［演習12］空間フィルタリング

学生番号 ： 20216187

氏名 ： 劉潤之

提出日 ： 2023/01/26

［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12-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 Jan 26 20:09:28 2023

u20216187@gw[31]: cat e x\_12\_1.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAXLENGTH 100

unsigned char header[54];

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

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

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

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

void convert(int n);

void get\_data();

void rgb\_to\_ybr();

void processing();

void ybr\_to\_rgb();

void put\_data();

int calculate(int a, int b);

int width, height, bite;

double rgb\_con\_ybr[3][3] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[3][3] = {

{ 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;

}

void get\_data(){

FILE \*fp;

char filename[MAXLENGTH];

int i, c, x, y;

int filesize, offset, bite\_pixel;

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("ファイルをオーブンできません.\n");

exit(1);

}

printf("ファイルをオーブンしました.\n");

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

c = fgetc(fp);

header[i] = (unsigned char)c;

}

/\* //---------------------------------

printf("\n<ファイルタイプ>\n");

printf("header[0]=");

convert(header[0]);

printf(" header[1]=");

convert(header[1]);

printf("\n");

\*/

//---------------------------------

printf("\n<ファイルサイズ>\n");

/\*printf("header[2]=");

convert(header[2]);

printf(" header[3]=");

convert(header[3]);

printf(" header[4]=");

convert(header[4]);

printf(" header[5]=");

convert(header[5]);

\*/

filesize = calculate(5, 2);

printf("\n%dバイト\n", filesize);

/\*//---------------------------------

printf("\n<予約領域>\n");

printf("header[6]=");

convert(header[6]);

printf(" header[7]=");

convert(header[7]);

printf(" header[8]=");

convert(header[8]);

printf(" header[9]=");

convert(header[9]);

printf("\n");

\*/

//---------------------------------

printf("\n<オフセット>\n");

/\*printf("header[10]=");

convert(header[10]);

printf(" header[11]=");

convert(header[11]);

printf(" header[12]=");

convert(header[12]);

printf(" header[13]=");

convert(header[13]);\*/

offset = calculate(13, 10),

printf("\n%dバイト\n", offset);

/\*//---------------------------------

printf("\n<情報ベッダサイズ>\n");

printf("header[14]=");

convert(header[14]);

printf(" header[15]=");

convert(header[15]);

printf(" header[16]=");

convert(header[16]);

printf(" header[17]=");

convert(header[17]);

printf("\n");

\*/

//---------------------------------

printf("\n<画像の幅>\n");

/\*printf("header[18]=");

convert(header[18]);

printf(" header[19]=");

convert(header[19]);

printf(" header[20]=");

convert(header[20]);

printf(" header[21]=");

convert(header[21]);\*/

width = calculate(21, 18);

printf("\n%d画素\n",width);

//---------------------------------

printf("\n<画像の高さ>\n");

/\*printf("header[22]=");

convert(header[22]);

printf(" header[23]=");

convert(header[23]);

printf(" header[24]=");

convert(header[24]);

printf(" header[25]=");

convert(header[25]);\*/

height = calculate(25, 22);

printf("\n%dライン\n",height);

/\*//---------------------------------

printf("\n<色プレーン数>\n");

printf("header[26]=");

convert(header[26]);

printf(" header[27]=");

convert(header[27]);

printf("\n");\*/

//---------------------------------

printf("\n<1画素当たりのビット数>\n");

/\*printf("header[28]=");

convert(header[28]);

printf(" header[29]=");

convert(header[29]);\*/

bite\_pixel = calculate(29, 28);

printf("\n%dビット\n",bite\_pixel);

/\*//---------------------------------

printf("\n<圧縮方式>\n");

printf("header[30]=");

convert(header[30]);

printf(" header[31]=");

convert(header[31]);

printf(" header[32]=");

convert(header[32]);

printf(" header[33]=");

convert(header[33]);

printf("\n");

//---------------------------------

printf("\n<画像データサイズ>\n");

printf("header[34]=");

convert(header[34]);

printf(" header[35]=");

convert(header[35]);

printf(" header[36]=");

convert(header[36]);

printf(" header[37]=");

convert(header[37]);

printf("\n");

//---------------------------------

printf("\n<水平解像度>\n");

printf("header[38]=");

convert(header[38]);

printf(" header[39]=");

convert(header[39]);

printf(" header[40]=");

convert(header[40]);

printf(" header[41]=");

convert(header[41]);

printf("\n");

//---------------------------------

printf("\n<垂直解像度>\n");

printf("header[42]=");

convert(header[42]);

printf(" header[43]=");

convert(header[43]);

printf(" header[44]=");

convert(header[44]);

printf(" header[45]=");

convert(header[45]);

printf("\n");

//---------------------------------

printf("\n<色数>\n");

printf("header[46]=");

convert(header[46]);

printf(" header[47]=");

convert(header[47]);

printf(" header[48]=");

convert(header[48]);

printf(" header[49]=");

convert(header[49]);

printf("\n");

//---------------------------------

printf("\n<重要な色数>\n");

printf("header[50]=");

convert(header[50]);

printf(" header[51]=");

convert(header[51]);

printf(" header[52]=");

convert(header[52]);

printf(" header[53]=");

convert(header[53]);

printf("\n");

\*/

//---------------------------------

printf("\n<挿入ビット数>\n");

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

printf("%dバイト\n",bite);

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

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

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

c = fgetc(fp);

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

}

}

}

fclose(fp);

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

}

void convert(int n){

printf("%02x", n);

}

int calculate(int a, int b){

int i;

int value;

value = header[a];

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

value <<= 8;

value += header[i];

}

return value;

}

void processing(){

int i,j,x,y;

int value;

double filter\_change[3][3];

double tmp[3][3];

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

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

filter\_change[i][j]=(1.0/9.0);

}

}

printf("\n<フィルタ係数>\n");

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

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

printf("%1.5f ",filter\_change[i][j]);

}

printf("\n");

}

printf("フィルタ処理する画素の座標を入力して下さい\n");

printf("水平座標：");

scanf("%d",&x);

printf("垂直座標：");

scanf("%d",&y);

while(x == 0|| x == width-1 || y==0 || y==height-1){

printf("----- この座標は画像の端に位置するためフィルタ処理ができません\n");

printf("水平座標：");

scanf("%d",&x);

printf("垂直座標：");

scanf("%d",&y);

}

printf("＜フィルタ処理に使用する画素の値＞\n");

for(i=-1;i<=1;i++){

for(j=-1;j<=1;j++){

printf("%3d ",imgin[0][x+i][y+j]);

}

printf("\n");

}

printf("＜フィルタ処理後の画素値＞\n");

for(i=-1;i<2;i++){

for(j=-1;j<2;j++){

tmp[i+1][j+1]=imgin[0][x+i][y+i]\*filter\_change[i+1][y+1];

}

}

value =0;

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

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

value +=(int)(tmp[i][j]+0.5);

}

}

printf("%3d\n",value);

}

//printf("入力画像データをコピーして出力画像データを作成しました.\n");

/\*void put\_data(){

FILE \*fp;

char filename1[MAXLENGTH];

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 < bite; 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, j;

/\*

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 ", 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 (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)imgout[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgout[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 e x\_12\_1.c -o e x\_12\_1

u20216187@gw[33]: ./e x\_12\_1

ファイル名を入力して下さい:goldhill.bmp

ファイルをオーブンしました.

<ファイルサイズ>

786488バイト

<オフセット>

54バイト

<画像の幅>

512画素

<画像の高さ>

512ライン

<1画素当たりのビット数>

24ビット

<挿入ビット数>

2バイト

goldhill.bmp をクローズしました.

<フィルタ係数>

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

フィルタ処理する画素の座標を入力して下さい

水平座標：1

垂直座標：1

＜フィルタ処理に使用する画素の値＞

230 229 231

229 227 234

230 231 235

＜フィルタ処理後の画素値＞

231

u20216187@gw[34]: ./ex\_12\_1

ファイル名を入力して下さい:goldhill.bmp

ファイルをオーブンしました.

<ファイルサイズ>

786488バイト

<オフセット>

54バイト

<画像の幅>

512画素

<画像の高さ>

512ライン

<1画素当たりのビット数>

24ビット

<挿入ビット数>

2バイト

goldhill.bmp をクローズしました.

<フィルタ係数>

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

フィルタ処理する画素の座標を入力して下さい

水平座標：1

垂直座標：0

----- この座標は画像の端に位置するためフィルタ処理ができません

水平座標：1

垂直座標：1

＜フィルタ処理に使用する画素の値＞

230 229 231

229 227 234

230 231 235

＜フィルタ処理後の画素値＞

231

u20216187@gw[35]: exit

exit

Script done on Thu Jan 26 20:10:10 2023

［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12-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 Jan 26 20:10:17 2023

u20216187@gw[31]: cat e x\_12\_2.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAXLENGTH 100

unsigned char header[54];

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

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

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

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

void convert(int n);

void get\_data();

void rgb\_to\_ybr();

void processing();

void ybr\_to\_rgb();

void put\_data();

int calculate(int a, int b);

int width, height, bite;

double rgb\_con\_ybr[3][3] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[3][3] = {

{ 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;

}

void get\_data(){

FILE \*fp;

char filename[MAXLENGTH];

int i, c, x, y;

int filesize, offset, bite\_pixel;

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("ファイルをオーブンできません.\n");

exit(1);

}

printf("ファイルをオーブンしました.\n");

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

c = fgetc(fp);

header[i] = (unsigned char)c;

}

/\* //---------------------------------

printf("\n<ファイルタイプ>\n");

printf("header[0]=");

convert(header[0]);

printf(" header[1]=");

convert(header[1]);

printf("\n");

\*/

//---------------------------------

printf("\n<ファイルサイズ>\n");

/\*printf("header[2]=");

convert(header[2]);

printf(" header[3]=");

convert(header[3]);

printf(" header[4]=");

convert(header[4]);

printf(" header[5]=");

convert(header[5]);

\*/

filesize = calculate(5, 2);

printf("\n%dバイト\n", filesize);

/\*//---------------------------------

printf("\n<予約領域>\n");

printf("header[6]=");

convert(header[6]);

printf(" header[7]=");

convert(header[7]);

printf(" header[8]=");

convert(header[8]);

printf(" header[9]=");

convert(header[9]);

printf("\n");

\*/

//---------------------------------

printf("\n<オフセット>\n");

/\*printf("header[10]=");

convert(header[10]);

printf(" header[11]=");

convert(header[11]);

printf(" header[12]=");

convert(header[12]);

printf(" header[13]=");

convert(header[13]);\*/

offset = calculate(13, 10),

printf("\n%dバイト\n", offset);

/\*//---------------------------------

printf("\n<情報ベッダサイズ>\n");

printf("header[14]=");

convert(header[14]);

printf(" header[15]=");

convert(header[15]);

printf(" header[16]=");

convert(header[16]);

printf(" header[17]=");

convert(header[17]);

printf("\n");

\*/

//---------------------------------

printf("\n<画像の幅>\n");

/\*printf("header[18]=");

convert(header[18]);

printf(" header[19]=");

convert(header[19]);

printf(" header[20]=");

convert(header[20]);

printf(" header[21]=");

convert(header[21]);\*/

width = calculate(21, 18);

printf("\n%d画素\n",width);

//---------------------------------

printf("\n<画像の高さ>\n");

/\*printf("header[22]=");

convert(header[22]);

printf(" header[23]=");

convert(header[23]);

printf(" header[24]=");

convert(header[24]);

printf(" header[25]=");

convert(header[25]);\*/

height = calculate(25, 22);

printf("\n%dライン\n",height);

/\*//---------------------------------

printf("\n<色プレーン数>\n");

printf("header[26]=");

convert(header[26]);

printf(" header[27]=");

convert(header[27]);

printf("\n");\*/

//---------------------------------

printf("\n<1画素当たりのビット数>\n");

/\*printf("header[28]=");

convert(header[28]);

printf(" header[29]=");

convert(header[29]);\*/

bite\_pixel = calculate(29, 28);

printf("\n%dビット\n",bite\_pixel);

/\*//---------------------------------

printf("\n<圧縮方式>\n");

printf("header[30]=");

convert(header[30]);

printf(" header[31]=");

convert(header[31]);

printf(" header[32]=");

convert(header[32]);

printf(" header[33]=");

convert(header[33]);

printf("\n");

//---------------------------------

printf("\n<画像データサイズ>\n");

printf("header[34]=");

convert(header[34]);

printf(" header[35]=");

convert(header[35]);

printf(" header[36]=");

convert(header[36]);

printf(" header[37]=");

convert(header[37]);

printf("\n");

//---------------------------------

printf("\n<水平解像度>\n");

printf("header[38]=");

convert(header[38]);

printf(" header[39]=");

convert(header[39]);

printf(" header[40]=");

convert(header[40]);

printf(" header[41]=");

convert(header[41]);

printf("\n");

//---------------------------------

printf("\n<垂直解像度>\n");

printf("header[42]=");

convert(header[42]);

printf(" header[43]=");

convert(header[43]);

printf(" header[44]=");

convert(header[44]);

printf(" header[45]=");

convert(header[45]);

printf("\n");

//---------------------------------

printf("\n<色数>\n");

printf("header[46]=");

convert(header[46]);

printf(" header[47]=");

convert(header[47]);

printf(" header[48]=");

convert(header[48]);

printf(" header[49]=");

convert(header[49]);

printf("\n");

//---------------------------------

printf("\n<重要な色数>\n");

printf("header[50]=");

convert(header[50]);

printf(" header[51]=");

convert(header[51]);

printf(" header[52]=");

convert(header[52]);

printf(" header[53]=");

convert(header[53]);

printf("\n");

\*/

//---------------------------------

printf("\n<挿入ビット数>\n");

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

printf("%dバイト\n",bite);

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

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

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

c = fgetc(fp);

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

}

}

}

fclose(fp);

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

}

void convert(int n){

printf("%02x", n);

}

int calculate(int a, int b){

int i;

int value;

value = header[a];

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

value <<= 8;

value += header[i];

}

return value;

}

void processing(){

int i,j,x,y;

double filter\_change[3][3];

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

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

filter\_change[i][j]=(1.0/9.0);

}

}

printf("\n<フィルタ係数>\n");

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

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

printf("%1.5f ",filter\_change[i][j]);

}

printf("\n");

}

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

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

if(x != 0 && x != width-1 && y !=0 && y !=height-1){

for(int m =-1;m<2;m++){

for(int n=-1;n<2;n++){

imgout[0][x][y]+=(int)(imgin[0][x+m][y+n]\*filter\_change[m+1][n+1]+0.5);

}

}

} else {

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

}

if(imgout[0][x][y]<0)imgout[0][x][y]=0;

if(imgout[0][x][y]>255)imgout[0][x][y]=255;

}

}

for(i=1;i<=2;i++){

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

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

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

}

}

}

}

//printf("入力画像データをコピーして出力画像データを作成しました.\n");

void put\_data(){

FILE \*fp;

char filename1[MAXLENGTH];

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 < bite; 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, j;

/\*

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 ", 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 (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)imgout[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgout[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 e x\_12\_2.c -o e x\_12\_2

u20216187@gw[33]: ./e x\_12\_2

ファイル名を入力して下さい:goldhill.bmp

ファイルをオーブンしました.

<ファイルサイズ>

786488バイト

<オフセット>

54バイト

<画像の幅>

512画素

<画像の高さ>

512ライン

<1画素当たりのビット数>

24ビット

<挿入ビット数>

2バイト

goldhill.bmp をクローズしました.

<フィルタ係数>

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

0.11111 0.11111 0.11111

出力ファイル名を入力して下さい:goldhill\_12\_B\_1.bmp

goldhill\_12\_B\_1.bmpをオープンしました.

goldhill\_12\_B\_1.bmpをクローズしました.

u20216187@gw[34]: ./ex\_12\_2[19G[Kexit

exit

Script done on Thu Jan 26 20:11:12 2023

［添付図］

|  |  |
| --- | --- |
|  |  |
| goldhill | Goldhill\_12\_B\_1 |

【問題12-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 Jan 26 20:17:42 2023

u20216187@gw[31]: cat e x\_12\_3.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAXLENGTH 100

unsigned char header[54];

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

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

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

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

void convert(int n);

void get\_data();

void rgb\_to\_ybr();

void processing();

void ybr\_to\_rgb();

void put\_data();

int calculate(int a, int b);

int width, height, bite;

double rgb\_con\_ybr[3][3] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[3][3] = {

{ 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;

}

void get\_data(){

FILE \*fp;

char filename[MAXLENGTH];

int i, c, x, y;

int filesize, offset, bite\_pixel;

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("ファイルをオーブンできません.\n");

exit(1);

}

printf("ファイルをオーブンしました.\n");

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

c = fgetc(fp);

header[i] = (unsigned char)c;

}

/\* //---------------------------------

printf("\n<ファイルタイプ>\n");

printf("header[0]=");

convert(header[0]);

printf(" header[1]=");

convert(header[1]);

printf("\n");

\*/

//---------------------------------

printf("\n<ファイルサイズ>\n");

/\*printf("header[2]=");

convert(header[2]);

printf(" header[3]=");

convert(header[3]);

printf(" header[4]=");

convert(header[4]);

printf(" header[5]=");

convert(header[5]);

\*/

filesize = calculate(5, 2);

printf("\n%dバイト\n", filesize);

/\*//---------------------------------

printf("\n<予約領域>\n");

printf("header[6]=");

convert(header[6]);

printf(" header[7]=");

convert(header[7]);

printf(" header[8]=");

convert(header[8]);

printf(" header[9]=");

convert(header[9]);

printf("\n");

\*/

//---------------------------------

printf("\n<オフセット>\n");

/\*printf("header[10]=");

convert(header[10]);

printf(" header[11]=");

convert(header[11]);

printf(" header[12]=");

convert(header[12]);

printf(" header[13]=");

convert(header[13]);\*/

offset = calculate(13, 10),

printf("\n%dバイト\n", offset);

/\*//---------------------------------

printf("\n<情報ベッダサイズ>\n");

printf("header[14]=");

convert(header[14]);

printf(" header[15]=");

convert(header[15]);

printf(" header[16]=");

convert(header[16]);

printf(" header[17]=");

convert(header[17]);

printf("\n");

\*/

//---------------------------------

printf("\n<画像の幅>\n");

/\*printf("header[18]=");

convert(header[18]);

printf(" header[19]=");

convert(header[19]);

printf(" header[20]=");

convert(header[20]);

printf(" header[21]=");

convert(header[21]);\*/

width = calculate(21, 18);

printf("\n%d画素\n",width);

//---------------------------------

printf("\n<画像の高さ>\n");

/\*printf("header[22]=");

convert(header[22]);

printf(" header[23]=");

convert(header[23]);

printf(" header[24]=");

convert(header[24]);

printf(" header[25]=");

convert(header[25]);\*/

height = calculate(25, 22);

printf("\n%dライン\n",height);

/\*//---------------------------------

printf("\n<色プレーン数>\n");

printf("header[26]=");

convert(header[26]);

printf(" header[27]=");

convert(header[27]);

printf("\n");\*/

//---------------------------------

printf("\n<1画素当たりのビット数>\n");

/\*printf("header[28]=");

convert(header[28]);

printf(" header[29]=");

convert(header[29]);\*/

bite\_pixel = calculate(29, 28);

printf("\n%dビット\n",bite\_pixel);

/\*//---------------------------------

printf("\n<圧縮方式>\n");

printf("header[30]=");

convert(header[30]);

printf(" header[31]=");

convert(header[31]);

printf(" header[32]=");

convert(header[32]);

printf(" header[33]=");

convert(header[33]);

printf("\n");

//---------------------------------

printf("\n<画像データサイズ>\n");

printf("header[34]=");

convert(header[34]);

printf(" header[35]=");

convert(header[35]);

printf(" header[36]=");

convert(header[36]);

printf(" header[37]=");

convert(header[37]);

printf("\n");

//---------------------------------

printf("\n<水平解像度>\n");

printf("header[38]=");

convert(header[38]);

printf(" header[39]=");

convert(header[39]);

printf(" header[40]=");

convert(header[40]);

printf(" header[41]=");

convert(header[41]);

printf("\n");

//---------------------------------

printf("\n<垂直解像度>\n");

printf("header[42]=");

convert(header[42]);

printf(" header[43]=");

convert(header[43]);

printf(" header[44]=");

convert(header[44]);

printf(" header[45]=");

convert(header[45]);

printf("\n");

//---------------------------------

printf("\n<色数>\n");

printf("header[46]=");

convert(header[46]);

printf(" header[47]=");

convert(header[47]);

printf(" header[48]=");

convert(header[48]);

printf(" header[49]=");

convert(header[49]);

printf("\n");

//---------------------------------

printf("\n<重要な色数>\n");

printf("header[50]=");

convert(header[50]);

printf(" header[51]=");

convert(header[51]);

printf(" header[52]=");

convert(header[52]);

printf(" header[53]=");

convert(header[53]);

printf("\n");

\*/

//---------------------------------

printf("\n<挿入ビット数>\n");

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

printf("%dバイト\n",bite);

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

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

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

c = fgetc(fp);

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

}

}

}

fclose(fp);

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

}

void convert(int n){

printf("%02x", n);

}

int calculate(int a, int b){

int i;

int value;

value = header[a];

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

value <<= 8;

value += header[i];

}

return value;

}

void processing(){

int j,i,x,y;

double n;

double avg[3][3]={{0.0,1.0,0.0},

{1.0,-4.0,1.0},

{0.0,1.0,0.0}};

double sum;

printf("\n<フィルタ係数>\n");

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

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

printf("%3.6f ",avg[i][j]);

}

printf("\n");

}

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

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

if(x<=0 || x>=width || y<=0 || y>=height){

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

}else{

sum=0.0;

for(j=-1;j<=1;j++){

for(i=-1;i<=1;i++){

sum += (double)imgin[0][x+i][y+j]\*avg[i+1][j+1];

}

}

if(sum>=0.0){

n=(int)(sum+0.5);

}

else{

n=(int)(sum-0.5);

}

n += 128;

if(n>255){

n=255;

}

else if(n<0){

n=0;

}

imgout[0][x][y] = (unsigned char)n;

}

imgout[1][x][y] = 128;

imgout[2][x][y] = 128;

}

}

printf("入力画像データをコピーして出力画像データを作成しました.\n");

}

void put\_data(){

FILE \*fp;

char filename1[MAXLENGTH];

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 < bite; 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, j;

/\*

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 ", 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 (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)imgout[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgout[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 e x\_12\_3.c -o e x\_12\_3

u20216187@gw[33]: ./e x\_12\_3

ファイル名を入力して下さい:goldhill.bmp

ファイルをオーブンしました.

<ファイルサイズ>

786488バイト

<オフセット>

54バイト

<画像の幅>

512画素

<画像の高さ>

512ライン

<1画素当たりのビット数>

24ビット

<挿入ビット数>

2バイト

goldhill.bmp をクローズしました.

<フィルタ係数>

0.000000 1.000000 0.000000

1.000000 -4.000000 1.000000

0.000000 1.000000 0.000000

入力画像データをコピーして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:goldhill\_12\_B\_2.bmp

goldhill\_12\_B\_2.bmpをオープンしました.

goldhill\_12\_B\_2.bmpをクローズしました.

u20216187@gw[34]: exit

exit

Script done on Thu Jan 26 20:18:17 2023

［添付図］

|  |  |
| --- | --- |
|  |  |
| Goldhill | Goldhill\_12\_B\_2 |

【問題12-B-3】

［プログラムリスト・コンパイル結果・実行結果］

Script started on Thu Jan 26 20:12:18 2023

u20216187@gw[31]: cat e x\_12\_3[K4.c

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#define MAXLENGTH 100

unsigned char header[54];

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

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

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

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

void convert(int n);

void get\_data();

void rgb\_to\_ybr();

void processing();

void ybr\_to\_rgb();

void put\_data();

int calculate(int a, int b);

int width, height, bite;

double rgb\_con\_ybr[3][3] = {

{ 0.2990, 0.5870, 0.1140},

{-0.1687,-0.3313, 0.5000},

{ 0.5000,-0.4187,-0.0813}

};

double ybr\_con\_rgb[3][3] = {

{ 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;

}

void get\_data(){

FILE \*fp;

char filename[MAXLENGTH];

int i, c, x, y;

int filesize, offset, bite\_pixel;

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("ファイルをオーブンできません.\n");

exit(1);

}

printf("ファイルをオーブンしました.\n");

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

c = fgetc(fp);

header[i] = (unsigned char)c;

}

/\* //---------------------------------

printf("\n<ファイルタイプ>\n");

printf("header[0]=");

convert(header[0]);

printf(" header[1]=");

convert(header[1]);

printf("\n");

\*/

//---------------------------------

printf("\n<ファイルサイズ>\n");

/\*printf("header[2]=");

convert(header[2]);

printf(" header[3]=");

convert(header[3]);

printf(" header[4]=");

convert(header[4]);

printf(" header[5]=");

convert(header[5]);

\*/

filesize = calculate(5, 2);

printf("\n%dバイト\n", filesize);

/\*//---------------------------------

printf("\n<予約領域>\n");

printf("header[6]=");

convert(header[6]);

printf(" header[7]=");

convert(header[7]);

printf(" header[8]=");

convert(header[8]);

printf(" header[9]=");

convert(header[9]);

printf("\n");

\*/

//---------------------------------

printf("\n<オフセット>\n");

/\*printf("header[10]=");

convert(header[10]);

printf(" header[11]=");

convert(header[11]);

printf(" header[12]=");

convert(header[12]);

printf(" header[13]=");

convert(header[13]);\*/

offset = calculate(13, 10),

printf("\n%dバイト\n", offset);

/\*//---------------------------------

printf("\n<情報ベッダサイズ>\n");

printf("header[14]=");

convert(header[14]);

printf(" header[15]=");

convert(header[15]);

printf(" header[16]=");

convert(header[16]);

printf(" header[17]=");

convert(header[17]);

printf("\n");

\*/

//---------------------------------

printf("\n<画像の幅>\n");

/\*printf("header[18]=");

convert(header[18]);

printf(" header[19]=");

convert(header[19]);

printf(" header[20]=");

convert(header[20]);

printf(" header[21]=");

convert(header[21]);\*/

width = calculate(21, 18);

printf("\n%d画素\n",width);

//---------------------------------

printf("\n<画像の高さ>\n");

/\*printf("header[22]=");

convert(header[22]);

printf(" header[23]=");

convert(header[23]);

printf(" header[24]=");

convert(header[24]);

printf(" header[25]=");

convert(header[25]);\*/

height = calculate(25, 22);

printf("\n%dライン\n",height);

/\*//---------------------------------

printf("\n<色プレーン数>\n");

printf("header[26]=");

convert(header[26]);

printf(" header[27]=");

convert(header[27]);

printf("\n");\*/

//---------------------------------

printf("\n<1画素当たりのビット数>\n");

/\*printf("header[28]=");

convert(header[28]);

printf(" header[29]=");

convert(header[29]);\*/

bite\_pixel = calculate(29, 28);

printf("\n%dビット\n",bite\_pixel);

/\*//---------------------------------

printf("\n<圧縮方式>\n");

printf("header[30]=");

convert(header[30]);

printf(" header[31]=");

convert(header[31]);

printf(" header[32]=");

convert(header[32]);

printf(" header[33]=");

convert(header[33]);

printf("\n");

//---------------------------------

printf("\n<画像データサイズ>\n");

printf("header[34]=");

convert(header[34]);

printf(" header[35]=");

convert(header[35]);

printf(" header[36]=");

convert(header[36]);

printf(" header[37]=");

convert(header[37]);

printf("\n");

//---------------------------------

printf("\n<水平解像度>\n");

printf("header[38]=");

convert(header[38]);

printf(" header[39]=");

convert(header[39]);

printf(" header[40]=");

convert(header[40]);

printf(" header[41]=");

convert(header[41]);

printf("\n");

//---------------------------------

printf("\n<垂直解像度>\n");

printf("header[42]=");

convert(header[42]);

printf(" header[43]=");

convert(header[43]);

printf(" header[44]=");

convert(header[44]);

printf(" header[45]=");

convert(header[45]);

printf("\n");

//---------------------------------

printf("\n<色数>\n");

printf("header[46]=");

convert(header[46]);

printf(" header[47]=");

convert(header[47]);

printf(" header[48]=");

convert(header[48]);

printf(" header[49]=");

convert(header[49]);

printf("\n");

//---------------------------------

printf("\n<重要な色数>\n");

printf("header[50]=");

convert(header[50]);

printf(" header[51]=");

convert(header[51]);

printf(" header[52]=");

convert(header[52]);

printf(" header[53]=");

convert(header[53]);

printf("\n");

\*/

//---------------------------------

printf("\n<挿入ビット数>\n");

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

printf("%dバイト\n",bite);

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

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

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

c = fgetc(fp);

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

}

}

}

fclose(fp);

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

}

void convert(int n){

printf("%02x", n);

}

int calculate(int a, int b){

int i;

int value;

value = header[a];

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

value <<= 8;

value += header[i];

}

return value;

}

void processing(){

int j,i,x,y;

double n;

double avg[3][3]={{0.0,-1.0,0.0},

{-1.0,5.0,-1.0},

{0.0,-1.0,0.0}};

double sum;

printf("\n<フィルタ係数>\n");

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

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

printf("%3.6f ",avg[i][j]);

}

printf("\n");

}

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

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

if(x<=0 || x>=width || y<=0 || y>=height){

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

}else{

sum=0.0;

for(j=-1;j<=1;j++){

for(i=-1;i<=1;i++){

sum += (double)imgin[0][x+i][y+j]\*avg[i+1][j+1];

}

}

if(sum>=0.0){

n=(int)(sum+0.5);

}

else{

n=(int)(sum-0.5);

}

if(n>255){

n=255;

}

else if(n<0){

n=0;

}

imgout[0][x][y] = (unsigned char)n;

}

imgout[1][x][y] = 128;

imgout[2][x][y] = 128;

}

}

printf("入力画像データをコピーして出力画像データを作成しました.\n");

}

void put\_data(){

FILE \*fp;

char filename1[MAXLENGTH];

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 < bite; 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, j;

/\*

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 ", 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 (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)imgout[j][x][y];

else

dtemp[i][x][y] += ybr\_con\_rgb[i][j] \* (double)(imgout[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 e x\_12\_4.c -o e x\_12\_4

u20216187@gw[33]: ./e x\_12\_4

ファイル名を入力して下さい:goldhill.bmp

ファイルをオーブンしました.

<ファイルサイズ>

786488バイト

<オフセット>

54バイト

<画像の幅>

512画素

<画像の高さ>

512ライン

<1画素当たりのビット数>

24ビット

<挿入ビット数>

2バイト

goldhill.bmp をクローズしました.

<フィルタ係数>

0.000000 -1.000000 0.000000

-1.000000 5.000000 -1.000000

0.000000 -1.000000 0.000000

入力画像データをコピーして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:goldhill\_12\_B\_3.bmp

goldhill\_12\_B\_3.bmpをオープンしました.

goldhill\_12\_B\_3.bmpをクローズしました.

u20216187@gw[34]: exit

exit

Script done on Thu Jan 26 20:12:53 2023

［添付図］

|  |  |
| --- | --- |
|  |  |
| Goldhill | Goldhill\_12\_B\_3 |