数字图像处理作业-第3章

计创18 181002222 连月菡

-目录-

一、作业描述

编写一个VC++的控制台程序。要求如下：

一、打开一个BMP图像文件（图像大小不要太大）。
二、显示图像文件的数据。
对于读入的图像文件，显示位图数据(选择一部分区域显示即可)。用256色或256级灰度图像进行验证。有两种显示方式：从上到下，从下到上。

三、计算图像的直方图(可以只显示灰度级100~200的直方图，为0的不显示；大小的处理，先归一化，再乘以50)
对于256级灰度级的图像进行计算。存入一个256元素的数组中，并显示其数据。用图形的方式显示（数据为0的不显示，如果值太大，按比例缩小一下）。如下。

四、图像的增亮或减暗
输入一个数字，如果是正数对图像进行增亮，如果是负数对图像进行减暗。
用其他软件进行查看，源图像和被增亮或减暗的图像进行对比，是不是达到了预想的效果。

五、main() 函数的要求

> 有选择。
> main()函数将指定BMP文件读入内存，将图像信息打印输出，最后又原样存入指定文件中。并进行计算需要的数据。
> 界面：
> 0----------结束
> 1----------读图像（8/24位）
> 2----------写图像（8/24位）
> 3----------显示图像数据

> 4----------计算直方图（100-200之间的灰度，1-10行，1-0列,10*10的大小）(8位)
> 5----------对图像进行增亮或减暗（如加上50或减去50等）(8位)
> 6----------图像反色
> 7----------变成灰度图（8/24位）
> 8----------显示调色板(8位)
> 9----------24位彩色图像转换为8位灰度图像
> 10----------24位彩色图像转换为8位彩色图像
>
> input choice:

要有操作提示。
功能 7-10 选做。

对于BMP文件的读写函数仅针对灰度图像（biBitCount=8）和彩色图像（biBitCount=24）两种格式，对于其他如biBitCount=1的图像类型，读者可以根据需要，自己对程序作简单的修改即可实现。

二、代码描述

本代码实现了0-8的功能。

分为3个部分:

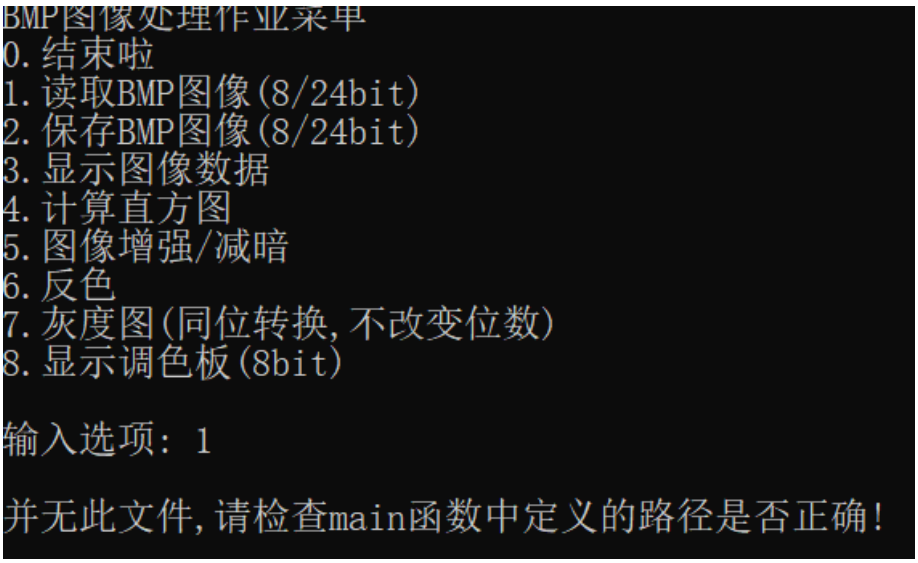
- 第一部分 bitmap文件结构的定义
- 第二部分 9个功能函数的定义
- 第三部分 主函数进行功能选择

在同一个文件夹中,**附有**对应的.cpp文件。

运行时需注意:main()函数和对应功能函数里 图像文件路径需要按照用户电脑存储的路径进行修改,否则会报错。

三、运行结果

报错提示:



表示路径错误,需要查看main()函数中设置的inputFile路径是否在用户的电脑上。

0.结束

```
输入选项：0

感谢你的使用！
```

1.读取BMP图像

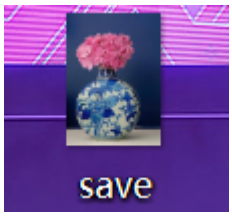
```
输入选项：1

宽度=265                长度=374
读取成功！
```

输出宽度和长度,表示读取成功。

2.保存BMP图像

```
输入选项：2

保存成功！
```

查看对应路径下是否有save.bmp文件。如果有,则运行正常。



3.显示图像数据

```
输入选项：3

109     61      37      52      52      52      52      52      52      52
52      52      52      52      52      0       52      52      52      52
52      52      52      52      56      0       0       0       52      52
52      52      52      52      52      52      28      3       0       0
52      52      52      52      52      52      52      52      1       0
0       0       52      52      52      52      52      52      52      52
3       0       0       0       52      52      52      52      52      52
52      52      27      3       0       0       52      52      52      52
52      52      52      52      118     1       0       0       52      52
52      52      52      52      52      52      9       1       0       0
显示完毕！
```

输出 10*10数据内容。

4.计算直方图

```
输入选项：4

像素数据为：

归一化后的直方图数据为：
  0     0     0     0     0    47   106    29    74    38    13    47   109    60    44    92
 28    32    63     0     0     0     0     0     0     0     0     0    66    66    65    65
 63    63    64    64    64    64    63    62    63    62    62    61    64    65    65    65
 65    65    65    65    62    62    62    62    62    62    62    62    63    62    61    61
 60    60    60    61    59    59    59    59    59    59    59    59    62    62    62    61
 60    60    59    59    56    57    57    58    58    57    57    56    54    53    53    53
 52    52    51    51    50    50    50    51    51    51    51    52    51    51    51    51
 51    51    51    51    53    53    54    54    53    52    52    51    52    50    49    49
 51    53    53    52    50    50    49    49    49    50    51    52    52    51    51    50
 51    52    53    53    51    51    51    51    51    52    53    53    53    53    53    53
 53    53    53    53    54    54    54    54    54    54    54    54    54    54    54    54
 54    54    54    54    56    56    56    56    56    56    56    56    56    56    56    56
 57    57    58    58    57    57    58    58    59    60    60    60    59    59    59    59
 60    60    61    61    63    62    62    64    66    64    60    55    62    62    63    63
 64    65    65    65    65    65    65    65    65    65    65    65    64    64    64    64
 64    64    64    64    65    65    65    65    65    65    65    65    64    64    64    64
```

```
13 :*
28 :*
29 :*
32 :*
38 :*
44 :*
47 :**
49 :*****
50 :********
51 :************************
52 :***********
53 :********************
54 :******************
55 :*
56 :**************
57 :********
58 :******
59 :****************
60 :************
61 :*******
62 :********************
63 :**********
64 :**********************
65 :****************************
66 :***
74 :*
92 :*
106 :*
109 :*
```

5.图像增强/减暗

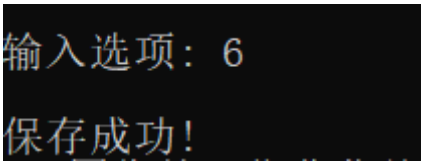
```
输入选项： 5

1. 增强
2. 减暗
输入你的选择：
1
保存成功！
```

查看对应路径下是否有intense.bmp文件。如果有,则运行正常。

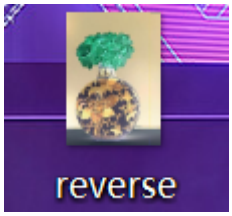
```
输入选项： 5

1. 增强
2. 减暗
输入你的选择：
2
保存成功！
```

查看对应路径下是否有reduce.bmp文件。如果有,则运行正常。

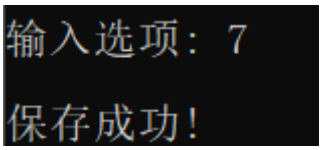
6.反色



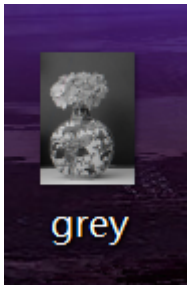
查看对应路径下是否有reverse.bmp文件。如果有,则运行正常。



7.灰度图



查看对应路径下是否有grey.bmp文件。如果有,则运行正常。



8.显示调色板(8bit)

如果该图是24bit BMP文件,则显示:

如果该图是8bit BMP文件,则显示调色板内容:

```
输入选项: 8

  0   0   0       0   0   0     128   0   0     128   0   0
  0 128 128       0 128   0       0   0 128       0 128   0
128 128   0       0 192 192     192   0 192     220 192   0
240 202 166       0   0  32      64   0   0      32  96   0
  0  32 128       0   0  32     160   0   0      32 192   0
  0  32 224       0   0  64       0   0   0      64  32   0
  0  64  64       0   0  64      96   0   0      64 128   0
  0  64 160       0   0  64     192   0   0      64 224   0
  0  96   0       0   0  96      32   0   0      96  64   0
  0  96  96       0   0  96     128   0   0      96 160   0
  0  96 192       0   0  96     224   0   0     128   0   0
  0 128  32       0   0 128      64   0   0     128  96   0
  0 128 128       0   0 128     160   0   0     128 192   0
  0 128 224       0   0 160       0   0   0     160  32   0
  0 160  64       0   0 160      96   0   0     160 128   0
  0 160 160       0   0 160     192   0   0     160 224   0
  0 192   0       0   0 192      32   0   0     192  64   0
  0 192  96       0   0 192     128   0   0     192 160   0
  0 192 192       0   0 192     224   0   0     224   0   0
  0 224  32       0   0 224      64   0   0     224  96   0
  0 224 128       0   0 224     160   0   0     224 192   0
  0 224 224       0  64   0       0   0  64       0  32   0
 64   0  64       0  64   0      96   0  64       0 128   0
 64   0 160       0  64   0     192   0  64       0 224   0
 64  32   0       0  64  32      32   0  64      32  64   0
 64  32  96       0  64  32     128   0  64      32 160   0
 64  32 192       0  64  32     224   0  64      64   0   0
 64  64  32       0  64  64      64   0  64      64  96   0
 64  64 128       0  64  64     160   0  64      64 192   0
 64  64 224       0  64  96       0   0  64      96  32   0
 64  96  64       0  64  96      96   0  64      96 128   0
 64  96 160       0  64  96     192   0  64      96 224   0
 64 128   0       0  64 128      32   0  64     128  64   0
 64 128  96       0  64 128     128   0  64     128 160   0
 64 128 192       0  64 128     224   0  64     160   0   0
 64 160  32       0  64 160      64   0  64     160  96   0
 64 160 128       0  64 160     160   0  64     160 192   0
 64 160 224       0  64 192       0   0  64     192  32   0
 64 192  64       0  64 192      96   0  64     192 128   0
 64 192 160       0  64 192     192   0  64     192 224   0
 64 224   0       0  64 224      32   0  64     224  64   0
 64 224  96       0  64 224     128   0  64     224 160   0
 64 224 192       0  64 224     224   0 128       0   0   0
128   0  32       0 128   0      64   0 128       0  96   0
128   0 128       0 128   0     160   0 128       0 192   0
128   0 224       0 128  32       0   0 128      32  32   0
128  32  64       0 128  32      96   0 128      32 128   0
128  32 160       0 128  32     192   0 128      32 224   0
128  64   0       0 128  64      32   0 128      64  64   0
128  64  96       0 128  64     128   0 128      64 160   0
128  64 192       0 128  64     224   0 128      96   0   0
128  96  32       0 128  96      64   0 128      96  96   0
128  96 128       0 128  96     160   0 128      96 192   0
128  96 224       0 128 128       0   0 128     128  32   0
128 128  64       0 128 128      96   0 128     128 128   0
128 128 160       0 128 128     192   0 128     128 224   0
128 160   0       0 128 160      32   0 128     160  64   0
128 160  96       0 128 160     128   0 128     160 160   0
128 160 192       0 128 160     224   0 128     192   0   0
128 192  32       0 128 192      64   0 128     192  96   0
128 192 128       0 128 192     160   0 128     192 192   0
128 192 224       0 128 224       0   0 128     224  32   0
128 224  64       0 128 224      96   0 128     224 128   0
128 224 160       0 128 224     192   0 128     224 224   0
显示完毕!
```

四、作业代码

```

#include <stdio.h>
#include<cmath>
#include <cstdint>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <vector>
#include <fstream>
#include <cmath>
#include <map>
/**********************************
*====================*
*==BMP文件结构定义开始===*
*====================*
**********************************/
typedef unsigned char BYTE;
typedef unsigned short WORD;
typedef unsigned int DWORD;
typedef int LONG;

#pragma pack(push, 1)
typedef struct tagBITMAPFILEHEADER {
    WORD bfType;
    DWORD bfSize;
    WORD bfReserved1;
    WORD bfReserved2;
    DWORD bfOffBits;
} BITMAPFILEHEADER;

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;
#pragma pack(pop)

typedef struct tagRGBTRIPLE {
    BYTE rgbBlue;
    BYTE rgbGreen;
    BYTE rgbRed;
} RGBTRIPLE;
static FILE* inputFile;
static FILE* outputFile;
/**********************************

```

```
*====================*
*==BMP文件结构定义结束===*
*====================*
**********************************/

/*********************************
*====================*
*==各个功能函数定义开始===*
*====================*
**********************************/

int read(const char* input) {// 1.读取图像
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,
重新修改文件路径
    }
    char header[122];
    fread(&header, sizeof(char), 122, inputFile); //获得图像头部信息
    int width = *(int*)&header[18]; //获得图像宽度
    int height = abs(*(int*)&header[22]); //获得图像高度
    printf("宽度=%d\t长度=%d\n", width, height);//功能1实现
    return 1;//成功,返回1
}

int save(const char* input) {// 2.保存图像
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,
重新修改文件路径
    }

    outputFile = fopen("C:/Users/yuehan lian/Desktop/save.bmp", "wb");
    if (outputFile == NULL) {
        printf("输出图像文件创建失败!\n");//文件创建失败
        return 0;//函数执行失败,返回0
    }
    char header[122];
    fread(&header, sizeof(char), 122, inputFile); //获得输入图像头部信息
    fwrite(&header, sizeof(char), 122, outputFile);//写进输出图像头部信息

    int width = *(int*)&header[18]; //获得图像宽度
    int height = abs(*(int*)&header[22]);//获得图像高度
    int stride = ((4 * ((8 * 3) * width) + 31) / 32); //计算行的数量
    int paddingCheck = (stride % 4);
    int padding;
    if (paddingCheck != 0) { //如果行数不能被4整除
        int padCalc = 0;
        for (int i = 1; padCalc < stride; i++) {
            padCalc = i * 4;
        }
        padding = padCalc - stride;
    }
    else
        padding = 0;

    char pad;
    char pixel[3];
```

```
    for (int j = 0; j < height; j++) {
        for (int k = 0; k < width; k++) {
            fread(pixel, sizeof(char), 3, inputFile); //读取RBG信息,并写入pixel中
            fwrite(&pixel, (sizeof(char)), 3, outputFile);//pixel中的信息写入输出图像中
        }
        for (int l = 0; l < padding; l++) {
            fread(&pad, 1, 1, inputFile); //读取末尾填充的图像信息
            fwrite(&pad, 1, 1, outputFile); //向输出图像写入末尾填充的图像信息
        }
    }
    int elms_read; //确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
        else {
            fwrite(&pad, 1, 1, outputFile);
        }
    }
    fclose(inputFile);
    fclose(outputFile);//关闭文件~
    return 1;//执行成功返回1
}

int show(const char* input) {//3. 显示图像信息
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,重新修改文件路径
    }

    char header[54];
    fread(&header, sizeof(char), 54, inputFile);//获得输入图像头部信息

    int width = *(int*)&header[18];//获得图像宽度
    int height = abs(*(int*)&header[22]); //获得图像高度
    int stride = (4 * ((8 * 3) * width) / 32);//计算行的数量
    int paddingCheck = (stride % 4);
    int padding;
    if (paddingCheck != 0) { //如果行数不能被4整除
        int padCalc = 0;
        for (int i = 1; padCalc < stride; i++) {
            padCalc = i * 4;
        }
        padding = padCalc - stride;
    }
    else
        padding = 0;

    char pad;
    char pixel[3];
    for (int j = 0; j < height; j++) {
        for (int k = 0; k < width; k++) {
            fread(pixel, sizeof(char), 3, inputFile); //读取RBG信息,并写入pixel中
```

```
        }
        for (int l = 0; l < padding; l++) {//pixel中的信息写入输出图像中
            fread(&pad, 1, 1, inputFile);  //读取填充的图像信息
        }
    }
    int elms_read;//确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
    }

    for (int i = 0; i < 100; ++i)
    {
        printf("%d\t", abs((int)pixel[i]));
        if ((i + 1) % 10 == 0)
            printf("\n");
    }
    fclose(inputFile);
    return 1;
}

int reverse(const char* input) {// 6.图像反色
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,
重新修改文件路径
        return 0;//函数执行失败,返回0
    }
    outputFile = fopen("C:/Users/yuehan lian/Desktop/reverse.bmp", "wb");
    if (outputFile == NULL) {
        printf("输出图像文件创建失败!\n");//文件创建失败
        return 0;//函数执行失败,返回0
    }
    char header[54];
    fread(&header, sizeof(char), 54, inputFile); //获得输入图像头部信息
    fwrite(&header, sizeof(char), 54, outputFile);//写进输出图像头部信息

    int width = *(int*)&header[18];  //获得图像宽度
    int height = abs(*(int*)&header[22]);  //获得图像高度
    int stride = (4 * ((8 * 3) * width) / 32); //计算行的数量
    int paddingCheck = (stride % 4);
    int padding;
    if (paddingCheck != 0) { //计算行的数量
        int padCalc = 0;
        for (int i = 1; padCalc < stride; i++) {
            padCalc = i * 4;
        }
        padding = padCalc - stride;
    }
    else
        padding = 0;

    char pad;
    char pixel[3];
    for (int j = 0; j < height; j++) {
        for (int k = 0; k < width; k++) {
```

```
            fread(pixel, sizeof(char), 3, inputFile); //读取RBG信息,并写入pixel中
            //pixel[0] =~pixel[0];//进行取反运算,得到反色后的像素值
            //pixel[1] = ~pixel[1];
            //pixel[2] = ~pixel[2];
            pixel[0] = 255 - pixel[0];
            pixel[1] = 255 - pixel[1];
            pixel[2] = 255 - pixel[2];
            fwrite(&pixel, (sizeof(char)), 3, outputFile);//向输出图像写入末尾填充的图像信息
        }
        for (int l = 0; l < padding; l++) {
            fread(&pad, 1, 1, inputFile); //读取末尾填充的图像信息
            fwrite(&pad, 1, 1, outputFile);  //向输出图像写入末尾填充的图像信息
        }
    }
    int elms_read;  //确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
        else {
            fwrite(&pad, 1, 1, outputFile);
        }
    }
    fclose(inputFile);
    fclose(outputFile);
    return 1;
}

int greyScale(const char* input) {//7. 获得同一位深的灰度图像
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,重新修改文件路径
        return 0;//函数执行失败,返回0
    }

    outputFile = fopen("C:/Users/yuehan lian/Desktop/grey.bmp", "wb");
    if (outputFile == NULL) {
        printf("输出图像文件创建失败!\n");//文件创建失败
        return 0;//函数执行失败,返回0
    }

    char header[54];
    fread(&header, sizeof(char), 54, inputFile); //获得输入图像头部信息
    fwrite(&header, sizeof(char), 54, outputFile);//写进输出图像头部信息

    int width = *(int*)&header[18]; //获得图像宽度
    int height = abs(*(int*)&header[22]); //获得图像高度
    int stride = (4 * ((8 * 3) * width + 31) / 32);//计算行的数量
    int paddingCheck = (stride % 4);
    int padding;
    if (paddingCheck != 0) {//如果行数不能被4整除
        int padCalc = 0;
        for (int i = 1; padCalc < stride; i++) {
            padCalc = i * 4;
```

```
        }
        padding = padCalc - stride;
    }
    else
        padding = 0;

    char pad;
    unsigned char pixel[3];
    unsigned char pixelEdit[3];
    for (int j = 0; j < height; j++) {
        for (int k = 0; k < width; k++) {
            fread(pixel, sizeof(unsigned char), 3, inputFile); // 读取RBG信息, 并写入pixel中
            unsigned char avg = (pixel[0] + pixel[1] + pixel[2]) / (unsigned char)3;
            pixelEdit[0] = pixelEdit[1] = pixelEdit[2] = avg;//如果要变灰度图,RGB三个分量要相等

            fwrite(&pixelEdit, (sizeof(unsigned char)), 3, outputFile);//pixel中的信息写入输出图像中
        }
        for (int l = 0; l < padding; l++) {// 读取末尾填充的图像信息
            fread(&pad, 1, 1, inputFile);
            fwrite(&pad, 1, 1, outputFile); //向输出图像写入末尾填充的图像信息
        }
    }
    int elms_read;//确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
        else {
            fwrite(&pad, 1, 1, outputFile);
        }
    }
    fclose(inputFile);
    fclose(outputFile);//关闭文件~
    return 1;//执行成功返回1
}

int intense(const char* input) {//5.图像增强或减暗
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,重新修改文件路径
    }

    char header[122];
    fread(header, sizeof(char), 122, inputFile); //获得输入图像头部信息

    int width = *(int*)&header[18]; //获得图像宽度
    int height = abs(*(int*)&header[22]); //获得图像高度
    int stride = (4 * ((8 * 3) * width) / 32); //计算行的数量
    int paddingcheck = (stride % 4);
    int padding;
    if (paddingcheck != 0) { //如果行数不能被4整除
        int padcalc = 0;
```

```
        for (int i = 1; padcalc < stride; i++) {
            padcalc = i * 4;
        }
        padding = padcalc - stride;
    }
    else
        padding = 0;
    int op;
    printf("1.增强\n");
    printf("2.减暗\n");
    printf("输入你的选择: \n");
    scanf("%d", &op);
    if(op==1)
     outputFile = fopen("C:/Users/yuehan lian/Desktop/intense.bmp",
"wb");//需要修改路径
    if (op == 2)
     outputFile = fopen("C:/Users/yuehan lian/Desktop/reduce.bmp",
"wb");//需要修改路径
    if (outputFile == NULL) {
        printf("the file cannot be found\n");
    }
    fwrite(header, sizeof(char), 122, outputFile);
    char pad;
    unsigned char pixel[3];
    unsigned char pixeledit[3];
    for (int j = 0; j < height; j++) {
        for (int k = 0; k < width; k++) {
            fread(pixel, sizeof(unsigned char), 3, inputFile); //读取RBG信
息,并写入pixel中
            if (op == 1) {//图像增强    RGB分量 +50,如果超出255,则等于255
                pixel[0] +50 >=255 ? pixel[0]=255 : pixel[0]= pixel[0] +
50;
                pixel[1] +50>= 255 ? pixel[1] = 255 : pixel[1] = pixel[1]
+ 50;
                pixel[2] +50 >=255 ? pixel[2] = 255 : pixel[2] = pixel[2]
+ 50;
            }
            if (op == 2) {//图像减暗   RGB分量 *0.6
                pixel[0] *= 0.6;
                pixel[1] *= 0.6;
                pixel[2] *= 0.6;
            }
            fwrite(&pixel, (sizeof(char)), 3, outputFile);//pixel中的信息写
入输出图像中
        }
        for (int l = 0; l < padding; l++) {
            fread(&pad, 1, 1, inputFile); //读取末尾填充的图像信息
            fwrite(&pad, 1, 1, outputFile); //向输出图像写入末尾填充的图像信息
        }
    }
    int elms_read;//确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
        else {
            fwrite(&pad, 1, 1, outputFile);
```

```
        }
    }
    fclose(inputFile);
    fclose(outputFile);
    return 1;
}

int map(int r, int c, int width) {
    return r * width + c;
}

int pixeldata(const char* input) {
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,重新修改文件路径
        return 0;
    }

    char header[54];
    fread(&header, sizeof(char), 54, inputFile); //获得输入图像头部信息

    int width = *(int*)&header[18]; //获得图像宽度
    int height = abs(*(int*)&header[22]); //获得图像高度
    int stride = (4 * ((8 * 3) * width) / 32); //计算行的数量
    int paddingCheck = (stride % 4);
    int padding;
    if (paddingCheck != 0) { //如果行数不能被4整除
        int padCalc = 0;
        for (int i = 1; padCalc < stride; i++) {
            padCalc = i * 4;
        }
        padding = padCalc - stride;
    }
    else
        padding = 0;

    char pad;
    char pixel[3];
    int merge[256];
    int count[256];
    for (int i = 0; i < 256; ++i)
    {
        count[i] = 0;
    }
    printf("像素数据为: \n");
    int cnt = 0; bool flag = 0;
    for (int j = 0; j < height; j++) {
        if (flag) break;
        for (int k = 0; k < width; k++) {
            fread(pixel, sizeof(char), 3, inputFile); //读取RBG信息,并写入pixel中
            if (cnt < 256)
            {
               merge[cnt]= abs((int)pixel[0])*0.3+abs((int)pixel[1])*0.59+abs((int)pixel[2])*0.11;//归一化
                cnt++;
            }
```

```
            else {
                flag = 1;
                break;
            }

        }
        for (int l = 0; l < padding; l++) {//读取末尾填充的图像信息
            fread(&pad, 1, 1, inputFile);
        }
    }
    int elms_read;  //确保没有空像素被读取
    while (1) {
        elms_read = fread(&pad, 1, 1, inputFile);
        if (elms_read == 0) {
            break;
        }
    }

    printf("\n归一化后的直方图数据为: \n");
    for (int i = 0; i < 256; ++i)
    {
        printf("%3d\t", merge[i]);
        count[merge[i]]++;
        if ((i + 1) % 16 == 0)//每输出16个,就换行
            printf("\n");
    }
    printf("\n");
    /*直方图*/
    for (int i = 1; i < 256; ++i)
    {
        if (count[i] == 0)//只输出大于0的直方图
            continue;
        printf("%d :",i);
        for (int j= 1; j<= count[i]; ++j)
        {
            printf("*");
        }
        printf("\n");
    }

    fclose(inputFile);
    return 1;
}

int showcolorpad(const char* input) {//8/ 显示调色板
    inputFile = fopen(input, "rb");
    if (inputFile == NULL) {
        printf("并无此文件,请检查main函数中定义的路径是否正确!\n");//文件读取失败,
重新修改文件路径
        return 0;
    }
    fseek(inputFile, sizeof(BITMAPFILEHEADER), 0);
    BITMAPINFOHEADER head;
    fread(&head, sizeof(BITMAPINFOHEADER), 1, inputFile);
    unsigned int width = head.biWidth;
    unsigned int height = head.biHeight;
    unsigned int bitcount = head.biBitCount;
    int linebyte = (width * bitcount / 8 + 3) / 4 * 4;
```

```
    RGBTRIPLE* pcolortable = new RGBTRIPLE[256];
    if (bitcount == 8) {
        fread(pcolortable, sizeof(RGBTRIPLE), 256, inputFile);
    }
    else {
        printf("该图不是8位BMP图像,没有调色盘!\n");
        return 0;
    }
    for (size_t i = 0; i < 256; ++i)
    {
        printf("%3d %3d %3d\t", (unsigned int)(pcolortable[i].rgbBlue),
(unsigned int)(pcolortable[i].rgbGreen), (unsigned int)
(pcolortable[i].rgbRed));
        if ((i + 1) % 4 == 0) printf("\n");//每输出4个数据,就换行
    }
    fclose(inputFile);
    return 1;
}

int main(void) {//主函数

    while (1)
    {
        int op;
        printf("BMP图像处理作业菜单\n");
        char infile[]= "C:/Users/yuehan lian/Desktop/a1.bmp";//需要修改文件路
径
       // 请输入BMP图像所在位置(例如:\"C:/Users/yuehan lian/Desktop/1.bmp
        printf("0.结束啦\n");
        printf("1.读取BMP图像(8/24bit)\n");
        printf("2.保存BMP图像(8/24bit)\n");
        printf("3.显示图像数据\n");
        printf("4.计算直方图\n");
        printf("5.图像增强/减暗\n");
        printf("6.反色\n");
        printf("7.灰度图(同位转换,不改变位数)\n");
        printf("8.显示调色板(8bit)\n");

        printf("\n输入选项: ");
        scanf("%d", &op);
        printf("\n");
        switch (op)
        {
                case 0: {
                    printf("感谢你的使用!\n");
                    return 0;
                    break;
                }
                case 1: {
                    if (read(infile) == 0) {
                        return 0;}// 1  读图像
                    printf("读取成功!\n");
                   break; }
                case 2:{
                    if (save(infile) == 0) {
                        return 0; }// 2 保存图像
                    printf("保存成功!\n");
                    break; }
```

```
                case 3: {
                    if (show(infile) == 0) {
                        return 0; }// 3 显示图像数据
                    printf("显示完毕!\n");
                    break;  }
                case 4: {
                    if (pixeldata(infile) == 0) {
                        return 0;}// 4 直方图
                    printf("显示完毕!\n");
                    break; }
                case 5: {
                    if (intense(infile) == 0) {
                        return 0;}// 5 图像增强/减暗
                    printf("保存成功!\n");
                    break;}
                case 6: {
                    if (reverse(infile) == 0) {
                        return 0; }// 6 反色
                    printf("保存成功!\n");
                    break; }
                case 7: {
                    if (greyScale(infile) == 0) {
                        return 0;  }// 7 灰度图
                    printf("保存成功!\n");
                    break; }
                case 8: {
                    if (showcolorpad(infile) == 0) {
                        return 0; }//8 显示调色板
                    printf("显示完毕!\n");
                    break; }
                default:
                    break;
            }
        }

        return 0;
}
```