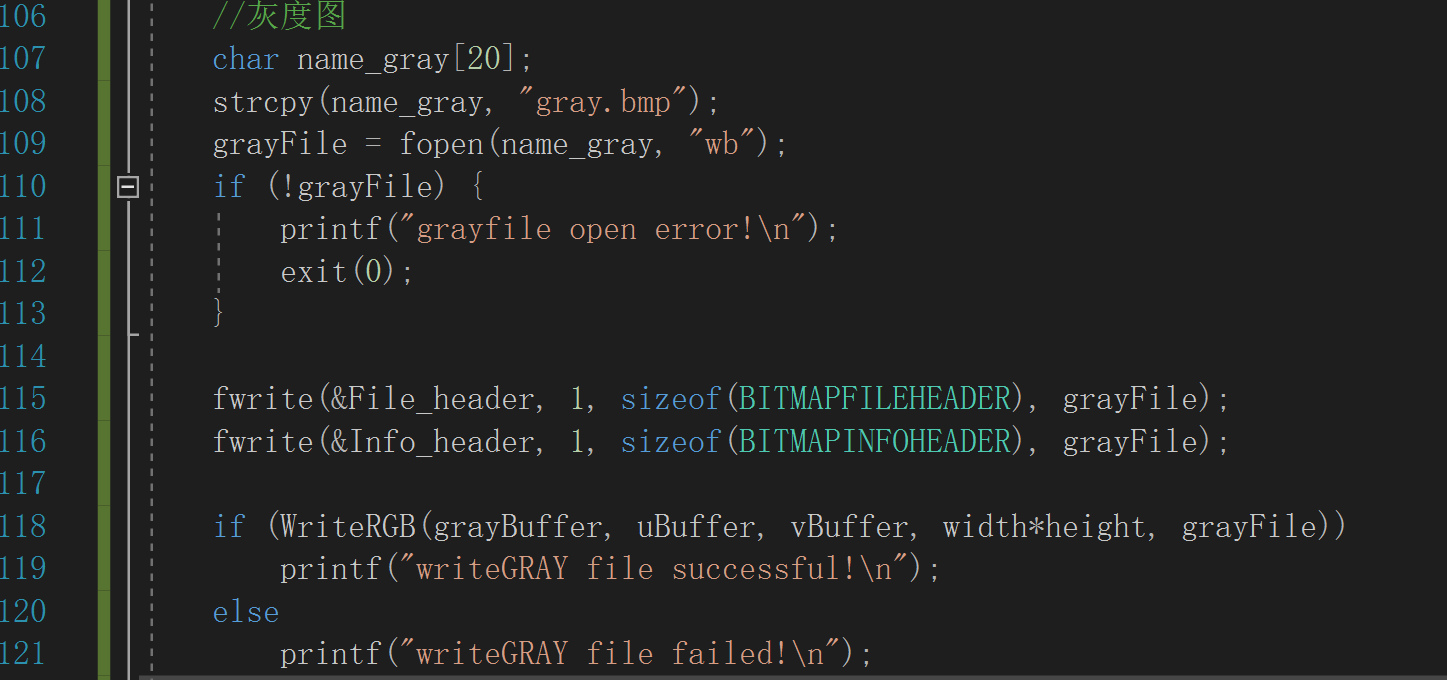
 5）对位图的尺寸进行修正

6）开辟储存信息的缓冲区

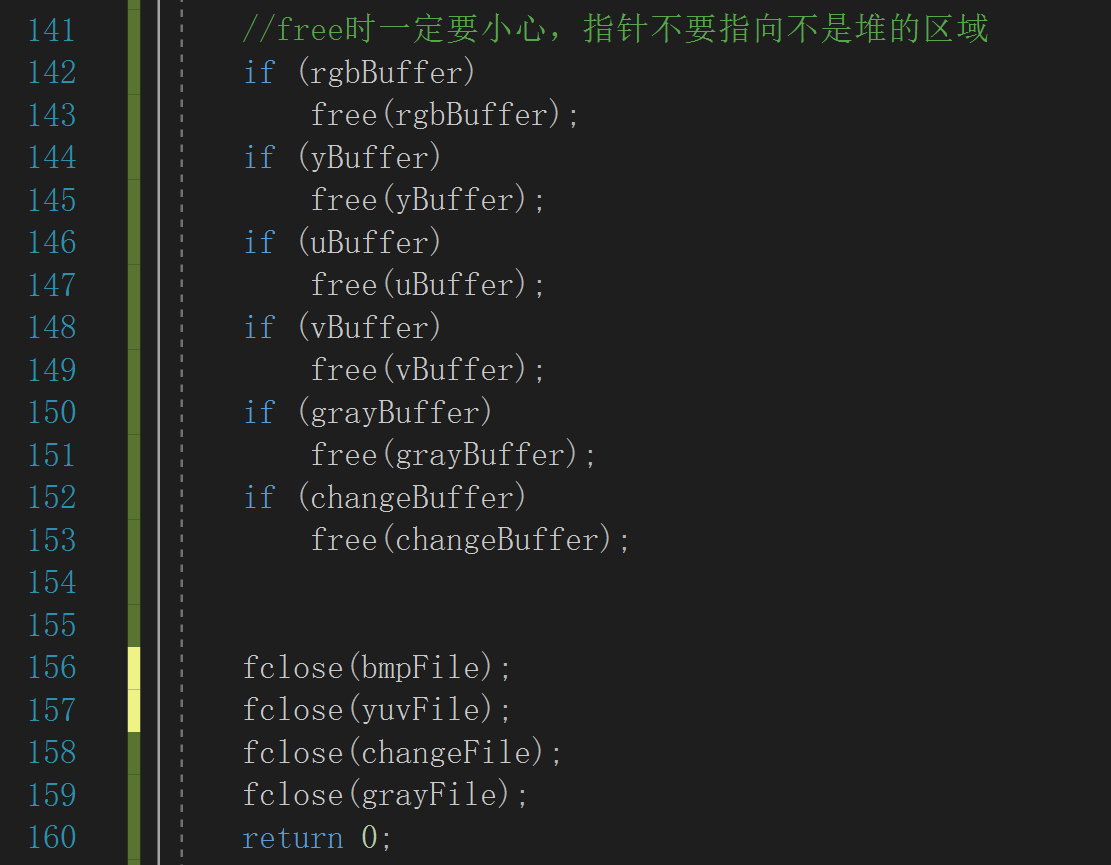
 7）调用函数

调用函数详细代码见下详解

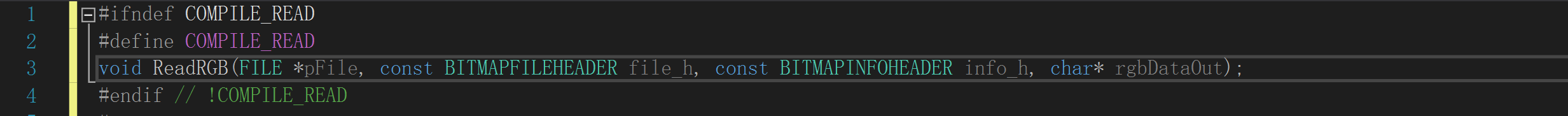
8）输出灰度图gray.bmp

输出修改y值后的图change.bmp

9）释放内存，关闭文件

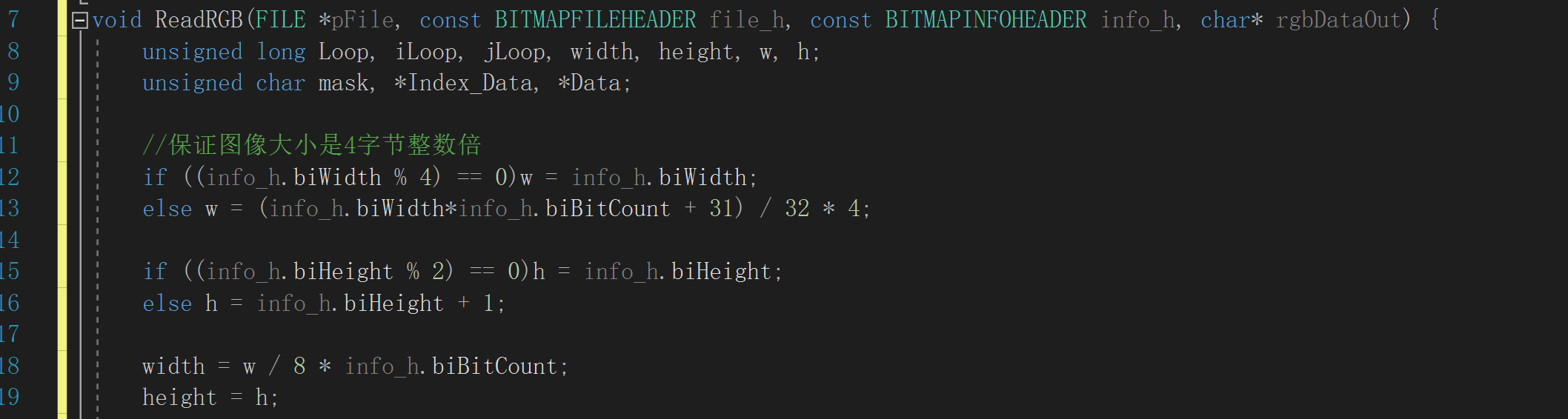


3、关于READBMP.c

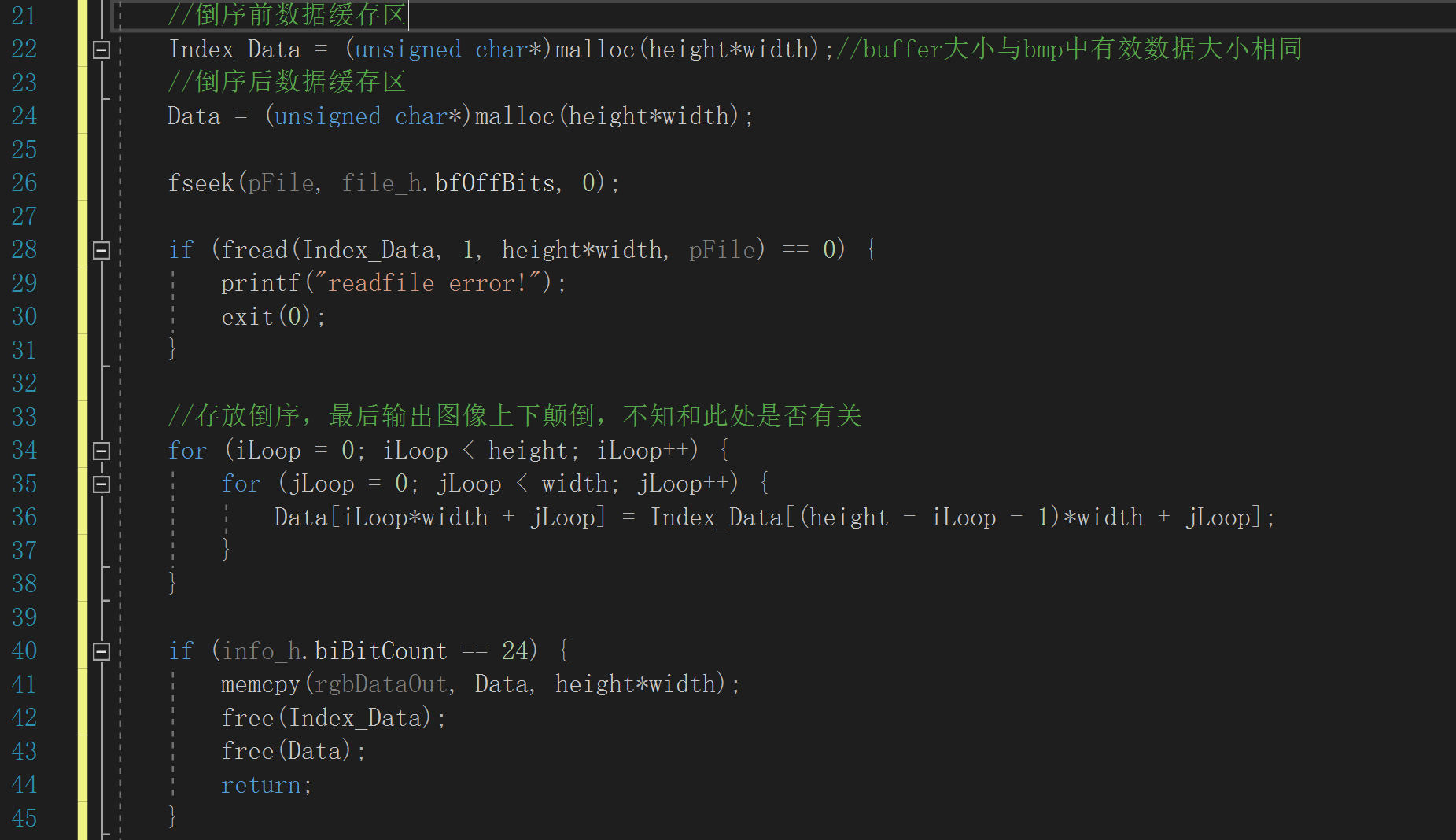


（对应的READBMP.h头文件）

1）修改图像尺寸，保证图像大小为4字节整数倍，否则uv值无法对应转换

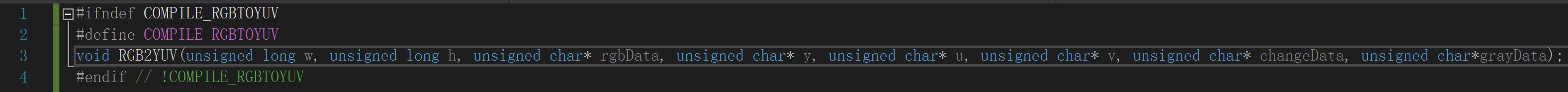


2）倒序读入图像rgb值，在bmp文件中颜色顺序按照每个像素bgr储存

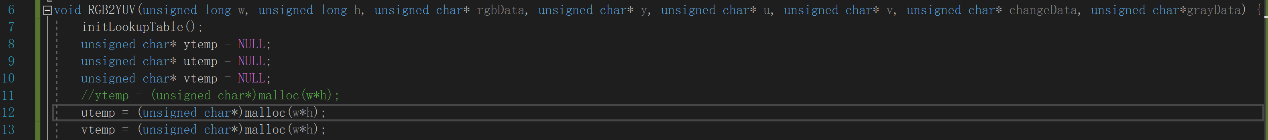
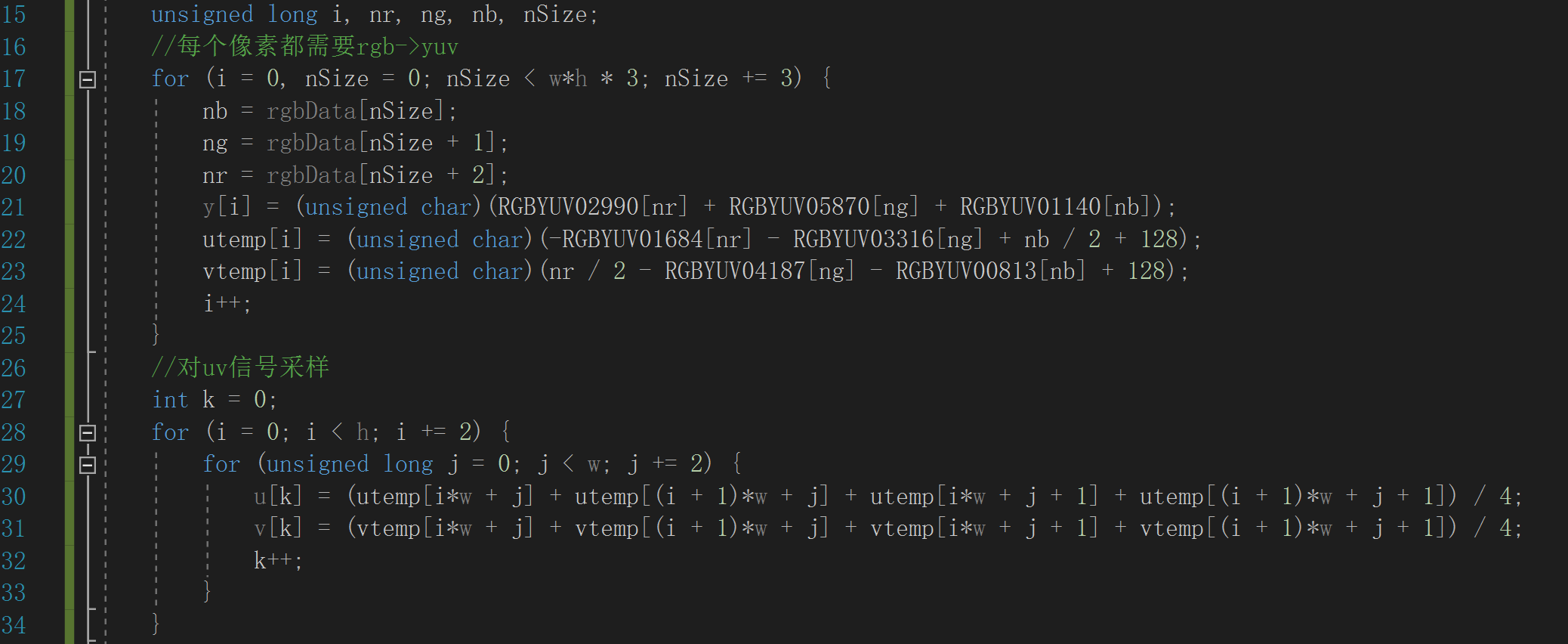
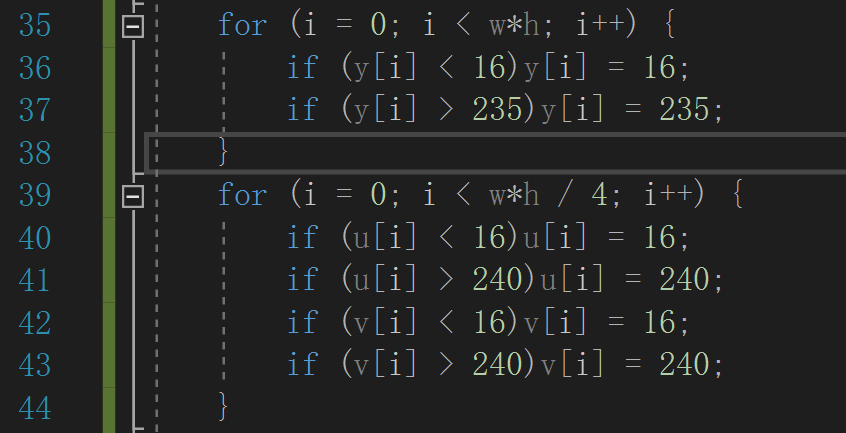


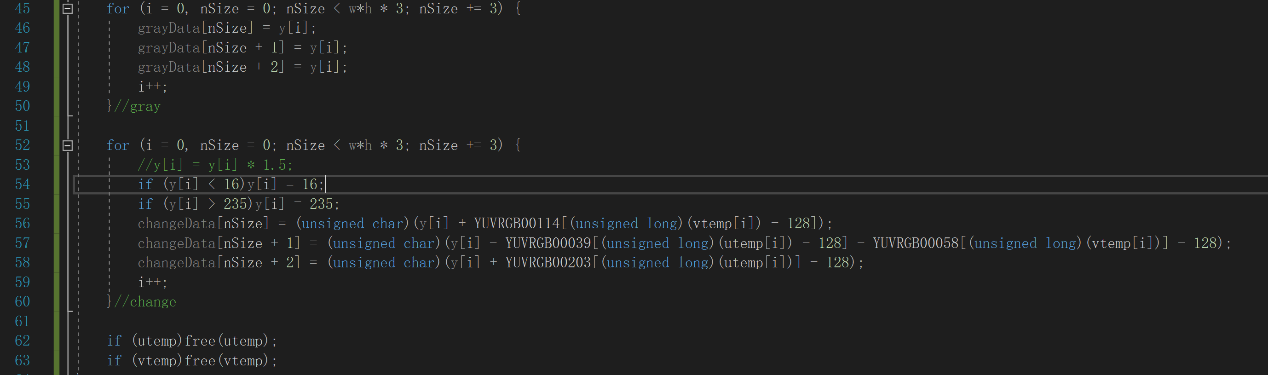
3）若非24位图，此处还需要添加调色板结构对图像进行解析

4、关于GRB2YUV.c

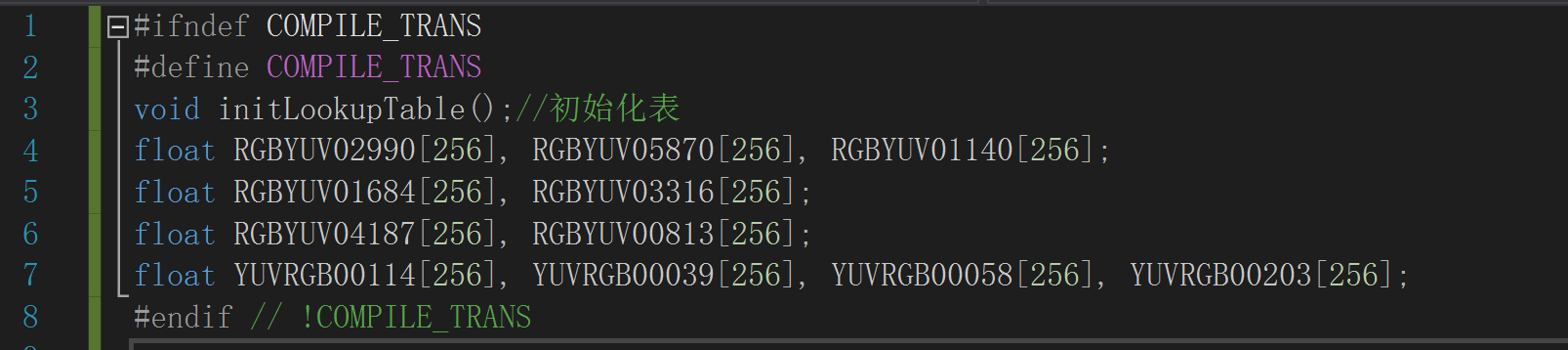


（对应GRB2YUV.h头文件）

1. 建立对各个像素点yuv值的指针，分配空间
2. 对每个像素进行rgb->yuv转换，对uv值进行调整
3. 控制yuv三值在要求范围内

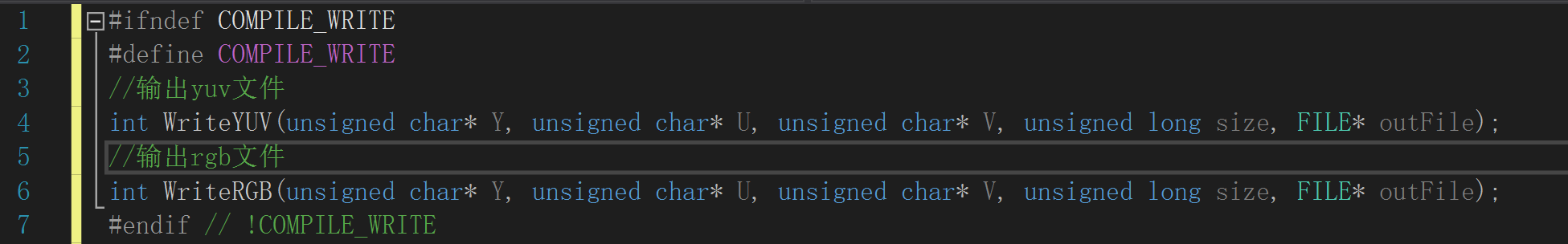
4）返回灰度图的rgb，更改y值后返回rgb，释放储存空间

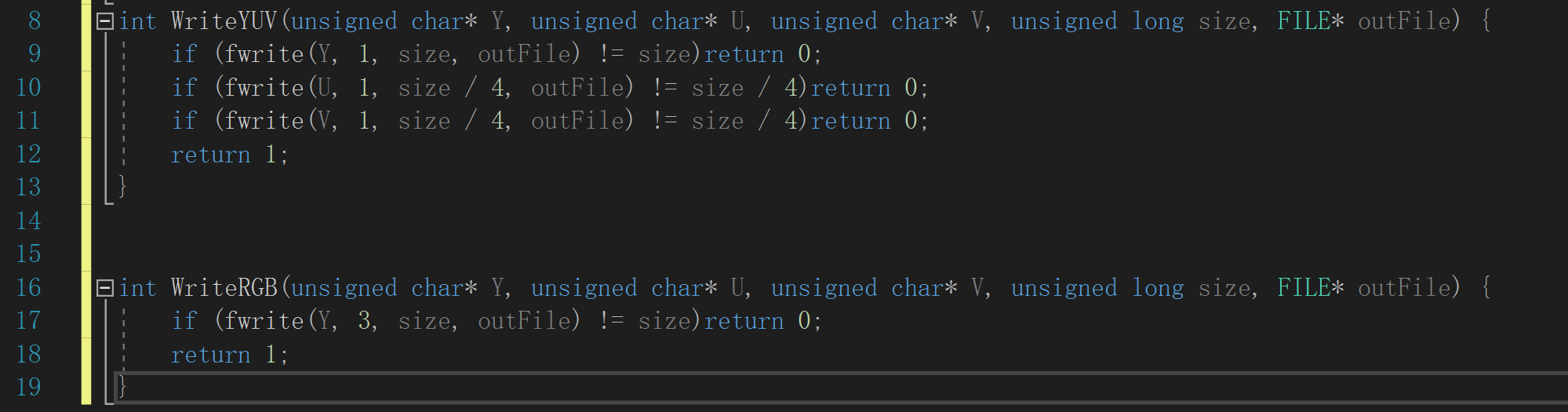
5、关于TRANSCONST.c

（对应TRANSCONST.h头文件）

用于定义在转换过程中所需的常数

6、关于WRITE.c

（对应WRITE.h头文件）



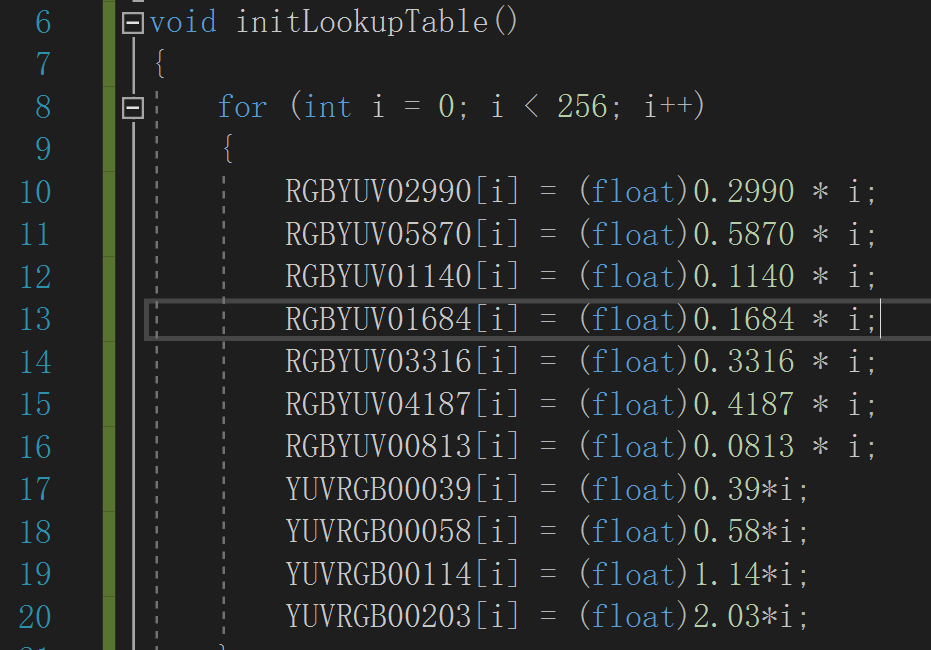
### 四、心得体悟

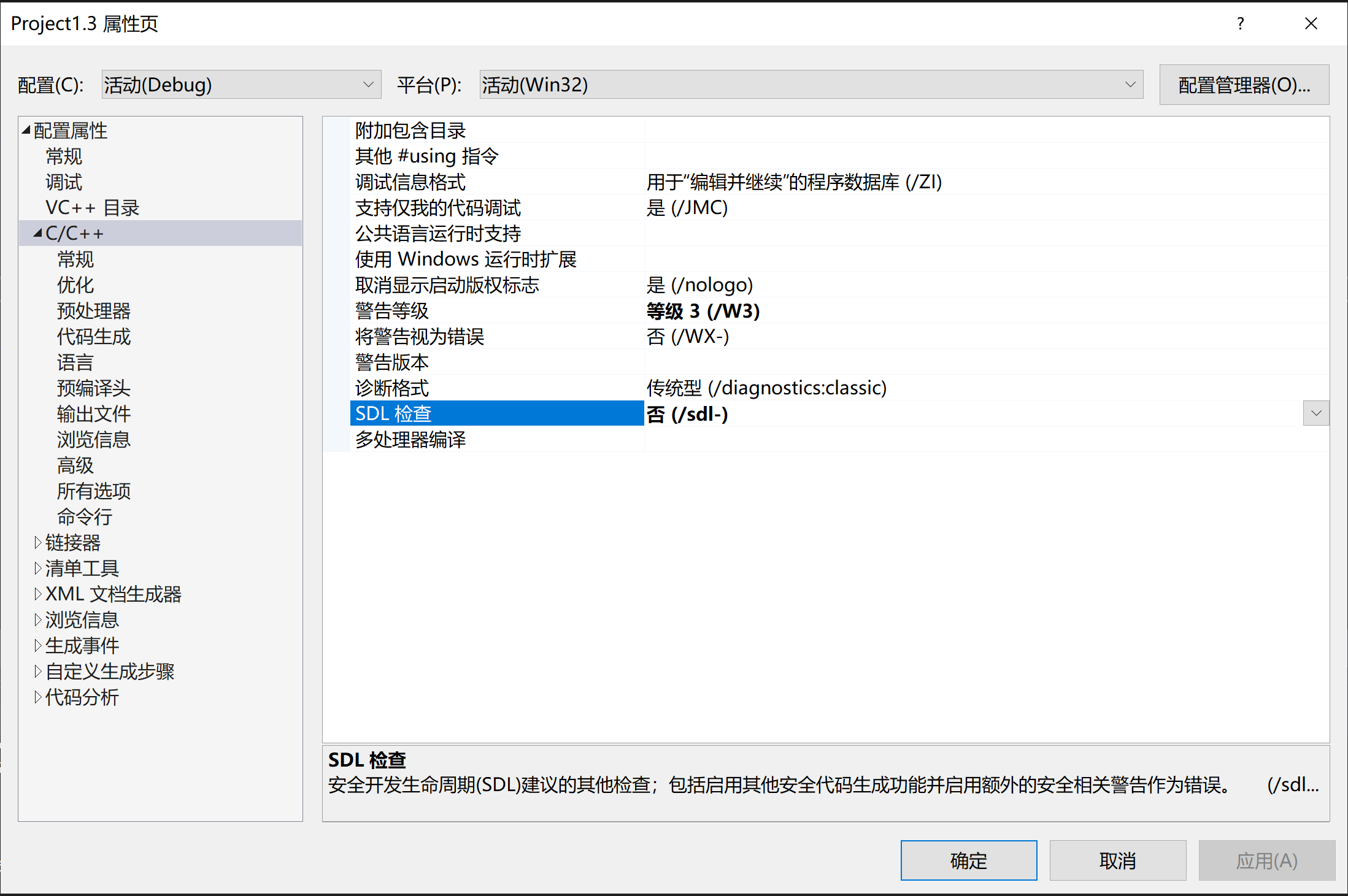
1、对于文件的读写修改有了更深的认识

存在问题：fopen等文件函数在vs2017中可能存在unsafe报错

解决方式： 1）修改函数为fopen\_s

2）如图修改SDL检查为否





2、bmp文件储存与我日常认知的文件储存方式不太一样，通过这次作业我对于bmp的文件格式有了更深的了解

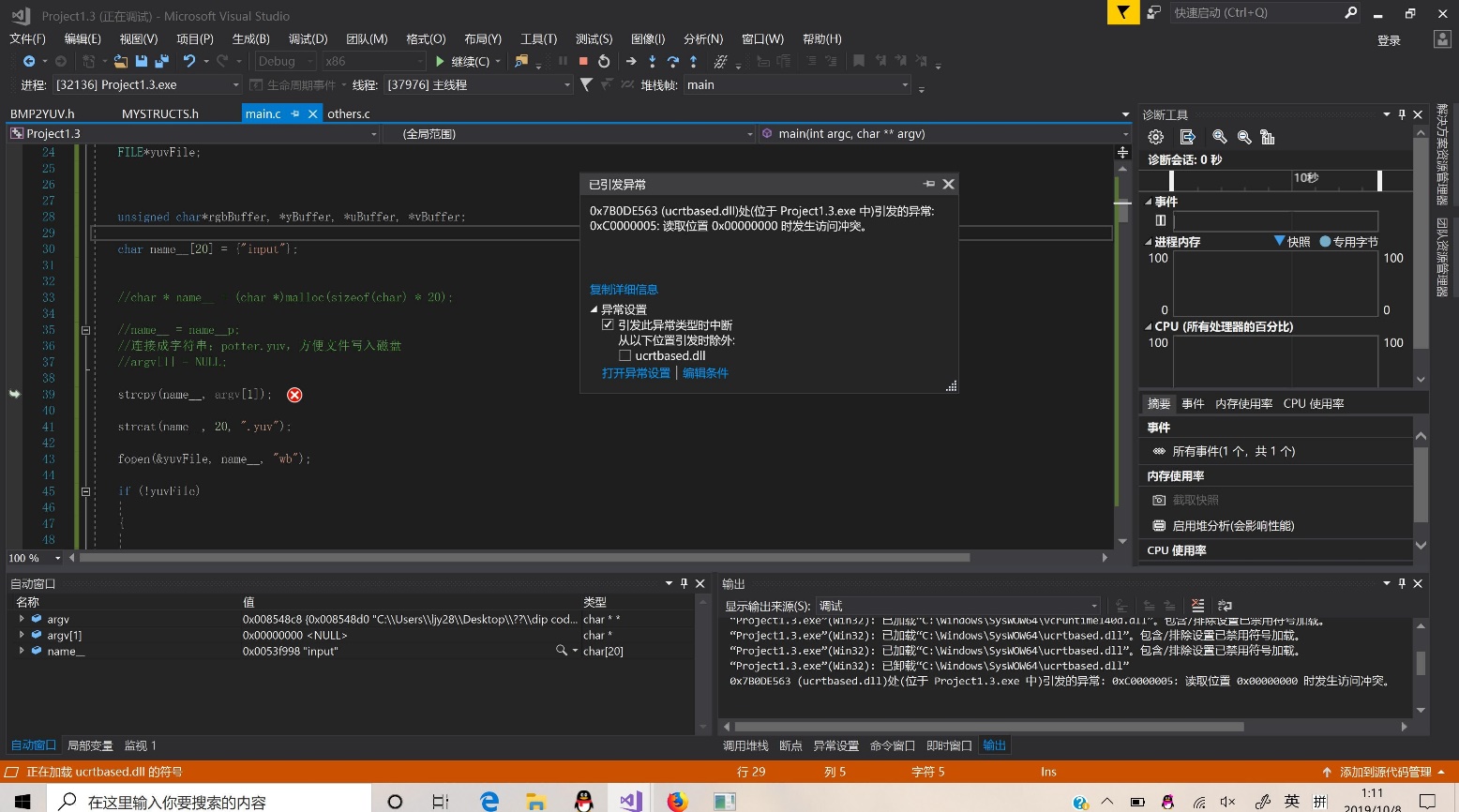
存在问题：bmp文件的rgb值无法正常读写

解决方案：关注rgb值的储存顺序，倒序读入

存在问题：输出的bmp文件图像上下颠倒

解决方案：尚未解决

3、文件读入时可能出现报错



解决方案：回归基础，重新学习数组指针赋值

结论：我太差了

4、拖ddl有罪，拖ddl该死