

[演習 09] 濃淡変換

学生番号 : 20216187

氏名 : 劉潤之

提出日 : 2022/12/06

[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\_xxxxxxx.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 9-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 Dec  6 18:37:41 2022
u20216187@gw[31]: cat e x_09_A_01.c
```

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#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("ファイルをオープンできません.¥n");  
    exit(1);  
}  
  
printf("ファイルをオープンしました.¥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<情報ベッタサイズ>¥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, x, y;
    ///int copy[3];
    //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("¥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;
            }
        }
    }
}

*/
```

```
/*
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];
    }
}
}
*/

//int i, x, y;

//int copy[3];
// int block/*, b_height, b_width*/;

//int xi, yi;
//int ave,sum;
/* while(1){
    printf("ブロックサイズを入力してください(2,4,8,16)¥n");
    scanf("%d",&block);
    double id , cal;
    id= log2(block);
    cal =(double) id -(int)id;
    if(cal ==0 && block<=16 && block>0){
break;
    } else {
printf("入力ミス、再入力してください¥n");
    }
}

for (y = 0; y < height; y += block) {
    for (x = 0; x < width; x += block) {

        sum = 0;
```

```
    for (yi = 0; yi < block; yi++)
        for (xi = 0; xi < block; xi++)
            sum += imgin[0][x + xi][y + yi];

ave = (int)((double)sum / (double)(block * block)+0.5);

if (ave > 255) {
    ave = 255;
}
else if (ave < 0) {
    ave = 0;
}

for (yi = 0; yi < block; yi++)
    for (xi = 0; xi < block; xi++)
        imgout[0][x + xi][y + yi] = (unsigned char)ave;

}

}

for (i = 1; i < 3; i++) {
    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {

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

        }
    }
}
```



```
    }

    }*/

    /*int i, x, y, n;

int block;

double id, cal;

while (1) {

    printf("¥n 出力画像の量子化レベル数を入力して下さい(2,4,8,16):");

    scanf("%d", &block);

    id = log2(block);

    cal = (double)id - (int)id;

    if (cal == 0 && block <= 16 && block > 0) {

        break;

    }

    else {

        printf("入力ミス、再入力してください¥n");

    }

}

n = 256 /block;

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

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

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

            if (i == 0) { imgout[i][x][y] = (imgin[i][x][y] / n) * n + n / 2; }

            else { imgout[i][x][y] = 128; }

        }

    }

}

}*/
```

```
int x,y,l;

int f[256];

for(l=0;l<256;l++){

    f[l]=0;

}

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

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

        f[imgin[0][x][y]]++;

    }

}

printf("¥n<度数表>¥n");

for(l=0;l<256;l++){

printf("画素値%3d:  %5d¥n",l,f[l]);

}

//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,"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 < 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\_09\_A\_01.c -o e x\_09\_A\_01

u20216187@gw[33]: ./e x\_09\_A\_01

ファイル名を入力して下さい:goldhill.bmp

ファイルをオープンしました.

<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.

<度数表>

画素値	0:	0
画素値	1:	0
画素値	2:	0
画素値	3:	0
画素値	4:	0
画素値	5:	0
画素値	6:	0
画素値	7:	0
画素値	8:	0

画素値 9:	0
画素値 10:	0
画素値 11:	0
画素値 12:	0
画素値 13:	0
画素値 14:	0
画素値 15:	0
画素値 16:	3
画素値 17:	3
画素値 18:	8
画素値 19:	14
画素値 20:	34
画素値 21:	51
画素値 22:	100
画素値 23:	159
画素値 24:	235
画素値 25:	283
画素値 26:	390
画素値 27:	430
画素値 28:	543
画素値 29:	577
画素値 30:	607
画素値 31:	609
画素値 32:	587
画素値 33:	601
画素値 34:	540
画素値 35:	512
画素値 36:	505
画素値 37:	560
画素値 38:	568
画素値 39:	622
画素値 40:	704
画素値 41:	847
画素値 42:	966
画素値 43:	1048
画素値 44:	1134
画素値 45:	1237
画素値 46:	1321
画素値 47:	1417
画素値 48:	1538
画素値 49:	1536
画素値 50:	1582
画素値 51:	1601
画素値 52:	1576
画素値 53:	1690
画素値 54:	1651
画素値 55:	1671
画素値 56:	1683
画素値 57:	1719
画素値 58:	1627
画素値 59:	1696
画素値 60:	1662
画素値 61:	1603
画素値 62:	1645
画素値 63:	1696
画素値 64:	1690
画素値 65:	1730

画素値 66:	1811
画素値 67:	1802
画素値 68:	1900
画素値 69:	1869
画素値 70:	1832
画素値 71:	1912
画素値 72:	1871
画素値 73:	1948
画素値 74:	1993
画素値 75:	1935
画素値 76:	2046
画素値 77:	1983
画素値 78:	1958
画素値 79:	1905
画素値 80:	1943
画素値 81:	1920
画素値 82:	1880
画素値 83:	1959
画素値 84:	2001
画素値 85:	1999
画素値 86:	1925
画素値 87:	1973
画素値 88:	1975
画素値 89:	1900
画素値 90:	1981
画素値 91:	2079
画素値 92:	2014
画素値 93:	2014
画素値 94:	2038
画素値 95:	2096
画素値 96:	2156
画素値 97:	2128
画素値 98:	2142
画素値 99:	2177
画素値 100:	2127
画素値 101:	2306
画素値 102:	2263
画素値 103:	2307
画素値 104:	2296
画素値 105:	2365
画素値 106:	2356
画素値 107:	2394
画素値 108:	2474
画素値 109:	2491
画素値 110:	2431
画素値 111:	2400
画素値 112:	2477
画素値 113:	2560
画素値 114:	2482
画素値 115:	2492
画素値 116:	2480
画素値 117:	2511
画素値 118:	2488
画素値 119:	2618
画素値 120:	2442
画素値 121:	2375
画素値 122:	2367

画素値 123:	2330
画素値 124:	2264
画素値 125:	2180
画素値 126:	2058
画素値 127:	2031
画素値 128:	1988
画素値 129:	1848
画素値 130:	1796
画素値 131:	1710
画素値 132:	1635
画素値 133:	1505
画素値 134:	1523
画素値 135:	1414
画素値 136:	1277
画素値 137:	1344
画素値 138:	1315
画素値 139:	1297
画素値 140:	1232
画素値 141:	1173
画素値 142:	1263
画素値 143:	1240
画素値 144:	1195
画素値 145:	1147
画素値 146:	1172
画素値 147:	1129
画素値 148:	1087
画素値 149:	1079
画素値 150:	1117
画素値 151:	1041
画素値 152:	996
画素値 153:	1013
画素値 154:	955
画素値 155:	943
画素値 156:	896
画素値 157:	872
画素値 158:	854
画素値 159:	819
画素値 160:	825
画素値 161:	777
画素値 162:	827
画素値 163:	742
画素値 164:	712
画素値 165:	714
画素値 166:	710
画素値 167:	647
画素値 168:	658
画素値 169:	666
画素値 170:	661
画素値 171:	615
画素値 172:	602
画素値 173:	511
画素値 174:	500
画素値 175:	497
画素値 176:	463
画素値 177:	441
画素値 178:	430
画素値 179:	449

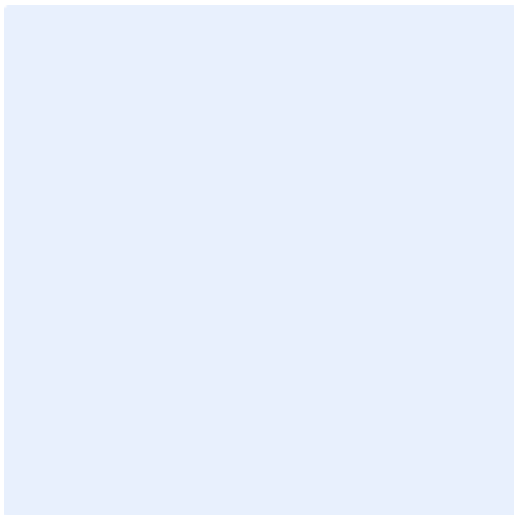
画素値 180:	458
画素値 181:	399
画素値 182:	431
画素値 183:	426
画素値 184:	448
画素値 185:	424
画素値 186:	406
画素値 187:	399
画素値 188:	415
画素値 189:	398
画素値 190:	403
画素値 191:	436
画素値 192:	439
画素値 193:	443
画素値 194:	455
画素値 195:	444
画素値 196:	469
画素値 197:	525
画素値 198:	501
画素値 199:	548
画素値 200:	479
画素値 201:	489
画素値 202:	433
画素値 203:	445
画素値 204:	472
画素値 205:	494
画素値 206:	455
画素値 207:	465
画素値 208:	429
画素値 209:	396
画素値 210:	387
画素値 211:	399
画素値 212:	352
画素値 213:	340
画素値 214:	343
画素値 215:	266
画素値 216:	276
画素値 217:	257
画素値 218:	303
画素値 219:	280
画素値 220:	274
画素値 221:	320
画素値 222:	382
画素値 223:	440
画素値 224:	505
画素値 225:	689
画素値 226:	889
画素値 227:	1037
画素値 228:	1068
画素値 229:	1335
画素値 230:	1447
画素値 231:	1370
画素値 232:	1193
画素値 233:	927
画素値 234:	546
画素値 235:	532
画素値 236:	0

```
画素値 237:      0
画素値 238:      0
画素値 239:      0
画素値 240:      0
画素値 241:      0
画素値 242:      0
画素値 243:      0
画素値 244:      0
画素値 245:      0
画素値 246:      0
画素値 247:      0
画素値 248:      0
画素値 249:      0
画素値 250:      0
画素値 251:      0
画素値 252:      0
画素値 253:      0
画素値 254:      0
画素値 255:      0
u20216187@gw[34]: exit
```

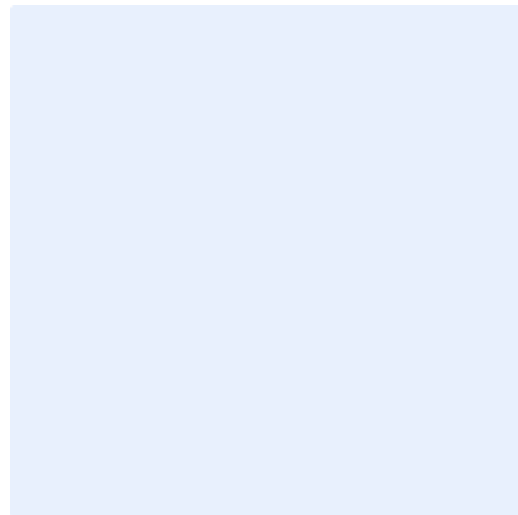
exit

Script done on Tue Dec 6 18:38:04 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 9-A-2】

[プログラムリスト・コンパイル結果・実行結果]

```
Script started on Tue Dec  6 18:42:42 2022
u20216187@gw[31]: cat e_x_09_A_02.c
```

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#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("ファイルをオープンできません.¥n");  
    exit(1);  
}  
  
printf("ファイルをオープンしました.¥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<情報ベッタサイズ>¥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, x, y;
    ///int copy[3];
    //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("¥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;
            }
        }
    }
}

*/
```

```
/*
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];
    }
}
}
*/

//int i, x, y;

//int copy[3];
// int block/*, b_height, b_width*/;


//int xi, yi;
//int ave,sum;
/* while(1){
    printf("ブロックサイズを入力してください(2,4,8,16)¥n");
    scanf("%d",&block);
    double id , cal;
    id= log2(block);
    cal =(double) id -(int)id;
    if(cal ==0 && block<=16 && block>0){
break;
    } else {
printf("入力ミス、再入力してください¥n");
    }
}

for (y = 0; y < height; y += block) {
    for (x = 0; x < width; x += block) {

        sum = 0;
```

```
    for (yi = 0; yi < block; yi++)
        for (xi = 0; xi < block; xi++)
            sum += imgin[0][x + xi][y + yi];

ave = (int)((double)sum / (double)(block * block)+0.5);

if (ave > 255) {
    ave = 255;
}
else if (ave < 0) {
    ave = 0;
}

for (yi = 0; yi < block; yi++)
    for (xi = 0; xi < block; xi++)
        imgout[0][x + xi][y + yi] = (unsigned char)ave;

}

}

for (i = 1; i < 3; i++) {
    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {

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

        }
    }
}
```

```
    }

    }*/

    /*int i, x, y, n;

int block;

double id, cal;

while (1) {

    printf("¥n 出力画像の量子化レベル数を入力して下さい(2,4,8,16):");

    scanf("%d", &block);

    id = log2(block);

    cal = (double)id - (int)id;

    if (cal == 0 && block <= 16 && block > 0) {

        break;

    }

    else {

        printf("入力ミス、再入力してください¥n");

    }

}

n = 256 /block;

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

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

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

            if (i == 0) { imgout[i][x][y] = (imgin[i][x][y] / n) * n + n / 2; }

            else { imgout[i][x][y] = 128; }

        }

    }

}

}*/
```



```
int x,y,l,max=0;

int f[256];

int fchange[256];

int maxlevel=0;

int *p;


for(l=0;l<256;l++){

    f[l]=0;

}

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

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

        f[imgin[0][x][y]]++;

    }

}

p=&f[0];

for(l=0;l<256;l++){

    if (*p>max) { max=*p;

        maxlevel=l;

        p++;

    }else{

        p++;

    }

}


printf("%n<最大度数>%n");
```

```
printf("画素値 %3d:%5d",maxlevel,max);

for(l=0;l<256;l++){

    fchange[l]=(int)((double)f[l]/(double)max*100.0+0.5);

}

printf("¥n<度数表>¥n");

for(l=0;l<256;l++){

    printf("画素値%3d:  %5d-->%5d¥n",l,f[l],fchange[l]);

}

}
```

```
/*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]: e [K
u20216187@gw[32]: gcc -Wall e x_09_A_02.c -o e x_09_A_02
u20216187@gw[33]: ./e x_09_A_02
```

ファイル名を入力して下さい:goldhill.bmp  
 ファイルをオープンしました.

<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.

<最大度数>

画素値 119: 2618

<度数表>

画素値 0:	0-->	0
画素値 1:	0-->	0
画素値 2:	0-->	0
画素値 3:	0-->	0
画素値 4:	0-->	0
画素値 5:	0-->	0
画素値 6:	0-->	0
画素値 7:	0-->	0
画素値 8:	0-->	0
画素値 9:	0-->	0
画素値 10:	0-->	0
画素値 11:	0-->	0
画素値 12:	0-->	0
画素値 13:	0-->	0
画素値 14:	0-->	0
画素値 15:	0-->	0
画素値 16:	3-->	0
画素値 17:	3-->	0
画素値 18:	8-->	0
画素値 19:	14-->	1
画素値 20:	34-->	1
画素値 21:	51-->	2
画素値 22:	100-->	4
画素値 23:	159-->	6
画素値 24:	235-->	9

画素値 25:	283-->	11
画素値 26:	390-->	15
画素値 27:	430-->	16
画素値 28:	543-->	21
画素値 29:	577-->	22
画素値 30:	607-->	23
画素値 31:	609-->	23
画素値 32:	587-->	22
画素値 33:	601-->	23
画素値 34:	540-->	21
画素値 35:	512-->	20
画素値 36:	505-->	19
画素値 37:	560-->	21
画素値 38:	568-->	22
画素値 39:	622-->	24
画素値 40:	704-->	27
画素値 41:	847-->	32
画素値 42:	966-->	37
画素値 43:	1048-->	40
画素値 44:	1134-->	43
画素値 45:	1237-->	47
画素値 46:	1321-->	50
画素値 47:	1417-->	54
画素値 48:	1538-->	59
画素値 49:	1536-->	59
画素値 50:	1582-->	60
画素値 51:	1601-->	61
画素値 52:	1576-->	60
画素値 53:	1690-->	65
画素値 54:	1651-->	63
画素値 55:	1671-->	64
画素値 56:	1683-->	64
画素値 57:	1719-->	66
画素値 58:	1627-->	62
画素値 59:	1696-->	65
画素値 60:	1662-->	63
画素値 61:	1603-->	61
画素値 62:	1645-->	63
画素値 63:	1696-->	65
画素値 64:	1690-->	65
画素値 65:	1730-->	66
画素値 66:	1811-->	69
画素値 67:	1802-->	69
画素値 68:	1900-->	73
画素値 69:	1869-->	71
画素値 70:	1832-->	70
画素値 71:	1912-->	73
画素値 72:	1871-->	71
画素値 73:	1948-->	74
画素値 74:	1993-->	76
画素値 75:	1935-->	74
画素値 76:	2046-->	78
画素値 77:	1983-->	76
画素値 78:	1958-->	75
画素値 79:	1905-->	73
画素値 80:	1943-->	74
画素値 81:	1920-->	73

画素値 82:	1880-->	72
画素値 83:	1959-->	75
画素値 84:	2001-->	76
画素値 85:	1999-->	76
画素値 86:	1925-->	74
画素値 87:	1973-->	75
画素値 88:	1975-->	75
画素値 89:	1900-->	73
画素値 90:	1981-->	76
画素値 91:	2079-->	79
画素値 92:	2014-->	77
画素値 93:	2014-->	77
画素値 94:	2038-->	78
画素値 95:	2096-->	80
画素値 96:	2156-->	82
画素値 97:	2128-->	81
画素値 98:	2142-->	82
画素値 99:	2177-->	83
画素値 100:	2127-->	81
画素値 101:	2306-->	88
画素値 102:	2263-->	86
画素値 103:	2307-->	88
画素値 104:	2296-->	88
画素値 105:	2365-->	90
画素値 106:	2356-->	90
画素値 107:	2394-->	91
画素値 108:	2474-->	94
画素値 109:	2491-->	95
画素値 110:	2431-->	93
画素値 111:	2400-->	92
画素値 112:	2477-->	95
画素値 113:	2560-->	98
画素値 114:	2482-->	95
画素値 115:	2492-->	95
画素値 116:	2480-->	95
画素値 117:	2511-->	96
画素値 118:	2488-->	95
画素値 119:	2618-->	100
画素値 120:	2442-->	93
画素値 121:	2375-->	91
画素値 122:	2367-->	90
画素値 123:	2330-->	89
画素値 124:	2264-->	86
画素値 125:	2180-->	83
画素値 126:	2058-->	79
画素値 127:	2031-->	78
画素値 128:	1988-->	76
画素値 129:	1848-->	71
画素値 130:	1796-->	69
画素値 131:	1710-->	65
画素値 132:	1635-->	62
画素値 133:	1505-->	57
画素値 134:	1523-->	58
画素値 135:	1414-->	54
画素値 136:	1277-->	49
画素値 137:	1344-->	51
画素値 138:	1315-->	50

画素値 139:	1297-->	50
画素値 140:	1232-->	47
画素値 141:	1173-->	45
画素値 142:	1263-->	48
画素値 143:	1240-->	47
画素値 144:	1195-->	46
画素値 145:	1147-->	44
画素値 146:	1172-->	45
画素値 147:	1129-->	43
画素値 148:	1087-->	42
画素値 149:	1079-->	41
画素値 150:	1117-->	43
画素値 151:	1041-->	40
画素値 152:	996-->	38
画素値 153:	1013-->	39
画素値 154:	955-->	36
画素値 155:	943-->	36
画素値 156:	896-->	34
画素値 157:	872-->	33
画素値 158:	854-->	33
画素値 159:	819-->	31
画素値 160:	825-->	32
画素値 161:	777-->	30
画素値 162:	827-->	32
画素値 163:	742-->	28
画素値 164:	712-->	27
画素値 165:	714-->	27
画素値 166:	710-->	27
画素値 167:	647-->	25
画素値 168:	658-->	25
画素値 169:	666-->	25
画素値 170:	661-->	25
画素値 171:	615-->	23
画素値 172:	602-->	23
画素値 173:	511-->	20
画素値 174:	500-->	19
画素値 175:	497-->	19
画素値 176:	463-->	18
画素値 177:	441-->	17
画素値 178:	430-->	16
画素値 179:	449-->	17
画素値 180:	458-->	17
画素値 181:	399-->	15
画素値 182:	431-->	16
画素値 183:	426-->	16
画素値 184:	448-->	17
画素値 185:	424-->	16
画素値 186:	406-->	16
画素値 187:	399-->	15
画素値 188:	415-->	16
画素値 189:	398-->	15
画素値 190:	403-->	15
画素値 191:	436-->	17
画素値 192:	439-->	17
画素値 193:	443-->	17
画素値 194:	455-->	17
画素値 195:	444-->	17

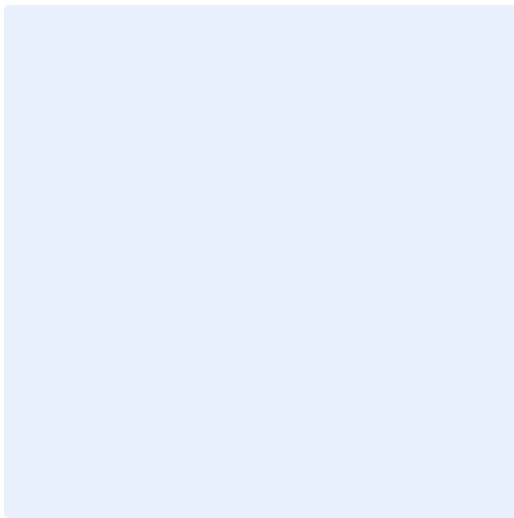
画素値 196:	469-->	18
画素値 197:	525-->	20
画素値 198:	501-->	19
画素値 199:	548-->	21
画素値 200:	479-->	18
画素値 201:	489-->	19
画素値 202:	433-->	17
画素値 203:	445-->	17
画素値 204:	472-->	18
画素値 205:	494-->	19
画素値 206:	455-->	17
画素値 207:	465-->	18
画素値 208:	429-->	16
画素値 209:	396-->	15
画素値 210:	387-->	15
画素値 211:	399-->	15
画素値 212:	352-->	13
画素値 213:	340-->	13
画素値 214:	343-->	13
画素値 215:	266-->	10
画素値 216:	276-->	11
画素値 217:	257-->	10
画素値 218:	303-->	12
画素値 219:	280-->	11
画素値 220:	274-->	10
画素値 221:	320-->	12
画素値 222:	382-->	15
画素値 223:	440-->	17
画素値 224:	505-->	19
画素値 225:	689-->	26
画素値 226:	889-->	34
画素値 227:	1037-->	40
画素値 228:	1068-->	41
画素値 229:	1335-->	51
画素値 230:	1447-->	55
画素値 231:	1370-->	52
画素値 232:	1193-->	46
画素値 233:	927-->	35
画素値 234:	546-->	21
画素値 235:	532-->	20
画素値 236:	0-->	0
画素値 237:	0-->	0
画素値 238:	0-->	0
画素値 239:	0-->	0
画素値 240:	0-->	0
画素値 241:	0-->	0
画素値 242:	0-->	0
画素値 243:	0-->	0
画素値 244:	0-->	0
画素値 245:	0-->	0
画素値 246:	0-->	0
画素値 247:	0-->	0
画素値 248:	0-->	0
画素値 249:	0-->	0
画素値 250:	0-->	0
画素値 251:	0-->	0
画素値 252:	0-->	0

```
画素値 253:      0-->    0  
画素値 254:      0-->    0  
画素値 255:      0-->    0  
u20216187@gw[34]: exit
```

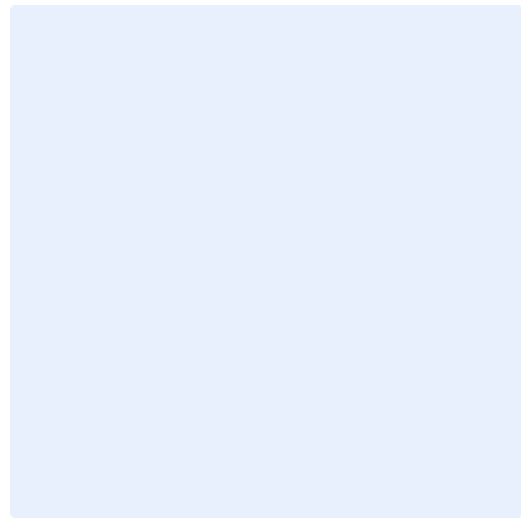
exit

Script done on Tue Dec 6 18:43:15 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 9-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 Dec  6 18:49:28 2022
```

```
u20216187@gw[31]: cat 09 _B_01.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 value(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("ファイルをオープンできません.¥n");
    exit(1);
}

printf("ファイルをオープンしました.¥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 = value(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 = value(13, 10),  
printf("¥n%d バイト¥n", offset);  
/*//-----  
printf("¥n<情報ベッタサイズ>¥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 = value(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 = value(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 = value(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 value(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 x,y,l,max=0;//l は度数
    int f[256];//画素値を記録
    int fchange[256];//変えた画素値を記録
    int maxlevel=0;//最大値
    int *p;//f[]

    for(l=0;l<256;l++){
        f[l]=0;
    }
    for(x=0;x<width;x++){
        for(y=0;y<height;y++){

            f[imgin[0][x][y]]++;
        }
    }
    p=&f[0];
    for(l=0;l<256;l++){
        if (*p>max) { max=*p;
            maxlevel=l;
        }
        p++;
    }
}
```

```
printf("¥n<最大度数>¥n");

printf("画素値 %3d:%5d",maxlevel,max);

for(l=0;l<256;l++){

    fchange[l]=(int)((double)f[l]/(double)max*100.0+0.5);

}

printf("¥n<度数表>¥n");

for(l=0;l<256;l++){

    printf("画素値%3d:  %5d-->%5d¥n",l,f[l],fchange[l]);

}

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

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

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

            if(i!=0)

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

            else

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

        }

    }

}

for(y = height - 100; y < height; y++){

    for(x = 0; x < 256; x++){

        imgout[0][x][y] = 0;//Y 0 Cb 128 Cr 128 背景

    }

}
```

```
for(l = 0; l < 256; l++){
    for(int i = 0; i <= fchange[l]; i++)

        imgout[2][l][height - i] = 255;//Y 0 Cb 128 Cr 255 赤線
    }
/*
while(1){
    printf("※n 出力画像の量子化レベル数を入力して下さい(2,4,8,16):");
    scanf("%d",&level);

    if(level == 2 || level == 4 || level == 8 || level == 16)
        break;
}

n = 256 /level;

for(y = 0; y < height; y++){
    for(x = 0; x < width; x++){
        imgout[0][x][y] = (imgin[0][x][y] / n) * n + n / 2;
        imgout[1][x][y] = 128;
        imgout[2][x][y] = 128;
    }
}*/
//int copy[3];
//int block/*, b_height, b_width*/;

//int xi,yi;
//int ave, sum;
```

```
//printf("¥n ブロックサイズを入力して下さい(2,4,8,16) : ");
//scanf("%d",&block);

/*
for(y = 0; y < height; y += block){
    for(x = 0; x < width; x += block){

        sum = 0;

        for (yi = 0; yi < block; yi++)
            for (xi = 0; xi < block; xi++)
                sum += imgin[0][x + xi][y + yi];

        ave = (int)((double) sum / (double)(block * block));

        if(ave > 255){
            ave = 255;
        }else if(ave < 0){
            ave = 0;
        }

        for (yi = 0; yi < block; yi++)
            for (xi = 0; xi < block; xi++)
                imgout[0][x + xi][y + yi] = ave;

    }
}
```

```
}

for(i = 1; i < 3; i++){
    for(y = 0; y < height; y++){
        for(x = 0; x < width; x++){
            imgout[i][x][y] = 128;
        }
    }
}

*/

//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("%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;

            }

        }

    }

}

}

*/

/*

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];  
    }  
}  
}  
*/  
  
//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 09 _B_01.c -o 09 _B_01
```

u20216187@gw[33]: ./09\_B\_01

ファイル名を入力して下さい:goldhill.bmp  
 ファイルをオープンしました.

<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.

<最大度数>

画素値 119: 2618

<度数表>

画素値 0:	0-->	0
画素値 1:	0-->	0
画素値 2:	0-->	0
画素値 3:	0-->	0
画素値 4:	0-->	0
画素値 5:	0-->	0
画素値 6:	0-->	0
画素値 7:	0-->	0
画素値 8:	0-->	0
画素値 9:	0-->	0
画素値 10:	0-->	0
画素値 11:	0-->	0
画素値 12:	0-->	0
画素値 13:	0-->	0
画素値 14:	0-->	0
画素値 15:	0-->	0
画素値 16:	3-->	0
画素値 17:	3-->	0
画素値 18:	8-->	0
画素値 19:	14-->	1
画素値 20:	34-->	1
画素値 21:	51-->	2
画素値 22:	100-->	4

画素値 23:	159-->	6
画素値 24:	235-->	9
画素値 25:	283-->	11
画素値 26:	390-->	15
画素値 27:	430-->	16
画素値 28:	543-->	21
画素値 29:	577-->	22
画素値 30:	607-->	23
画素値 31:	609-->	23
画素値 32:	587-->	22
画素値 33:	601-->	23
画素値 34:	540-->	21
画素値 35:	512-->	20
画素値 36:	505-->	19
画素値 37:	560-->	21
画素値 38:	568-->	22
画素値 39:	622-->	24
画素値 40:	704-->	27
画素値 41:	847-->	32
画素値 42:	966-->	37
画素値 43:	1048-->	40
画素値 44:	1134-->	43
画素値 45:	1237-->	47
画素値 46:	1321-->	50
画素値 47:	1417-->	54
画素値 48:	1538-->	59
画素値 49:	1536-->	59
画素値 50:	1582-->	60
画素値 51:	1601-->	61
画素値 52:	1576-->	60
画素値 53:	1690-->	65
画素値 54:	1651-->	63
画素値 55:	1671-->	64
画素値 56:	1683-->	64
画素値 57:	1719-->	66
画素値 58:	1627-->	62
画素値 59:	1696-->	65
画素値 60:	1662-->	63
画素値 61:	1603-->	61
画素値 62:	1645-->	63
画素値 63:	1696-->	65
画素値 64:	1690-->	65
画素値 65:	1730-->	66
画素値 66:	1811-->	69
画素値 67:	1802-->	69
画素値 68:	1900-->	73
画素値 69:	1869-->	71
画素値 70:	1832-->	70
画素値 71:	1912-->	73
画素値 72:	1871-->	71
画素値 73:	1948-->	74
画素値 74:	1993-->	76
画素値 75:	1935-->	74
画素値 76:	2046-->	78
画素値 77:	1983-->	76
画素値 78:	1958-->	75
画素値 79:	1905-->	73

画素値 80:	1943-->	74
画素値 81:	1920-->	73
画素値 82:	1880-->	72
画素値 83:	1959-->	75
画素値 84:	2001-->	76
画素値 85:	1999-->	76
画素値 86:	1925-->	74
画素値 87:	1973-->	75
画素値 88:	1975-->	75
画素値 89:	1900-->	73
画素値 90:	1981-->	76
画素値 91:	2079-->	79
画素値 92:	2014-->	77
画素値 93:	2014-->	77
画素値 94:	2038-->	78
画素値 95:	2096-->	80
画素値 96:	2156-->	82
画素値 97:	2128-->	81
画素値 98:	2142-->	82
画素値 99:	2177-->	83
画素値 100:	2127-->	81
画素値 101:	2306-->	88
画素値 102:	2263-->	86
画素値 103:	2307-->	88
画素値 104:	2296-->	88
画素値 105:	2365-->	90
画素値 106:	2356-->	90
画素値 107:	2394-->	91
画素値 108:	2474-->	94
画素値 109:	2491-->	95
画素値 110:	2431-->	93
画素値 111:	2400-->	92
画素値 112:	2477-->	95
画素値 113:	2560-->	98
画素値 114:	2482-->	95
画素値 115:	2492-->	95
画素値 116:	2480-->	95
画素値 117:	2511-->	96
画素値 118:	2488-->	95
画素値 119:	2618-->	100
画素値 120:	2442-->	93
画素値 121:	2375-->	91
画素値 122:	2367-->	90
画素値 123:	2330-->	89
画素値 124:	2264-->	86
画素値 125:	2180-->	83
画素値 126:	2058-->	79
画素値 127:	2031-->	78
画素値 128:	1988-->	76
画素値 129:	1848-->	71
画素値 130:	1796-->	69
画素値 131:	1710-->	65
画素値 132:	1635-->	62
画素値 133:	1505-->	57
画素値 134:	1523-->	58
画素値 135:	1414-->	54
画素値 136:	1277-->	49

画素値 137:	1344-->	51
画素値 138:	1315-->	50
画素値 139:	1297-->	50
画素値 140:	1232-->	47
画素値 141:	1173-->	45
画素値 142:	1263-->	48
画素値 143:	1240-->	47
画素値 144:	1195-->	46
画素値 145:	1147-->	44
画素値 146:	1172-->	45
画素値 147:	1129-->	43
画素値 148:	1087-->	42
画素値 149:	1079-->	41
画素値 150:	1117-->	43
画素値 151:	1041-->	40
画素値 152:	996-->	38
画素値 153:	1013-->	39
画素値 154:	955-->	36
画素値 155:	943-->	36
画素値 156:	896-->	34
画素値 157:	872-->	33
画素値 158:	854-->	33
画素値 159:	819-->	31
画素値 160:	825-->	32
画素値 161:	777-->	30
画素値 162:	827-->	32
画素値 163:	742-->	28
画素値 164:	712-->	27
画素値 165:	714-->	27
画素値 166:	710-->	27
画素値 167:	647-->	25
画素値 168:	658-->	25
画素値 169:	666-->	25
画素値 170:	661-->	25
画素値 171:	615-->	23
画素値 172:	602-->	23
画素値 173:	511-->	20
画素値 174:	500-->	19
画素値 175:	497-->	19
画素値 176:	463-->	18
画素値 177:	441-->	17
画素値 178:	430-->	16
画素値 179:	449-->	17
画素値 180:	458-->	17
画素値 181:	399-->	15
画素値 182:	431-->	16
画素値 183:	426-->	16
画素値 184:	448-->	17
画素値 185:	424-->	16
画素値 186:	406-->	16
画素値 187:	399-->	15
画素値 188:	415-->	16
画素値 189:	398-->	15
画素値 190:	403-->	15
画素値 191:	436-->	17
画素値 192:	439-->	17
画素値 193:	443-->	17

画素値 194:	455-->	17
画素値 195:	444-->	17
画素値 196:	469-->	18
画素値 197:	525-->	20
画素値 198:	501-->	19
画素値 199:	548-->	21
画素値 200:	479-->	18
画素値 201:	489-->	19
画素値 202:	433-->	17
画素値 203:	445-->	17
画素値 204:	472-->	18
画素値 205:	494-->	19
画素値 206:	455-->	17
画素値 207:	465-->	18
画素値 208:	429-->	16
画素値 209:	396-->	15
画素値 210:	387-->	15
画素値 211:	399-->	15
画素値 212:	352-->	13
画素値 213:	340-->	13
画素値 214:	343-->	13
画素値 215:	266-->	10
画素値 216:	276-->	11
画素値 217:	257-->	10
画素値 218:	303-->	12
画素値 219:	280-->	11
画素値 220:	274-->	10
画素値 221:	320-->	12
画素値 222:	382-->	15
画素値 223:	440-->	17
画素値 224:	505-->	19
画素値 225:	689-->	26
画素値 226:	889-->	34
画素値 227:	1037-->	40
画素値 228:	1068-->	41
画素値 229:	1335-->	51
画素値 230:	1447-->	55
画素値 231:	1370-->	52
画素値 232:	1193-->	46
画素値 233:	927-->	35
画素値 234:	546-->	21
画素値 235:	532-->	20
画素値 236:	0-->	0
画素値 237:	0-->	0
画素値 238:	0-->	0
画素値 239:	0-->	0
画素値 240:	0-->	0
画素値 241:	0-->	0
画素値 242:	0-->	0
画素値 243:	0-->	0
画素値 244:	0-->	0
画素値 245:	0-->	0
画素値 246:	0-->	0
画素値 247:	0-->	0
画素値 248:	0-->	0
画素値 249:	0-->	0
画素値 250:	0-->	0

```
画素値 251:      0-->    0
画素値 252:      0-->    0
画素値 253:      0-->    0
画素値 254:      0-->    0
画素値 255:      0-->    0
出力ファイル名を入力して下さい:goldhill_09_B_1.bmp
goldhill_09_B_1.bmp をオープンしました.
goldhill_09_B_1.bmp をクローズしました.
u20216187@gw[34]: exit
```

```
exit
```

```
Script done on Tue Dec  6 18:50:26 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。



goldhill\_09\_B\_1.bmp



## 【問題 9-B-2】

[プログラムリスト・コンパイル結果・実行結果]

```
Script started on Tue Dec  6 18:54:36 2022
u20216187@gw[31]: cat 09 _B_02.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 value(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("ファイルをオープンできません.¥n");
    exit(1);
}

printf("ファイルをオープンしました.¥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 = value(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 = value(13, 10),  
printf("¥n%d バイト¥n", offset);  
/*//-----  
printf("¥n<情報ベッタサイズ>¥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 = value(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 = value(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 = value(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 value(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 x,y;
    //int f[256];//画素値を記録
    //int fchange[256];//変えた画素値を記録
    //int maxlevel=0;//最大値
    //int *p;
    int max,min;
    min = imgin[0][0][0];
    max = imgin[0][0][0];

    for(y = 0; y < height; y++){
        for(x = 0; x < width; x++){
            if (imgin[0][x][y]>=max) {
                max=imgin[0][x][y];

            }else if(imgin[0][x][y]<=min){
                min=imgin[0][x][y];
            }

        }

    }
```

```
}

printf("¥n");

printf("最大画素値 : %3d¥n", max);

printf("最小画素値 : %3d¥n", min);

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

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

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

            if(i==0){

                imgout[i][x][y] = (int)(255.0 / (double)(max - min) *
(double)(imgin[i][x][y] - min) + 0.5);

            }

            else {

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

            }

        }

    }

}

}
```

```
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, "wb");

}
```

```
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;100[K00[Kl 09 _B_02.c -o 09 _B_02

u20216187@gw[33]: ./09 _B_02

ファイル名を入力して下さい:goldhill09.bmp
ファイルを開きました.

<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

goldhill09.bmp をクローズしました.

最大画素値:181
最小画素値: 72
出力ファイル名を入力して下さい:goldhill_09_B_2.bmp
goldhill_09_B_2.bmp を開きました.
goldhill_09_B_2.bmp をクローズしました.
u20216187@gw[34]: exit

exit

Script done on Tue Dec 6 18:55:46 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。



goldhill\_09\_B\_2.bmp