アイテムを選択してください。

学生番号 ： 20216187

氏名 ： 劉潤之

提出日 ： 2022/11/10

［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］

Script started on Thu Nov 10 15:10:08 2022

u20216187@gw[31]: cat ctog.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, bite\_px;//画像の属性

//int offset;

//printf("ファイル名を入力して下さい:");

//scanf("%s", filename);

char filename[20] = "parrots.bmp";

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() {

int i, x, y;

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

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

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

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

if (i != 0) imgin[i][x][y] = 128;

}

}

}

printf("カラー画像を濃淡画像に変換しました.\n");

}

void put\_data() {

FILE\* fp;

//char filename1[MAX];

char filename1[20] ="parrots\_gray.bmp";

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, "w");

//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("出力画像:%s.\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 ctog.c -o ctog

u20216187@gw[33]: ctog parrots.bmp parrots\_gray.bmp

<ファイルサイズ>

196664バイト

<画像の幅>

256画素

<画像の高さ>カラー画像を濃淡画像に変換しました.

出力画像:parrots\_gray.bmp.

u20216187@gw[34]: exit

exit

Script done on Thu Nov 10 15:11:24 2022

［添付図］

|  |  |
| --- | --- |
|  |  |
| Parrots.bmp | Parrots\_gray.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。 |

【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。 |

【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。 |