［演習06］画像情報処理（ウォーミングアップ）

学生番号 ： 20216187

氏名 ： 劉潤之

提出日 ： 2022/11/03

［レポート作成の準備］

1. scriptコマンドを用いて，プログラムリスト・コンパイル結果・実行結果を一つのファイルに書き出しておく．（演習問題ごとに，ファイルを書き出しておくこと．）
2. レポートに画像を載せる必要がある場合には，画像を準備しておく．（どのような図を記載すべきかについては問題文に示されている．）

［レポート作成方法］

次ページ以降において，演習問題ごとに，以下を実施すること．

1. 問題番号を，ドロップダウンリストより2箇所選択する．
2. プログラムリスト・コンパイル結果・実行結果を，準備しておいたファイルからコピーし，所定の場所に，テキスト形式でペーストする．
3. 必要に応じて，所定の場所に図を挿入するとともに，その題目や説明を記入する．

なお，未使用なページやスペースは，そのままにしておいてよい（削除しなくてよい）．

［レポート提出方法］

すべての演習問題を終了した後，レポートを作成して提出する．ここで，レポートの提出期限は，原則として，次回演習日の前夜までとする．

提出は，工学部の「ポータルサイト」の課題提出のページから実施すること．なお，提出の際，コメントを特に記載する必要はない．

［提出ファイル］

提出は，PDFファイルとする．（wordでレポートを作成し，一旦保存する．その後，同ファイルを「名前を付けて保存（コピーを保存）」する．ここで，“ファイル名”の下にある“ファイルの種類”をPDFとすれば，PDFファイルが作成できる．）

なお，ファイル名は，“01\_xxxxxxxx.pdf”とする．ここで，先頭の2桁の値は演習問題の回，末尾のxxxxxxxxの部分は学生番号とする．

【問題6-A-1】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Thu Nov 3 14:25:24 2022

u20216187@gw[31]: cat 06\_A\_01.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAX 100

#define CH 3

#define Ych 0

#define ROW 3

#define COL 3

unsigned char header[54];

unsigned char imgin[3][512][512];//入力画像

unsigned char imgout[3][512][512];//

double dtemp[3][512][512];

int itemp[3][512][512];

void processing();

void get\_data();

void put\_data();

void rgb\_to\_ybr();

void ybr\_to\_rgb();

int width, height;//画像の幅と高さ

int bits;//画像のビット数

double rgb\_con\_ybr[ROW][COL] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[ROW][COL] = {

{ 1.0000, 0.0000, 1.4020},

{ 1.0000,-0.3441,-0.7141},

{ 1.0000, 1.7720, 0.0000}

};

int main() {

get\_data();

rgb\_to\_ybr();

processing();

ybr\_to\_rgb();

put\_data();

return 0;

}

int calculate(int a, int b);//バイト数計算

void get\_data() {

FILE\* fp;

char filename[MAX];

int c;

int filesize, offset, bite\_px;//画像の属性

printf("ファイル名を入力して下さい:");

scanf("%s", filename);

fp = fopen(filename, "rb");

if (fp == NULL) {

printf("%sをオーブンできません\n", filename);

exit(1);

}

printf("%sをオーブンしました.\n", filename);

for (int i = 0; i < 54; i++) {

c = fgetc(fp);

header[i] = c;

}

/\* printf("\n\n<ファイルタイプ>\n");

for (int i = 0; i < 2; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\nファイルサイズ>\n");

filesize = calculate(5, 2);

/\* for (int i = 2; i < 6; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n\n%dバイト", filesize);

/\* printf("\n\n<予約領域>\n");

for (int i = 6; i < 10; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<オフセット>\n");

/\*for (int i = 10; i < 14; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

offset = calculate(13, 10);

printf("\n\n%dバイト", offset);

/\* printf("\n\n<情報へッダサイズ>\n");

for (int i = 14; i < 18; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<画像の幅>\n");

width = calculate(21, 18);

/\*for (int i = 18; i < 22; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%d画素\n", width);

printf("\n\n<画像の高さ>");

height = calculate(25, 22);

/\*for (int i = 22; i < 26; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dライン\n", height);

/\* printf("\n\n<色プレーン数>\n");

for (int i = 26; i < 28; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<1画素にあたりのビット数>\n");

bite\_px = calculate(29, 28);

/\* for (int i = 28; i < 30; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dビット", bite\_px);

/\* printf("\n\n<圧縮方式>\n");

for (int i = 30; i < 34; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<画像データサイズ>\n");

for (int i = 34; i < 38; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<水平解像度>\n");

for (int i = 39; i < 42; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<垂直解像度>\n");

for (int i = 42; i < 46; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<色数>\n");

for (int i = 46; i < 50; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<重要な色数>\n");

for (int i = 50; i < 54; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<挿入ビット数>\n");

bits = (offset + width \* height \* (bite\_px / 8)) % 4;

printf("%dバイト\n", bits);

//imgin[][][]の初期化

for (int h = height - 1; h >= 0; h--) {

for (int w = 0; w < width; w++) {

for (int i = 2; i >= 0; i--) {

imgin[i][w][h] = (unsigned char)fgetc(fp);

}

}

}

fclose(fp);

printf("\n%sをクローズしました.\n", filename);

}

int calculate(int a, int b) {

int value;

value = header[a];

for (int i = a - 1; i >= b; i--) {

value <<= 8;

value += header[i];

}

return value;

}

void processing() {

/\* for (int y=height-1;y>=0;y--){

for(int x=0;x<width;x++){

for(int i=3;i>0;i--){

imgout[i][x][y]=imgin[i][width-1-x][y];

}

}

}\*/

printf("出力画像データを作成しました.\n");

}

void put\_data() {

FILE\* fp;

char cpfilename[MAX];//出力画像名前

printf("出力ファイル名を入力してください:");

scanf(" %s", cpfilename);

fp = fopen(cpfilename, "wp");

printf("%sをオープンしました.\n", cpfilename);

for (int i = 0; i < 54; i++) {

fputc(header[i], fp);

}

for (int h = height - 1; h >= 0; h--) {

for (int w =width; w > 0; w--) {

for (int i = 2; i >= 0; i--) {

fputc(imgout[i][w][h], fp);

}

}

}

for (int i = 0; i < bits; i++) {

fputc('\0', fp);

}

fclose(fp);

printf("%s をクローズしました.\n", cpfilename);

}

void rgb\_to\_ybr() {

int i, x, y;

/\*if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (int j = 0; j < 3; j++)

dtemp[i][x][y] += rgb\_con\_ybr[i][j] \* (double)imgin[j][x][y];

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (i != 0) {

itemp[i][x][y] += 128;

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgin[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

void ybr\_to\_rgb() {

int i, x, y, j;

printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (j = 0; j < 3; j++)

if (j == 0)

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)imgin[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgin[j][x][y] - 128);

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgout[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力RGB信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

u20216187@gw[32]: gcc -Wall 06\_A\_01.c -o 06\_ A\_01.c [K[K

u20216187@gw[33]: ./06\_A\_01

ファイル名を入力して下さい:parrots.bmp

parrots.bmpをオーブンしました.

ファイルサイズ>

196664バイト

<オフセット>

54バイト

<画像の幅>

256画素

<画像の高さ>

256ライン

<1画素にあたりのビット数>

24ビット

<挿入ビット数>

2バイト

parrots.bmpをクローズしました.

出力画像データを作成しました.

<入力YCbCr信号(整数値)>

画像サイズが大きいため表示しません.

出力ファイル名を入力してください:parrots\_06\_A\_1.bmp

parrots\_06\_A\_1.bmpをオープンしました.

parrots\_06\_A\_1.bmp をクローズしました.

u20216187@gw[34]: exit

exit

Script done on Thu Nov 3 14:26:42 2022

［添付図］

|  |  |
| --- | --- |
|  |  |
| Parrots.bmp | Parrots\_06\_A\_1.bmp |

【問題6-B-1】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Thu Nov 3 14:26:55 2022

u20216187@gw[31]: cat 6-B-1.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAX 100

#define CH 3

#define Ych 0

#define ROW 3

#define COL 3

unsigned char header[54];

unsigned char imgin[3][512][512];//入力画像

unsigned char imgout[3][512][512];//

double dtemp[3][512][512];

int itemp[3][512][512];

void processing();

void get\_data();

void put\_data();

void rgb\_to\_ybr();

void ybr\_to\_rgb();

int width, height;//画像の幅と高さ

int bits;//画像のビット数

double rgb\_con\_ybr[ROW][COL] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[ROW][COL] = {

{ 1.0000, 0.0000, 1.4020},

{ 1.0000,-0.3441,-0.7141},

{ 1.0000, 1.7720, 0.0000}

};

char ybr\_name[3][3] = { "Y","Cb","Cr" };

int main() {

get\_data();

rgb\_to\_ybr();

processing();

ybr\_to\_rgb();

put\_data();

return 0;

}

int calculate(int a, int b);//バイト数計算

void get\_data() {

FILE\* fp;

char filename[MAX];

int c;

int filesize, offset, bite\_px;//画像の属性

printf("ファイル名を入力して下さい:");

scanf("%s", filename);

fp = fopen(filename, "rb");

if (fp == NULL) {

printf("%sをオーブンできません\n", filename);

exit(1);

}

printf("%sをオーブンしました.\n", filename);

for (int i = 0; i < 54; i++) {

c = fgetc(fp);

header[i] = c;

}

/\* printf("\n\n<ファイルタイプ>\n");

for (int i = 0; i < 2; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\nファイルサイズ>\n");

filesize = calculate(5, 2);

/\* for (int i = 2; i < 6; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n\n%dバイト", filesize);

/\* printf("\n\n<予約領域>\n");

for (int i = 6; i < 10; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<オフセット>\n");

/\*for (int i = 10; i < 14; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

offset = calculate(13, 10);

printf("\n\n%dバイト", offset);

/\* printf("\n\n<情報へッダサイズ>\n");

for (int i = 14; i < 18; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<画像の幅>\n");

width = calculate(21, 18);

/\*for (int i = 18; i < 22; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%d画素\n", width);

printf("\n\n<画像の高さ>");

height = calculate(25, 22);

/\*for (int i = 22; i < 26; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dライン\n", height);

/\* printf("\n\n<色プレーン数>\n");

for (int i = 26; i < 28; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<1画素にあたりのビット数>\n");

bite\_px = calculate(29, 28);

/\* for (int i = 28; i < 30; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dビット", bite\_px);

/\* printf("\n\n<圧縮方式>\n");

for (int i = 30; i < 34; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<画像データサイズ>\n");

for (int i = 34; i < 38; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<水平解像度>\n");

for (int i = 39; i < 42; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<垂直解像度>\n");

for (int i = 42; i < 46; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<色数>\n");

for (int i = 46; i < 50; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<重要な色数>\n");

for (int i = 50; i < 54; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<挿入ビット数>\n");

bits = (offset + width \* height \* (bite\_px / 8)) % 4;

printf("%dバイト\n", bits);

//imgin[][][]の初期化

for (int h = height - 1; h >= 0; h--) {

for (int w = 0; w < width; w++) {

for (int i = 2; i >= 0; i--) {

imgin[i][w][h] = (unsigned char)fgetc(fp);

}

}

}

fclose(fp);

printf("\n%sをクローズしました.\n", filename);

}

int calculate(int a, int b) {

int value;

value = header[a];

for (int i = a - 1; i >= b; i--) {

value <<= 8;

value += header[i];

}

return value;

}

void processing() {

/\* for (int y=height-1;y>=0;y--){

for(int x=0;x<width;x++){

for(int i=3;i>0;i--){

imgout[i][x][y]=imgin[i][width-1-x][y];

}

}

}\*/

int i, x, y;

int copy[3];

printf("\n コピーモードを入力して下さい.\n");

printf("(コピーする場合 : 1, 固定値に置き換える場合 : 0)\n");

for (i = 0; i < 3; i++) {

printf("%-2s : ", ybr\_name[i]);

scanf("%d", &copy[i]);

}

for (i = 0; i < 3; i++) {

if (copy[i] == 1) {

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

imgout[i][x][y] = imgin[i][x][y];

}

}

}

else {

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

imgout[i][x][y] = 128;

}

}

}

}

printf("出力画像データを作成しました.\n");

}

void put\_data() {

FILE\* fp;

char filename1[MAX];

int i, x, y;

printf("出力ファイル名を入力して下さい:");

scanf("%s", filename1);

fp = fopen(filename1, "wp");

printf("%sをオープンしました.\n", filename1);

for (i = 0; i < 54; i++) {

fputc(header[i], fp);

}

for (y = height - 1; y >= 0; y--) {

for (x = 0; x < width; x++) {

for (i = 2; i >= 0; i--) {

fputc(imgout[i][x][y], fp);

}

}

}

for (i = 0; i < bits; i++) {

fputc('\0', fp);

}

fclose(fp);

printf("%sをクローズしました.\n", filename1);

}

void rgb\_to\_ybr() {

int i, x, y;

/\*if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (int j = 0; j < 3; j++)

dtemp[i][x][y] += rgb\_con\_ybr[i][j] \* (double)imgin[j][x][y];

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (i != 0) {

itemp[i][x][y] += 128;

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgin[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

void ybr\_to\_rgb() {

int i, x, y, j;

/\* printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (j = 0; j < 3; j++)

if (j == 0)

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)imgin[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgin[j][x][y] - 128);

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgout[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力RGB信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

u20216187@gw[32]: gcc 6-B-1.c -o 6-B-1

u20216187@gw[33]: ./6-B-1

ファイル名を入力して下さい:pepper.bmp

pepper.bmpをオーブンしました.

ファイルサイズ>

196664バイト

<オフセット>

54バイト

<画像の幅>

256画素

<画像の高さ>

256ライン

<1画素にあたりのビット数>

24ビット

<挿入ビット数>

2バイト

pepper.bmpをクローズしました.

コピーモードを入力して下さい.

(コピーする場合 : 1, 固定値に置き換える場合 : 0)

Y : 1

Cb : 0

Cr : 0

出力画像データを作成しました.

出力ファイル名を入力して下さい:pepper\_06\_B\_1\_1.bmp

pepper\_06\_B\_1\_1.bmpをオープンしました.

pepper\_06\_B\_1\_1.bmpをクローズしました.

u20216187@gw[34]: ./6-B-1

ファイル名を入力して下さい:pepper.bmp

pepper.bmpをオーブンしました.

ファイルサイズ>

196664バイト

<オフセット>

54バイト

<画像の幅>

256画素

<画像の高さ>

256ライン

<1画素にあたりのビット数>

24ビット

<挿入ビット数>

2バイト

pepper.bmpをクローズしました.

コピーモードを入力して下さい.

(コピーする場合 : 1, 固定値に置き換える場合 : 0)

Y : 0

Cb : 1

Cr : 1

出力画像データを作成しました.

出力ファイル名を入力して下さい:pepper\_06\_B\_1\_2.bmp

pepper\_06\_B\_1\_2.bmpをオープンしました.

pepper\_06\_B\_1\_2.bmpをクローズしました.

u20216187@gw[35]: exit

exit

Script done on Thu Nov 3 14:28:39 2022

［添付図］

|  |  |
| --- | --- |
|  |  |
| Pepper\_06\_B\_1\_1.bmp | Pepper\_06\_B\_1\_2.bmp |

【問題6-B-2】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Thu Nov 3 14:28:50 2022

u20216187@gw[31]: cat 6-B-2.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAX 100

#define CH 3

#define Ych 0

#define ROW 3

#define COL 3

unsigned char header[54];

unsigned char imgin[3][512][512];//入力画像

unsigned char imgout[3][512][512];//

double dtemp[3][512][512];

int itemp[3][512][512];

void processing();

void get\_data();

void put\_data();

void rgb\_to\_ybr();

void ybr\_to\_rgb();

int width, height;//画像の幅と高さ

int bits;//画像のビット数

double rgb\_con\_ybr[ROW][COL] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[ROW][COL] = {

{ 1.0000, 0.0000, 1.4020},

{ 1.0000,-0.3441,-0.7141},

{ 1.0000, 1.7720, 0.0000}

};

//char ybr\_name[3][3] = { "Y","Cb","Cr" };

int main() {

get\_data();

rgb\_to\_ybr();

processing();

ybr\_to\_rgb();

put\_data();

return 0;

}

int calculate(int a, int b);//バイト数計算

void get\_data() {

FILE\* fp;

char filename[MAX];

int c;

int filesize, offset, bite\_px;//画像の属性

printf("ファイル名を入力して下さい:");

scanf("%s", filename);

fp = fopen(filename, "rb");

if (fp == NULL) {

printf("%sをオーブンできません\n", filename);

exit(1);

}

printf("%sをオーブンしました.\n", filename);

for (int i = 0; i < 54; i++) {

c = fgetc(fp);

header[i] = c;

}

/\* printf("\n\n<ファイルタイプ>\n");

for (int i = 0; i < 2; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\nファイルサイズ>\n");

filesize = calculate(5, 2);

/\* for (int i = 2; i < 6; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n\n%dバイト", filesize);

/\* printf("\n\n<予約領域>\n");

for (int i = 6; i < 10; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<オフセット>\n");

/\*for (int i = 10; i < 14; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

offset = calculate(13, 10);

printf("\n\n%dバイト", offset);

/\* printf("\n\n<情報へッダサイズ>\n");

for (int i = 14; i < 18; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<画像の幅>\n");

width = calculate(21, 18);

/\*for (int i = 18; i < 22; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%d画素\n", width);

printf("\n\n<画像の高さ>");

height = calculate(25, 22);

/\*for (int i = 22; i < 26; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dライン\n", height);

/\* printf("\n\n<色プレーン数>\n");

for (int i = 26; i < 28; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<1画素にあたりのビット数>\n");

bite\_px = calculate(29, 28);

/\* for (int i = 28; i < 30; i++) {

printf("header[%d]=%02x ", i, header[i]);

}\*/

printf("\n%dビット", bite\_px);

/\* printf("\n\n<圧縮方式>\n");

for (int i = 30; i < 34; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<画像データサイズ>\n");

for (int i = 34; i < 38; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<水平解像度>\n");

for (int i = 39; i < 42; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<垂直解像度>\n");

for (int i = 42; i < 46; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<色数>\n");

for (int i = 46; i < 50; i++) {

printf("header[%d]=%02x ", i, header[i]);

}

printf("\n\n<重要な色数>\n");

for (int i = 50; i < 54; i++) {

printf("header[%d]=%02x ", i, header[i]);

} \*/

printf("\n\n<挿入ビット数>\n");

bits = (offset + width \* height \* (bite\_px / 8)) % 4;

printf("%dバイト\n", bits);

//imgin[][][]の初期化

for (int h = height - 1; h >= 0; h--) {

for (int w = 0; w < width; w++) {

for (int i = 2; i >= 0; i--) {

imgin[i][w][h] = (unsigned char)fgetc(fp);

}

}

}

fclose(fp);

printf("\n%sをクローズしました.\n", filename);

}

int calculate(int a, int b) {

int value;

value = header[a];

for (int i = a - 1; i >= b; i--) {

value <<= 8;

value += header[i];

}

return value;

}

void processing() {

/\* for (int y=height-1;y>=0;y--){

for(int x=0;x<width;x++){

for(int i=3;i>0;i--){

imgout[i][x][y]=imgin[i][width-1-x][y];

}

}

}\*/

/\*int i, x, y;

int copy[3];

printf("\n コピーモードを入力して下さい.\n");

printf("(コピーする場合 : 1, 固定値に置き換える場合 : 0)\n");

for (i = 0; i < 3; i++) {

printf("%-2s : ", ybr\_name[i]);

scanf("%d", &copy[i]);

}

for (i = 0; i < 3; i++) {

if (copy[i] == 1) {

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

imgout[i][x][y] = imgin[i][x][y];

}

}

}

else {

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

imgout[i][x][y] = 128;

}

}

}

}

\*/

int i, x, y;

int block, b\_height, b\_width;

block = 2;

b\_height = height / block;

b\_width = width / block;

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if ((x / b\_width + y / b\_height) % 2 == 0) {

imgout[i][x][y] = imgin[i][x][y];

}

else if (i != 0) {

imgout[i][x][y] = 128;

}

else {

imgout[i][x][y] = 0;

}

}

}

}

printf("出力画像データを作成しました.\n");

}

void put\_data() {

FILE\* fp;

char filename1[MAX];

int i, x, y;

printf("出力ファイル名を入力して下さい:");

scanf("%s", filename1);

fp = fopen(filename1, "wp");

printf("%sをオープンしました.\n", filename1);

for (i = 0; i < 54; i++) {

fputc(header[i], fp);

}

for (y = height - 1; y >= 0; y--) {

for (x = 0; x < width; x++) {

for (i = 2; i >= 0; i--) {

fputc(imgout[i][x][y], fp);

}

}

}

for (i = 0; i < bits; i++) {

fputc('\0', fp);

}

fclose(fp);

printf("%sをクローズしました.\n", filename1);

}

void rgb\_to\_ybr() {

int i, x, y;

/\*if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (int j = 0; j < 3; j++)

dtemp[i][x][y] += rgb\_con\_ybr[i][j] \* (double)imgin[j][x][y];

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (i != 0) {

itemp[i][x][y] += 128;

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgin[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

void ybr\_to\_rgb() {

int i, x, y, j;

/\* printf("\n<入力YCbCr信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<Y信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[0][x][y]);

}

printf("\n");

}

printf("\n<Cb信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[1][x][y]);

}

printf("\n");

}

printf("\n<Cr信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgin[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

dtemp[i][x][y] = 0.0;

for (j = 0; j < 3; j++)

if (j == 0)

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)imgin[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgin[j][x][y] - 128);

}

}

}

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

for (i = 0; i < 3; i++) {

if (dtemp[i][x][y] > 0.0) {

itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);

}

else {

itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);

}

if (itemp[i][x][y] > 255) {

itemp[i][x][y] = 255;

}

else if (itemp[i][x][y] < 0) {

itemp[i][x][y] = 0;

}

imgout[i][x][y] = (unsigned char)itemp[i][x][y];

}

}

}

/\*printf("\n<入力RGB信号(整数値)>\n");

if (height <= 16 || width <= 16) {

printf("\n<R信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[0][x][y]);

}

printf("\n");

}

printf("\n<G信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[1][x][y]);

}

printf("\n");

}

printf("\n<B信号>\n");

for (y = 0; y < height; y++) {

for (x = 0; x < width; x++) {

printf("%02x ", imgout[2][x][y]);

}

printf("\n");

}

}

else {

printf("画像サイズが大きいため表示しません.\n");

}\*/

}

u20216187@gw[32]: gcc -Wall 6-B-2.c -o 6-B-2

u20216187@gw[33]: 6-B-2[1@.[1@/

ファイル名を入力して下さい:maldives.bmp

maldives.bmpをオーブンしました.

ファイルサイズ>

230456バイト

<オフセット>

54バイト

<画像の幅>

320画素

<画像の高さ>

240ライン

<1画素にあたりのビット数>

24ビット

<挿入ビット数>

2バイト

maldives.bmpをクローズしました.

出力画像データを作成しました.

出力ファイル名を入力して下さい:maldives\_06\_B\_2.bmp

maldives\_06\_B\_2.bmpをオープンしました.

maldives\_06\_B\_2.bmpをクローズしました.

u20216187@gw[34]: exit

exit

Script done on Thu Nov 3 14:29:43 2022

［添付図］

|  |  |
| --- | --- |
|  |  |
| Maldives.bmp | Maldib\ves\_06\_B\_2.bmp |

【問題アイテムを選択してください。-アイテムを選択してください。】

［プログラムリスト・コンパイル結果・実行結果］

ここをクリックまたはタップしてテキストを入力してください。

［添付図］

|  |  |
| --- | --- |
|  |  |
| ここをクリックまたはタップしてテキストを入力してください。 | ここをクリックまたはタップしてテキストを入力してください。 |