［演習04］ビットマップファイル (2)

学生番号 ： 20216187

氏名 ： 劉潤之

提出日 ： 2022/10/18

［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4-A-1】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Tue Oct 18 17:29:06 2022

u20216187@gw[31]: cat 4\_A\_!.c.c[K1.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAX 100

unsigned char header[54];

unsigned char imgin[3][512][512];

void processing();

void get\_data();

int width,height;//画像の幅と高さ

int bits;//画像のビット数

int main() {

get\_data();

processing();

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(){

int x,y;//RGBを表示するため変数

printf("入力画像データを表示します.");

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");

}

}

u20216187@gw[32]: gcc -Wall 4\_A\_1.c -o 4\_A\_1

u20216187@gw[33]: ./4\_A\_1

ファイル名を入力して下さい:test04.bmp

test04.bmpをオーブンしました.

<ファイルタイプ>

header[0]=42 header[1]=4d

ファイルサイズ>

header[2]=68 header[3]=00 header[4]=00 header[5]=00

104バイト

<予約領域>

header[6]=00 header[7]=00 header[8]=00 header[9]=00

<オフセット>

header[10]=36 header[11]=00 header[12]=00 header[13]=00

54バイト

<情報へッダサイズ>

header[14]=28 header[15]=00 header[16]=00 header[17]=00

<画像の幅>

header[18]=04 header[19]=00 header[20]=00 header[21]=00

4画素

<画像の高さ>header[22]=04 header[23]=00 header[24]=00 header[25]=00

4ライン

<色プレーン数>

header[26]=01 header[27]=00

<1画素にあたりのビット数>

header[28]=18 header[29]=00

24ビット

<圧縮方式>

header[30]=00 header[31]=00 header[32]=00 header[33]=00

<画像データサイズ>

header[34]=32 header[35]=00 header[36]=00 header[37]=00

<水平解像度>

header[39]=0b header[40]=00 header[41]=00

<垂直解像度>

header[42]=12 header[43]=0b header[44]=00 header[45]=00

<色数>

header[46]=00 header[47]=00 header[48]=00 header[49]=00

<重要な色数>

header[50]=00 header[51]=00 header[52]=00 header[53]=00

<挿入ビット数>

2バイト

test04.bmpをクローズしました.

入力画像データを表示します.

<R信号>

00 01 02 03

04 05 06 07

08 09 0a 0b

0c 0d 0e 0f

<G信号>

10 11 12 13

14 15 16 17

18 19 1a 1b

1c 1d 1e 1f

<B信号>

20 21 22 23

24 25 26 27

28 29 2a 2b

2c 2d 2e 2f

u20216187@gw[34]: exit

exit

Script done on Tue Oct 18 17:29:48 2022

［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4-B-1】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Tue Oct 18 17:33:25 2022

u20216187@gw[31]: cat 4\_B\_1.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAX 100

unsigned char header[54];

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

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

void processing();

void get\_data();

void put\_data();

int width,height;//画像の幅と高さ

int bits;//画像のビット数

int main() {

get\_data();

processing();

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(){

int x,y;//RGBを表示するため変数

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

printf("入力画像データを表示します.");

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ません.\n");

}

for(int h=0;h<height;h++){

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

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

imgout[i][w][h]=imgin[i][w][h];

}

}

}

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=0;w<width;w++){

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

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

}

}

}

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

fputc('\0',fp);

}

fclose(fp);

printf("%s をクローズしました.\n",cpfilename);

}

u20216187@gw[32]: gcc -Wall 4\_B\_1.c -o 4\_B\_1

u20216187@gw[33]: ./4\_b[KB\_1

ファイル名を入力して下さい:test04.bmp

test04.bmpをオーブンしました.

<ファイルタイプ>

header[0]=42 header[1]=4d

ファイルサイズ>

header[2]=68 header[3]=00 header[4]=00 header[5]=00

104バイト

<予約領域>

header[6]=00 header[7]=00 header[8]=00 header[9]=00

<オフセット>

header[10]=36 header[11]=00 header[12]=00 header[13]=00

54バイト

<情報へッダサイズ>

header[14]=28 header[15]=00 header[16]=00 header[17]=00

<画像の幅>

header[18]=04 header[19]=00 header[20]=00 header[21]=00

4画素

<画像の高さ>header[22]=04 header[23]=00 header[24]=00 header[25]=00

4ライン

<色プレーン数>

header[26]=01 header[27]=00

<1画素にあたりのビット数>

header[28]=18 header[29]=00

24ビット

<圧縮方式>

header[30]=00 header[31]=00 header[32]=00 header[33]=00

<画像データサイズ>

header[34]=32 header[35]=00 header[36]=00 header[37]=00

<水平解像度>

header[39]=0b header[40]=00 header[41]=00

<垂直解像度>

header[42]=12 header[43]=0b header[44]=00 header[45]=00

<色数>

header[46]=00 header[47]=00 header[48]=00 header[49]=00

<重要な色数>

header[50]=00 header[51]=00 header[52]=00 header[53]=00

<挿入ビット数>

2バイト

test04.bmpをクローズしました.

入力画像データを表示します.

<R信号>

00 01 02 03

04 05 06 07

08 09 0a 0b

0c 0d 0e 0f

<G信号>

10 11 12 13

14 15 16 17

18 19 1a 1b

1c 1d 1e 1f

<B信号>

20 21 22 23

24 25 26 27

28 29 2a 2b

2c 2d 2e 2f

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ください:test04cp.bmp

test04cp.bmpをオープンしました.

test04cp.bmp をクローズしました.

u20216187@gw[34]: ./4\_B\_1

ファイル名を入力して下さい:lenna.bmp

lenna.bmpをオーブンしました.

<ファイルタイプ>

header[0]=42 header[1]=4d

ファイルサイズ>

header[2]=38 header[3]=00 header[4]=0c header[5]=00

786488バイト

<予約領域>

header[6]=00 header[7]=00 header[8]=00 header[9]=00

<オフセット>

header[10]=36 header[11]=00 header[12]=00 header[13]=00

54バイト

<情報へッダサイズ>

header[14]=28 header[15]=00 header[16]=00 header[17]=00

<画像の幅>

header[18]=00 header[19]=02 header[20]=00 header[21]=00

512画素

<画像の高さ>header[22]=00 header[23]=02 header[24]=00 header[25]=00

512ライン

<色プレーン数>

header[26]=01 header[27]=00

<1画素にあたりのビット数>

header[28]=18 header[29]=00

24ビット

<圧縮方式>

header[30]=00 header[31]=00 header[32]=00 header[33]=00

<画像データサイズ>

header[34]=00 header[35]=00 header[36]=00 header[37]=00

<水平解像度>

header[39]=0b header[40]=00 header[41]=00

<垂直解像度>

header[42]=12 header[43]=0b header[44]=00 header[45]=00

<色数>

header[46]=00 header[47]=00 header[48]=00 header[49]=00

<重要な色数>

header[50]=00 header[51]=00 header[52]=00 header[53]=00

<挿入ビット数>

2バイト

lenna.bmpをクローズしました.

画像サイズが大きいため表示しまません.

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ください:lennacp.bmp

lennacp.bmpをオープンしました.

lennacp.bmp をクローズしました.

u20216187@gw[35]: exit

exit

Script done on Tue Oct 18 17:34:31 2022

［添付図］

|  |  |
| --- | --- |
|  |  |
| Lenna.bmp | Lennacp.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。 |