

Bmp 反色程序

1. 概述

本程序能读入 **bmp** 图片，修改后，另存为另一个图片。具体修改内容为把 **bmp** 的下半部分所有颜色反转。通过写这个程序，我学习到了 **c** 语言操作二进制文件，以及图片格式基础。本程序参考了 **github** 开源图片操作代码

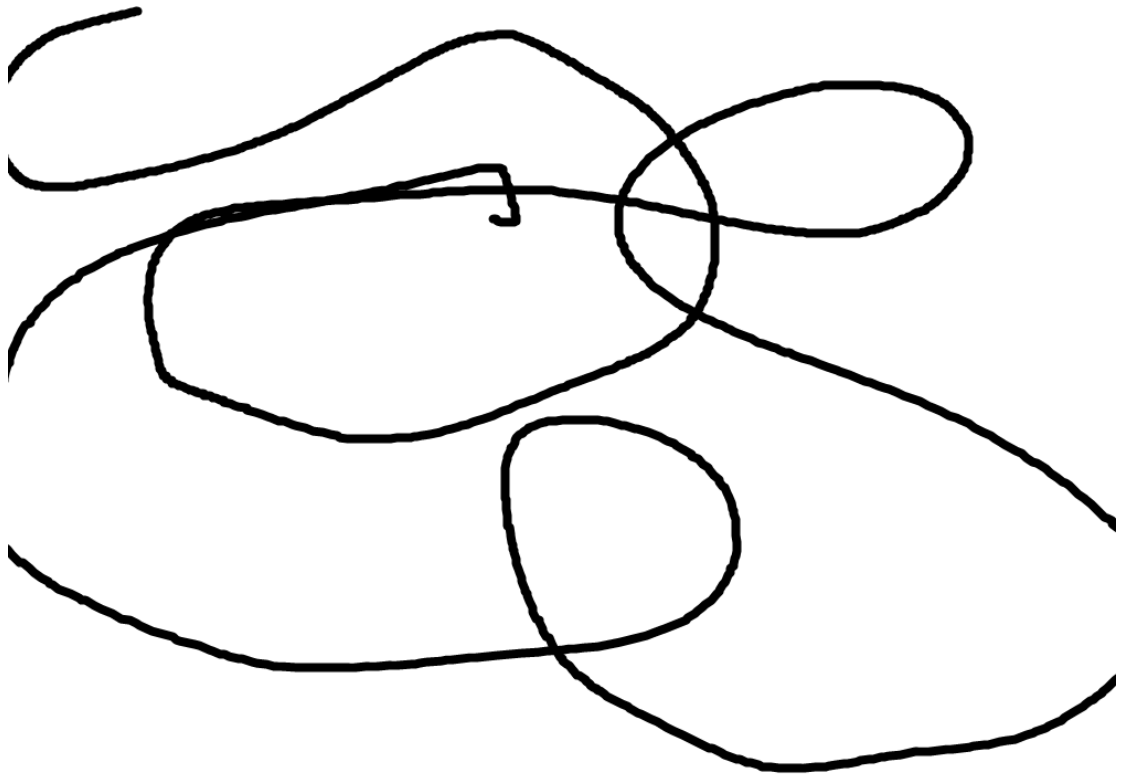
<https://github.com/edimetia3d/simpleBMP>。但反色功能为我自己所写。

2. 评分点说明

- a) 功能：读入 **in.bmp**，反色修改后，另存为 **out.bmp**。
- b) 结构：
 - a) 顺序结构： **Main** 函数从上到下即为顺序结构
 - b) 选择分支结构： **main** 函数立首先判断 **in.bmp** 是否存在，不存在则提示用户并退出程序。如果存在则继续运行后续步骤。程序里还有其他 **if** 语句。
 - c) 循环结构：反色过程为双重 **for** 循环。也就是对每个像素的遍历，每个像素由长和宽唯一决定，另外每个像素由 3 个字节组成，以此是 **RGB** 中的一个整数。每个字节范围是 0-255，255-原来的数，既是颜色反转。**For** 循环的高的循环终点值是图片高的一半，是为了方便打开输出图片后对比效果。
- c) 包含数组：程序中图片读入内存后，**bmp.imgData** 就是一个一维数组，每个元素表示图片的一个像素。
- d) 包含函数：程序中 **LoadImage** 和 **SaveImage** 是典型的函数，我自己也写了个函数 **exists** 判断文件是否存在。此外 **main** 函数也是特殊的函数。
- e) 包含指针：程序里对文件的操作包含 **FILE** 指针，对图片的读写更包含大量指针。
- f) 结构体：**CBitmapFileHeader** 和 **CBitmapInfoHeader** 都是结构体，是图片头信息在内存中的表现，比如包含图片的长宽等信息。具体字段全部看明白的话需要参考 **bmp** 的官方标准。

3. 运行截图：

- a) 输入文件：



b) 运行界面（分别是正常情况和异常情况）：

i.

```
成功打开in.bmp!宽度=1024      高度=768      深度=3  
正在把一半反色!  
成功反色保存到out.bmp!请按任意键继续. . .
```

ii.

```
需要在exe旁边放置in.bmp!  
请按任意键继续. . .
```

c) 输出文件：

