［演習 05］色空間変換

学生番号 ： 20216187
氏名 ： 劉潤之
提出日 ： 2022/10/25

［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 5-A-1】

［プログラムリスト・コンパイル結果・実行結果］

```
Script started on Tue Oct 25 15:26:54 2022
u20216187@gw[31]: cat 05_A_01.c

/*--- ex_05_ref_1 ---*/


#include <stdio.h>


#define CH 3

#define Ych 0

#define ROW 3

#define COL 3


int main(void){

  double rgb_to_ybr[ROW][COL] = {

    { 0.2990, 0.5870, 0.1140},

    {-0.1687,-0.3313, 0.5000},

    { 0.5000,-0.4187,-0.0813}

  };


  double ybr_to_rgb[ROW][COL] = {

    { 1.0000, 0.0000, 1.4020},

    { 1.0000,-0.3441,-0.7141},

    { 1.0000, 1.7720, 0.0000}

  };


  char char_rgb[CH][2] = { "R", "G", "B"};//入力表示するため文字列

  char char_ybr[CH][3] = { "Y", "Cb", "Cr"};
```

```
int rgb_in[CH]; // 入力 RGB 信号(整数値)
int ybr[CH]; // YCbCr 信号(整数値)
int rgb_out[CH]; // 出力 RGB 信号(整数値)
double dtemp[CH];
int ch;
int i;
int itemp;

printf("信号値を入力して下さい(0 以上 255 以下の整数)¥n");

for(ch = 0; ch < CH; ch++){
  while (1){
    printf("%-2s : ", char_rgb[ch]);
    scanf("%d", &rgb_in[ch]);
    if(rgb_in[ch] >= 0 && rgb_in[ch] <= 255){
      break;
    }
  }
}


//--- 入力 RGB 信号(整数値)の表示 ---
printf("¥n<入力 RGB 信号(整数値)>¥n");
for (ch = 0; ch < CH; ch++){
  printf("%-2s : %4d¥n", char_rgb[ch], rgb_in[ch]);
}
//--- RGB 信号(整数値)から YCbCr 信号(実数値)への変換 ---
for (ch = 0; ch < CH; ch++){
```

```
  dtemp[ch] = 0.0;
  for (i = 0; i < COL; i++)
    dtemp[ch] += rgb_to_ybr[ch][i] * (double)rgb_in[i];
}

//--- 変換された YCbCr 信号(実数値)の表示 ---
printf("¥n<変換された YCbCr 信号(実数値)>¥n");
for (ch = 0; ch < CH; ch++)
  printf("%-2s : %10.4f¥n", char_ybr[ch], dtemp[ch]);

//--- YCbCr 信号(実数値)から YCbCr 信号(整数値)への変換 ---
for (ch = 0; ch < CH; ch++){
  // 四捨五入
  if (dtemp[ch] > 0.0)
    itemp = (int)(dtemp[ch] + 0.5);
  else
    itemp = (int)(dtemp[ch] - 0.5);

  // Cb,Cr 信号にオフセット値 128 を加える
  if (ch != Ych)
    itemp += 128;

  // 信号値を 0~255 の範囲内に制限する
  if (itemp > 255)
    itemp = 255;
  else if (itemp < 0)
    itemp = 0;
```

```
  // YCbCr 信号値(整数値)を格納
  ybr[ch] = itemp;
}

//--- 変換された YCbCr 信号(整数値)の表示 ---
printf("¥n<変換された YCbCr 信号(整数値)>¥n");
for (ch = 0; ch < CH; ch++)
  printf("%-2s : %4d¥n", char_ybr[ch], ybr[ch]);

//--- YCbCr 信号(整数値)から RGB 信号(実数値)への変換 ---
for (ch = 0; ch < CH; ch++){
  dtemp[ch] = 0.0;
  for (i = 0; i < COL; i++){
    if (i == Ych)
      dtemp[ch] += ybr_to_rgb[ch][i] * (double)ybr[i];
    else // Cb,Cr 信号は,オフセット値 128 を減ずる
      dtemp[ch] += ybr_to_rgb[ch][i] * (double)(ybr[i] - 128);
  }
}

//--- 変換された RGB 信号(実数値)の表示 ---
printf("¥n<変換された RGB 信号(実数値)>¥n");
for (ch = 0; ch < CH; ch++)
  printf("%-2s : %10.4f¥n", char_rgb[ch], dtemp[ch]);

//--- RGB 信号(実数値)から RGB 信号(整数値)への変換 ---
for (ch = 0; ch < CH; ch++){
  // 四捨五入
  if (dtemp[ch] > 0.0)
```

```
      itemp = (int)(dtemp[ch] + 0.5);
    else
      itemp = (int)(dtemp[ch] - 0.5);

  // 信号値を 0~255 の範囲内に制限する
    if (itemp > 255)
      itemp = 255;
    else if (itemp < 0)
      itemp = 0;

  // RGB 信号値(整数値)を格納
    rgb_out[ch] = itemp;
  }

  //--- 出力 RGB 信号(整数値)の表示 ---
  printf("¥n<出力 RGB 信号(整数値)>¥n");
  for (ch = 0; ch < CH; ch++)
    printf("%-2s : %4d¥n", char_rgb[ch], rgb_out[ch]);
  return 0;
}
u20216187@gw[32]: gcc -Wall 05_A_01.c -o 05_ A_01
u20216187@gw[33]: ./05_A_01
信号値を入力して下さい(0 以上 255 以下の整数)
R  : 0
G  : 255
B  : 0

<入力 RGB 信号(整数値)>
R  :    0
G  :  255
B  :    0

<変換された YCbCr 信号(実数値)>
Y  :   149.6850
```

```
Cb :   -84.4815
Cr :  -106.7685

<変換された YCbCr 信号(整数値)>
Y  :  150
Cb :   44
Cr :   21

<変換された RGB 信号(実数値)>
R  :    -0.0140
G  :   255.3131
B  :     1.1520

<出力 RGB 信号(整数値)>
R  :    0
G  :  255
B  :    1
u20216187@gw[34]: exit

exit

Script done on Tue Oct 25 15:27:52 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 5-B-1】

［プログラムリスト・コンパイル結果・実行結果］

```
Script started on Tue Oct 25 15:28:04 2022
u20216187@gw[31]: cat 05_B_01.c

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define MAX 100
#define CH 3
#define Ych 0
#define ROW 3
#define COL 3

unsigned char header[54];
unsigned char imgin[3][512][512];//入力画像
unsigned char imgout[3][512][512];//
double dtemp[3][512][512];
int itemp[3][512][512];

void processing();
void get_data();
void put_data();
void rgb_to_ybr();
void ybr_to_rgb();

int width, height;//画像の幅と高さ
int bits;//画像のビット数

double rgb_con_ybr[ROW][COL] = {
  { 0.2990, 0.5870, 0.1140},
   {-0.1687,-0.3313, 0.5000},
   { 0.5000,-0.4187,-0.0813}
};
double ybr_con_rgb[ROW][COL] = {
  { 1.0000, 0.0000, 1.4020},
  { 1.0000,-0.3441,-0.7141},
  { 1.0000, 1.7720, 0.0000}
};
int main() {
    get_data();
    rgb_to_ybr();
    processing();
    ybr_to_rgb();
    put_data();
    return 0;
}

int calculate(int a, int b);//バイト数計算
void get_data() {
    FILE* fp;
```

```
char filename[MAX];
int c;
int filesize, offset, bite_px;//画像の属性


printf("ファイル名を入力して下さい:");
scanf("%s", filename);
fp = fopen(filename, "rb");
if (fp == NULL) {
    printf("%s をオープンできません¥n", filename);
    exit(1);
}
printf("%s をオープンしました.¥n", filename);
for (int i = 0; i < 54; i++) {
    c = fgetc(fp);
    header[i] = c;
}
/* printf("¥n¥n<ファイルタイプ>¥n");
for (int i = 0; i < 2; i++) {
  printf("header[%d]=%02x ", i, header[i]);
  } */
printf("¥n¥n ファイルサイズ>¥n");
filesize = calculate(5, 2);
for (int i = 2; i < 6; i++) {
    printf("header[%d]=%02x ", i, header[i]);
}
printf("¥n¥n%d バイト", filesize);
/* printf("¥n¥n<予約領域>¥n");
for (int i = 6; i < 10; i++) {
  printf("header[%d]=%02x ", i, header[i]);
  } */
printf("¥n¥n<オフセット>¥n");
for (int i = 10; i < 14; i++) {
    printf("header[%d]=%02x ", i, header[i]);

}
offset = calculate(13, 10);
printf("¥n¥n%d バイト", offset);
/*  printf("¥n¥n<情報ヘッダサイズ>¥n");
for (int i = 14; i < 18; i++) {
  printf("header[%d]=%02x ", i, header[i]);
  } */
printf("¥n¥n<画像の幅>¥n");
width = calculate(21, 18);
for (int i = 18; i < 22; i++) {
    printf("header[%d]=%02x ", i, header[i]);

}
printf("¥n%d 画素¥n", width);
printf("¥n¥n<画像の高さ>");
height = calculate(25, 22);
for (int i = 22; i < 26; i++) {
    printf("header[%d]=%02x ", i, header[i]);
}

printf("¥n%d ライン¥n", height);
/*  printf("¥n¥n<色プレーン数>¥n");
```

```
    for (int i = 26; i < 28; i++) {
      printf("header[%d]=%02x ", i, header[i]);
      } */
    printf("¥n¥n<1 画素にあたりのビット数>¥n");
    bite_px = calculate(29, 28);
    for (int i = 28; i < 30; i++) {
        printf("header[%d]=%02x ", i, header[i]);

    }
    printf("¥n%d ビット", bite_px);
    /* printf("¥n¥n<圧縮方式>¥n");
    for (int i = 30; i < 34; i++) {
      printf("header[%d]=%02x ", i, header[i]);
    }
    printf("¥n¥n<画像データサイズ>¥n");
    for (int i = 34; i < 38; i++) {
      printf("header[%d]=%02x ", i, header[i]);
    }
    printf("¥n¥n<水平解像度>¥n");
    for (int i = 39; i < 42; i++) {
      printf("header[%d]=%02x ", i, header[i]);
    }
    printf("¥n¥n<垂直解像度>¥n");
    for (int i = 42; i < 46; i++) {
      printf("header[%d]=%02x ", i, header[i]);
    }
    printf("¥n¥n<色数>¥n");
    for (int i = 46; i < 50; i++) {
      printf("header[%d]=%02x ", i, header[i]);
    }
    printf("¥n¥n<重要な色数>¥n");
    for (int i = 50; i < 54; i++) {
      printf("header[%d]=%02x ", i, header[i]);
      } */
    printf("¥n¥n<挿入ビット数>¥n");
    bits = (offset + width * height * (bite_px / 8)) % 4;
    printf("%d バイト¥n", bits);
    //imgin[][][]の初期化
    for (int h = height - 1; h >= 0; h--) {
        for (int w = 0; w < width; w++) {
            for (int i = 2; i >= 0; i--) {
                imgin[i][w][h] = (unsigned char)fgetc(fp);
            }
        }
    }

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

}
int calculate(int a, int b) {
    int value;
    value = header[a];
    for (int i = a - 1; i >= b; i--) {
        value <<= 8;
        value += header[i];
    }
```

```
    return value;

}

void processing() {

    int x, y;//RGB を表示するため変数
    if (height <= 16 || width <= 16) {
        printf("入力画像データを表示します.");
        printf("¥n<R 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[0][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<G 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[1][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<B 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[2][x][y]);
            }
            printf("¥n");
        }
    }
    else {
        printf("画像サイズが大きいため表示しませんん.¥n");
    }
    for (int h = 0; h < height; h++) {
        for (int w = 0; w < width; w++) {
            for (int i = 0; i < 3; i++) {
                imgout[i][w][h] = imgin[i][w][h];
            }
        }
    }
    printf("出力画像データを作成しました.¥n");
}
void put_data() {
    FILE* fp;
    char cpfilename[MAX];//出力画像名前

    printf("出力ファイル名を入力してください:");
    scanf(" %s", cpfilename);
    fp = fopen(cpfilename, "wp");
    printf("%s をオープンしました.¥n", cpfilename);
    for (int i = 0; i < 54; i++) {
        fputc(header[i], fp);

    }
```

```
    for (int h = height - 1; h >= 0; h--) {
        for (int w = 0; w < width; w++) {
            for (int i = 0; i <= 2; i++) {
                fputc(imgout[i][w][h], fp);
            }
        }
    }
    for (int i = 0; i < bits; i++) {
        fputc('¥0', fp);
    }
    fclose(fp);
    printf("%s をクローズしました.¥n", cpfilename);
}

void rgb_to_ybr() {
    int i, x, y;
    if (height <= 16 || width <= 16) {

        printf("¥n<R 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[0][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<G 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[1][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<B 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[2][x][y]);
            }
            printf("¥n");
        }
    }
    else {
        printf("画像サイズが大きいため表示しません.¥n");
    }

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

            for (i = 0; i < 3; i++) {
                dtemp[i][x][y] = 0.0;
                for (int j = 0; j < 3; j++)
                    dtemp[i][x][y] += rgb_con_ybr[i][j] * (double)imgin[j][x][y];
            }
        }
    }
```

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

            if (dtemp[i][x][y] > 0.0) {
                itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);
            }
            else {
                itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);
            }

            if (i != 0) {
                itemp[i][x][y] += 128;
            }

            if (itemp[i][x][y] > 255) {
                itemp[i][x][y] = 255;
            }
            else if (itemp[i][x][y] < 0) {
                itemp[i][x][y] = 0;
            }

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

printf("¥n<入力 YCbCr 信号(整数値)>¥n");

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

    printf("¥n<Y 信号>¥n");
    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {
            printf("%02x ", imgin[0][x][y]);
        }
        printf("¥n");
    }

    printf("¥n<Cb 信号>¥n");
    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {
            printf("%02x ", imgin[1][x][y]);
        }
        printf("¥n");
    }

    printf("¥n<Cr 信号>¥n");
    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {
            printf("%02x ", imgin[2][x][y]);
        }
        printf("¥n");
    }
}
else {
```

```
        printf("画像サイズが大きいため表示しません.¥n");
    }

}

void ybr_to_rgb() {
    int i, x, y, j;

    printf("¥n<入力 YCbCr 信号(整数値)>¥n");

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

        printf("¥n<Y 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[0][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<Cb 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[1][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<Cr 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgin[2][x][y]);
            }
            printf("¥n");
        }
    }
    else {
        printf("画像サイズが大きいため表示しません.¥n");
    }

    for (y = 0; y < height; y++) {
        for (x = 0; x < width; x++) {
            for (i = 0; i < 3; i++) {
                dtemp[i][x][y] = 0.0;
                for (j = 0; j < 3; j++)
                    if (j == 0)
                        dtemp[i][x][y]       +=       ybr_con_rgb[i][j]       *
(double)imgin[j][x][y];
                    else
                        dtemp[i][x][y]       +=       ybr_con_rgb[i][j]       *
(double)(imgin[j][x][y] - 128);
            }
        }
    }


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

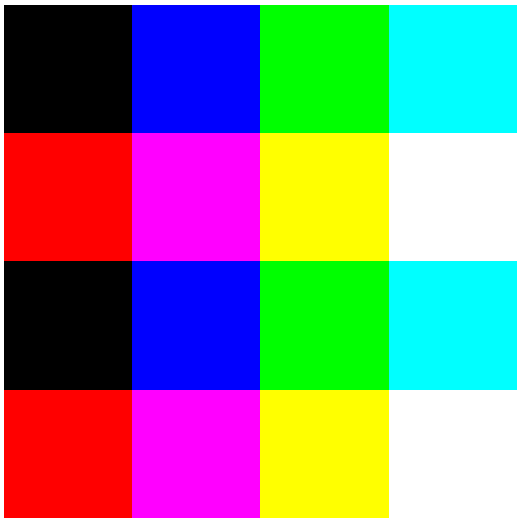
```
        for (x = 0; x < width; x++) {
            for (i = 0; i < 3; i++) {

                if (dtemp[i][x][y] > 0.0) {
                    itemp[i][x][y] = (int)(dtemp[i][x][y] + 0.5);
                }
                else {
                    itemp[i][x][y] = (int)(dtemp[i][x][y] - 0.5);
                }

                if (itemp[i][x][y] > 255) {
                    itemp[i][x][y] = 255;
                }
                else if (itemp[i][x][y] < 0) {
                    itemp[i][x][y] = 0;
                }

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

    printf("¥n<入力 RGB 信号(整数値)>¥n");

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

        printf("¥n<R 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgout[0][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<G 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgout[1][x][y]);
            }
            printf("¥n");
        }

        printf("¥n<B 信号>¥n");
        for (y = 0; y < height; y++) {
            for (x = 0; x < width; x++) {
                printf("%02x ", imgout[2][x][y]);
            }
            printf("¥n");
        }
    }
    else {
        printf("画像サイズが大きいため表示しません.¥n");
    }

}
u20216187@gw[32]: gcc -Wall 05_B_01.c -o 05_B_01
```

```
u20216187@gw[33]: ./05 _B_01

ファイル名を入力して下さい:test05.bmp
test05.bmp をオープンしました.


ファイルサイズ>
header[2]=68 header[3]=00 header[4]=00 header[5]=00

104 バイト

<オフセット>
header[10]=36 header[11]=00 header[12]=00 header[13]=00

54 バイト

<画像の幅>
header[18]=04 header[19]=00 header[20]=00 header[21]=00
4 画素


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


<1 画素にあたりのビット数>
header[28]=18 header[29]=00
24 ビット

<挿入ビット数>
2 バイト

test05.bmp をクローズしました.

<R 信号>
00 00 00 00
ff ff ff ff
00 00 00 00
ff ff ff ff

<G 信号>
00 00 ff ff
00 00 ff ff
00 00 ff ff
00 00 ff ff

<B 信号>
00 ff 00 ff
00 ff 00 ff
00 ff 00 ff
00 ff 00 ff

<入力 YCbCr 信号(整数値)>

<Y 信号>
00 1d 96 b3
4c 69 e2 ff
```

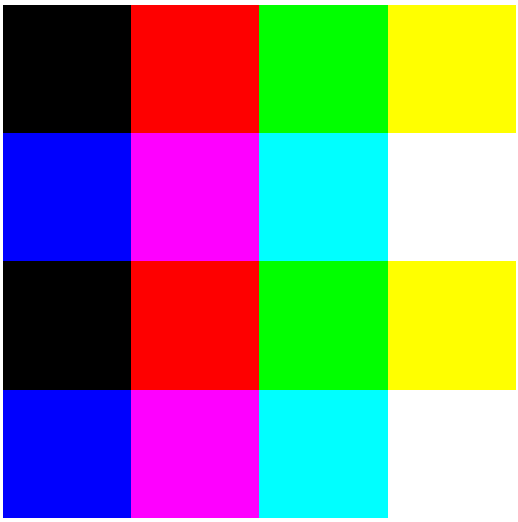
```
00 1d 96 b3
4c 69 e2 ff

<Cb 信号>
80 ff 2c ab
55 d4 00 80
80 ff 2c ab
55 d4 00 80

<Cr 信号>
80 6b 15 00
ff eb 95 80
80 6b 15 00
ff eb 95 80
入力画像データを表示します.
<R 信号>
00 1d 96 b3
4c 69 e2 ff
00 1d 96 b3
4c 69 e2 ff

<G 信号>
80 ff 2c ab
55 d4 00 80
80 ff 2c ab
55 d4 00 80

<B 信号>
80 6b 15 00
ff eb 95 80
80 6b 15 00
ff eb 95 80
出力画像データを作成しました.

<入力 YCbCr 信号(整数値)>

<Y 信号>
00 1d 96 b3
4c 69 e2 ff
00 1d 96 b3
4c 69 e2 ff

<Cb 信号>
80 ff 2c ab
55 d4 00 80
80 ff 2c ab
55 d4 00 80

<Cr 信号>
80 6b 15 00
ff eb 95 80
80 6b 15 00
ff eb 95 80

<入力 RGB 信号(整数値)>

<R 信号>
```

```
00 00 00 00
fe ff ff ff
00 00 00 00
fe ff ff ff

<G 信号>
00 00 ff ff
00 00 ff ff
00 00 ff ff
00 00 ff ff

<B 信号>
00 fe 01 ff
00 fe 00 ff
00 fe 01 ff
00 fe 00 ff
出力ファイル名を入力してください:test05cp.bmp
test05cp.bmp をオープンしました.
test05cp.bmp をクローズしました.
u20216187@gw[34]: exit

exit

Script done on Tue Oct 25 15:29:08 2022
```

［添付図］



Test05.bmp



Test05cp.bmp

【問題アイテムを選択してください。-アイテムを選択してください。】

［プログラムリスト・コンパイル結果・実行結果］

ここをクリックまたはタップしてテキストを入力してください。

［添付図］

ここをクリックまたはタップしてテキストを入力してください。

ここをクリックまたはタップしてテキストを入力してください。

【問題アイテムを選択してください。-アイテムを選択してください。】

［プログラムリスト・コンパイル結果・実行結果］

ここをクリックまたはタップしてテキストを入力してください。

［添付図］

ここをクリックまたはタップしてテキストを入力してください。

ここをクリックまたはタップしてテキストを入力してください。