

[演習 05] 色空間変換

学生番号 : 316098

氏名 : 周いくしん

提出日 : 2021/10/25

[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_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 5-A-1】

[プログラムリスト・コンパイル結果・実行結果]

```
#pragma warning(disable:4996)
#include <stdio.h>
double get_rgb(char input[]);
void TranToYcr(double b[], double c[]);
void TranToRGB(double b[], double c[]);
int main() {
    double rgb[3], ycr[3];
    printf("信号値を入力して下さい(実数値)\n");
    rgb[0] = get_rgb("R:");
    rgb[1] = get_rgb("G:");
    rgb[2] = get_rgb("B:");
    printf("<入力された RGB 信号(実数値)>\n");
    printf("R: %.4f\nG: %.4f\nB: %.4f\n", rgb[0], rgb[1], rgb[2]);
    TranToYcr(rgb, ycr);
    printf("<変換された YCbCr 信号(実数値)>\n");
    printf("Y: %.4f\nCb: %.4f\nCr: %.4f\n", ycr[0], ycr[1], ycr[2]);
    TranToRGB(ycr, rgb);
    printf("<再度変換された RGB 信号(実数値)>\n");
    printf("R: %.4f\nG: %.4f\nB: %.4f\n", rgb[0], rgb[1], rgb[2]);
}

double get_rgb(char input[]) {
    double rgb;
    printf("%s", input);
    scanf("%lf", &rgb);
    return rgb;
}

void TranToYcr(double b[], double c[]) {
    double a[9] = { 0.299, 0.587, 0.114, -0.1687, -0.3313, 0.5, 0.5, -0.4187, -0.0813 };
    c[0] = a[0] * b[0] + a[1] * b[1] + a[2] * b[2];
    c[1] = a[3] * b[0] + a[4] * b[1] + a[5] * b[2];
    c[2] = a[6] * b[0] + a[7] * b[1] + a[8] * b[2];
}

void TranToRGB(double b[], double c[]) {
    double a[9] = { 1, 0, 1.402, 1, -0.3441, -0.7141, 1, 1.772, 0 };
```

```
c[0] = a[0] * b[0] + a[1] * b[1] + a[2] * b[2];  
c[1] = a[3] * b[0] + a[4] * b[1] + a[5] * b[2];  
c[2] = a[6] * b[0] + a[7] * b[1] + a[8] * b[2];  
}
```

信号値を入力して下さい(実数値)

R:0

G:255

B:0

<入力された RGB 信号(実数値)>

R: 0.0000

G: 255.0000

B: 0.0000

<変換された YCbCr 信号(実数値)>

Y: 149.6850

Cb: -84.4815

Cr: -106.7685

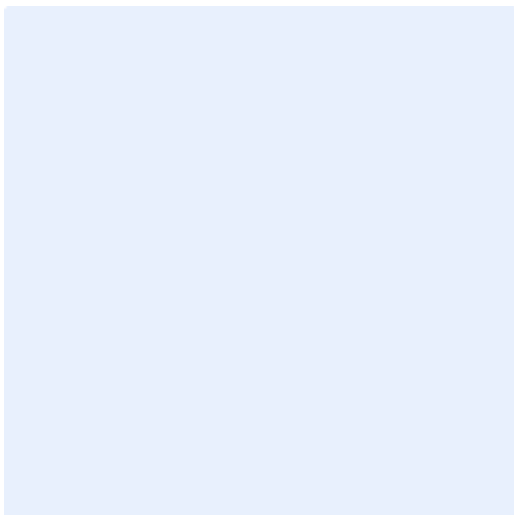
<再度変換された RGB 信号(実数値)>

R: -0.0044

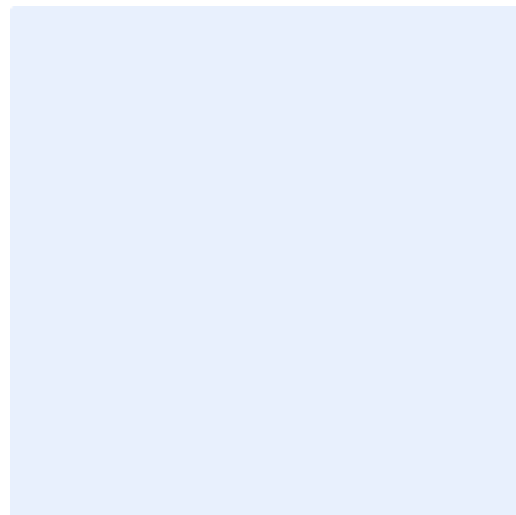
G: 254.9985

B: -0.0162

[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]

```
#pragma warning(disable:4996)
#include <stdio.h>
int get_rgb(char input[]);
void TranToYcr(double b[], double c[]);
void TranToRGB(double b[], double c[]);
int offset = 128;
int Round(double c);
int main() {
    double rgb[3], ycr[3];
    int ycr2[3], rgb2[3];
    printf("信号値を入力して下さい(整数値)\n");
    rgb[0] = get_rgb("R:");
    rgb[1] = get_rgb("G:");
    rgb[2] = get_rgb("B:");
    printf("<入力された RGB 信号(整数値)>\n");
    printf("R: %d\nG: %d\nB: %d\n", (int)rgb[0], (int)rgb[1], (int)rgb[2]);
    //RGBから変換
    TranToYcr(rgb, ycr);
    printf("<変換された YCbCr 信号(実数値)>\n");
    printf("Y: %.4f\nCb: %.4f\nCr: %.4f\n", ycr[0], ycr[1], ycr[2]);
    //四捨五入
    printf("<変換された YCbCr 信号(整数値)>\n");
    for (int i = 0; i < 3; i++) {
        ycr2[i]=Round(ycr[i]);
    }
    printf("Y: %d\nCb: %d\nCr: %d\n", ycr2[0], ycr2[1]+offset, ycr2[2]+offset);
    //Ycrから変換
    TranToRGB(ycr, rgb);
    printf("<再度変換された RGB 信号(実数値)>\n");
    printf("R: %.4f\nG: %.4f\nB: %.4f\n", rgb[0], rgb[1], rgb[2]);
    //四捨五入
    printf("<再度変換された RGB 信号(整数値)>\n");
    for (int i = 0; i < 3; i++) {
        rgb2[i] = Round(rgb[i]);
```

```

    }
    printf("R: %d\nG: %d\nB: %d\n", rgb2[0], rgb2[1], rgb2[2]);
}

int get_rgb(char input[]) {
    int rgb;
    printf("%s", input);
    scanf("%d", &rgb);
    return rgb;
}

void TranToYcr(double b[], double c[]) {
    double a[9] = { 0.299, 0.587, 0.114, -0.1687, -0.3313, 0.5, 0.5, -0.4187, -0.0813 };
    c[0] = a[0] * b[0] + a[1] * b[1] + a[2] * b[2];
    c[1] = a[3] * b[0] + a[4] * b[1] + a[5] * b[2];
    c[2] = a[6] * b[0] + a[7] * b[1] + a[8] * b[2];
}

void TranToRGB(double b[], double c[]) {
    double a[9] = { 1, 0, 1.402, 1, -0.3441, -0.7141, 1, 1.772, 0 };
    c[0] = a[0] * b[0] + a[1] * b[1] + a[2] * b[2];
    c[1] = a[3] * b[0] + a[4] * b[1] + a[5] * b[2];
    c[2] = a[6] * b[0] + a[7] * b[1] + a[8] * b[2];
}

int Round(double c) {
    if (0.01*c > 0)
        c += 0.5;
    if (0.01*c < 0)
        c -= 0.5;
    return c;
}

```

実行結果：

信号値を入力して下さい(整数値)

R:0

G:255

B:0

<入力された 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

<再度変換された RGB 信号(実数値)>

R: -0.0044

G: 254.9985

B: -0.0162

<再度変換された RGB 信号(整数値)>

R: 0

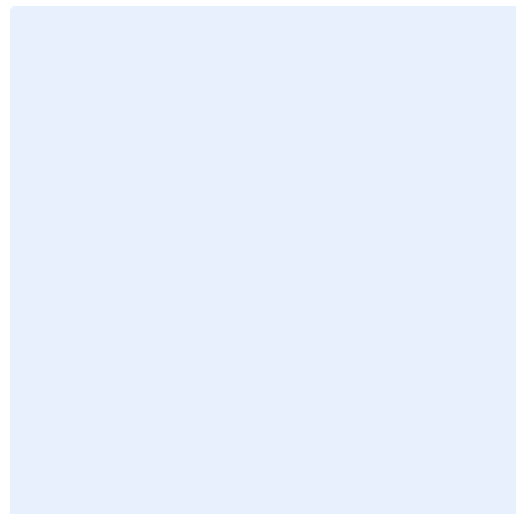
G: 255

B: 0

[添付図]



コピー後の画像
(画像が小さすぎるため、アップロードすると真っ黒にしか表示しません)



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-2】

[プログラムリスト・コンパイル結果・実行結果]

```
#pragma warning(disable:4996)
#include<stdio.h>
void get_Data();
int TransformTo10hex(int a, int b, int c, int d);
void processing();
void put_data();
void rgb_to_ybr();
void ybr_to_rgb();
int offset = 128;
int Round(double c);
unsigned char header[54];
unsigned char data[48];
unsigned char imgin[3][512][512];
unsigned char imgout[3][512][512];
double ybr[3][512][512];
int ybr2[3][512][512];
double rgb[3][512][512];
int rgb2[3][512][512];
int width, height;
int main() {
    get_Data();
    rgb_to_ybr();
    processing();
    ybr_to_rgb();
    put_data();
    return 0;
}
void get_Data() {
    int i = 0, size;
    char fName[15] = "testc.bmp";
    FILE* fp;
    printf("ファイル名を入力してください: ");
    //scanf("%s", fName);
    fp = fopen(fName, "r");
```

```
if (fp == NULL)
{
    printf("ファイルをオープンできません.\n");
    return;
}
printf("ファイルをオープンしました.\n");
for (; i < 100000; i++)
{
    if (i < 54) {
        header[i] = fgetc(fp);
        if (feof(fp))
        {
            break;
        }
    }
    else {
        data[i - 54] = fgetc(fp);
        //printf("%.2x\n", data[i-54]);
        if (feof(fp))
        {
            break;
        }
    }
}

size = TransformTo10hex(header[2], header[3], header[4], header[5]);
for (int j = 0; j < 54; j++) {
    switch (j) {
        case 0:printf("\n<ファイルタイプ>\n");
            break;
        case 2:printf("\n\n<ファイルサイズ>\n");
            break;
        case 6:printf("\n%d byte\n\n<予約領域>\n", size);
            break;
        case 10:printf("\n\n<オフセット>\n");
            break;
        case 14:printf("\n%d byte\n\n<情報ヘッダサイズ>\n",
TransformTo10hex(header[10],
                    header[11], header[12], header[13]));
```



```

        break;
case 18:printf("\n\n<画像の幅>\n");
        break;
case 22:printf("\n%d pixel\n\n<画像の高さ>", TransformTo10hex(header[18],
header[19],
        header[20], header[21]));
        break;
case 26:printf("\n%d line\n\n<色プレーン数>", TransformTo10hex(header[22],
header[23],
        header[24], header[25]));
        break;
case 28:printf("\n\n<1画素当たりのビット数>\n");
        break;
case 30:printf("\n%d bit\n\n<圧縮方式>", TransformTo10hex(header[28],
header[29], 0,
        0));
        break;
case 34:printf("\n\n<画像データサイズ>\n");
        break;
case 38:printf("\n\n<水平解像度>\n");
        break;
case 42:printf("\n\n<垂直解像度>\n");
        break;
case 46:printf("\n\n<色数>\n");
        break;
case 50:printf("\n\n<重要な色数>\n");
}
printf("header[%d]=%.2x ", j, header[j]);
width = TransformTo10hex(header[18], header[19], header[20], header[21]);
height = TransformTo10hex(header[22], header[23], header[24], header[25]);
//dataを代入
int q = 0;
for (int y = 0; y < height; y++) {
    // printf("\n");
    for (int x = 0; x < width; x++) {
        //printf("\n");
        for (int z = 0; z < 3; z++) {
            imgin[z][x][y] = data[q];

```

```

        q++;

//printf("%d,%d,%d : %.2x\n", z, x, y, imgin[z][x][y]);
    }
}

}

}

printf("\n\n<挿入ビット数>\n%dバイト\n\n", (size + 54) % 4);
fclose(fp);
printf("ファイルをクローズしました\n");
printf("<入力信号(RGB)>\n");
for (int k = 2; k >= 0; k--) {
    switch (k) {
        case 2:printf("--- R ---"); break;
        case 1:printf("\n--- G ---"); break;
        case 0:printf("\n--- B ---"); break;
    }
    for (int y = height - 1; y >= 0; y--) {
        printf("\n");
        for (int x = 0; x < width; x++) {
            imgout[k][x][y] = imgin[k][x][y];
            printf(" %.2x ", imgin[k][x][y]);
        }
    }
}

printf("\n");
}

int TransformTo10hex(int a, int b, int c, int d)
{
    return d * 16908288 + c * 65536 + b * 256 + a;
}

void processing() {
    for (int k = 2; k >= 0; k--) {
        for (int y = height - 1; y >= 0; y--) {
            for (int x = 0; x < width; x++) {
                imgout[k][x][y] = imgin[k][x][y];
            }
        }
    }
}

```

```

    }

    }

    printf("\n");
}

void put_data() {
    FILE* fp;
    char Fname[15] = "testc_cp.bmp";
    printf("\n出力ファイル名を入力して下さい:");
    //scanf("%s", Fname);
    fp = fopen(Fname, "w");
    printf("\n\n");
    printf("ファイルをオープンしました.\n");
    for (int i = 0; i < 54; i++) {
        fputc(header[i], fp);
    }
    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            for (int k = 0; k < 3; k++) {
                fputc(imgout[k][x][y], fp);
            }
        }
    }
    fclose(fp);
    printf("ファイルをクローズしました.\n");
}

void rgb_to_ybr() {
    printf("\n<入力信号(YCbCr)>\n");
    double a[3][3] = { 0.299, 0.587, 0.114, -0.1687, -0.3313, 0.5, 0.5, -0.4187, -0.0813 };
    int i = 0;
    for (int y = height - 1; y >= 0; y--) {
        i = 0;
        for (int x = 0; x < width; x++) {
            i = 0;
            for (int k = 2; k >= 0; k--) {
                ybr[k][x][y] = imgin[2][x][y] * a[i][0] + imgin[1][x][y] *
a[i][1] + imgin[0][x][y] * a[i][2];
                i++;
            }
        }
    }
}

```

```

        ybr2[k][x][y]=Round(ybr[k][x][y]);
        if (k == 1 || k == 0)
            ybr2[k][x][y] = ybr2[k][x][y] + offset;
        if (ybr2[k][x][y] > 255)
            ybr2[k][x][y] = 255;
        if (ybr2[k][x][y] < 0)
            ybr2[k][x][y] = 0;
    }
}

for (int k = 2; k >= 0; k--) {
    switch (k) {
        case 2:printf("---- Y ----"); break;
        case 1:printf("\n--- Cb ----"); break;
        case 0:printf("\n--- Cr ----"); break;
    }
    for (int y = height - 1; y >= 0; y--) {
        printf("\n");
        for (int x = 0; x < width; x++) {
            printf(" %.2x ", ybr2[k][x][y]);

        }
    }
}

}

void ybr_to_rgb() {
    printf("\n\n<出力信号(YCbCr)>\n");
    for (int k = 2; k >= 0; k--) {
        switch (k) {
            case 2:printf("---- Y ----"); break;
            case 1:printf("\n--- Cb ----"); break;
            case 0:printf("\n--- Cr ----"); break;
        }
        for (int y = height - 1; y >= 0; y--) {
            printf("\n");

```

```

        for (int x = 0; x < width; x++) {
            printf(" %.2x ", ybr2[k][x][y]);

        }

    }

}

printf("\n\n<出力信号 (RGB)>\n");
double b[3][3] = { 1, 0, 1.402,
                    1, -0.3441, -0.7141,
                    1, 1.772, 0 };

int j = 0;
//for (int y = 0; y < height; y++) {
    //for (int x = 0; x < width; x++) {
        //offsetを減ずる
        //ybr[1][x][y] = ybr[1][x][y] - offset;
        //ybr[0][x][y] = ybr[0][x][y] - offset;

    //}

//}

for (int y = height - 1; y >= 0; y--) {
    j = 0;
    for (int x = 0; x < width; x++) {
        j = 0;

        for (int k = 2; k >= 0; k--) {

            //色空間転換
            rgb[k][x][y] = ybr[2][x][y] * b[j][0] + ybr[1][x][y] *
b[j][1] + ybr[0][x][y] * b[j][2];

            j++;

            //四捨五入
            rgb2[k][x][y] = Round(rgb[k][x][y]);
            //置き換え
            if (rgb2[k][x][y] > 255)
                rgb2[k][x][y] = 255;
            if (rgb2[k][x][y] < 0)

```

```

        rgb2[k][x][y] = 0;

    }

}

}

for (int k = 2; k >= 0; k--) {
    switch (k) {
        case 2:printf("---- R ----"); break;
        case 1:printf("\n--- G ---"); break;
        case 0:printf("\n--- B ---"); break;
    }
    for (int y = height - 1; y >= 0; y--) {
        printf("\n");
        for (int x = 0; x < width; x++) {
            printf(" %.2x ", rgb2[k][x][y]);
            imgout[k][x][y] = rgb2[k][x][y];

        }
    }
}

}

}

int Round(double c) {
    if ( c > 0)
        c += 0.5;

    if ( c < 0)
        c -= 0.5;

    return c;
}

```

実行結果：

ファイル名を入力してください：ファイルをオープンしました.

<ファイルタイプ>

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

<ファイルサイズ>

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

104 byte

<予約領域>

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

<オフセット>

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

54 byte

<情報ヘッダサイズ>

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

<画像の幅>

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

4 pixel

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

4 line

<色プレーン数>header[26]=01 header[27]=00

<1画素当たりのビット数>

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

24 bit

<圧縮方式>header[30]=00 header[31]=00 header[32]=00 header[33]=00

<画像データサイズ>

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

<水平解像度>

header[38]=12 header[39]=0b header[40]=00 header[41]=00

<垂直解像度>

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

<色数>

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

<重要な色数>

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

<挿入ビット数>

2バイト

ファイルをクローズしました

<入力信号(RGB)>

--- R ---

00 00 00 00

ff ff ff ff

00 00 00 00

ff ff ff ff

--- G ---

00 00 ff ff

00 00 ff ff

00 00 ff ff

00 00 ff ff

--- B ---

00 ff 00 ff

00 ff 00 ff

00 ff 00 ff

00 ff 00 ff

<入力信号(YCbCr)>

--- Y ---

00 1d 96 b3

4c 69 e2 ff

00 1d 96 b3

4c 69 e2 ff

--- Cb ---

80 ff 2c ab

55 d4 00 80

80 ff 2c ab

55 d4 00 80

--- Cr ---

80 6b 15 00


```
ff eb 95 80
80 6b 15 00
ff eb 95 80
```

<出力信号(YCbCr)>

--- Y ---

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

--- Cb ---

```
80 ff 2c ab
55 d4 00 80
80 ff 2c ab
55 d4 00 80
```

--- Cr ---

```
80 6b 15 00
ff eb 95 80
80 6b 15 00
ff eb 95 80
```

<出力信号(RGB)>

--- R ---

```
00 00 00 00
ff ff ff ff
00 00 00 00
ff ff ff ff
```

--- G ---

```
00 00 ff ff
00 00 ff ff
00 00 ff ff
00 00 ff ff
```

--- B ---

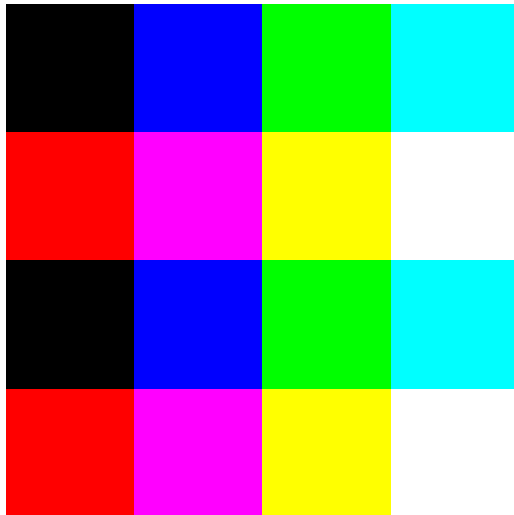
```
00 ff 00 ff
00 ff 00 ff
00 ff 00 ff
00 ff 00 ff
```

出力ファイル名を入力して下さい:

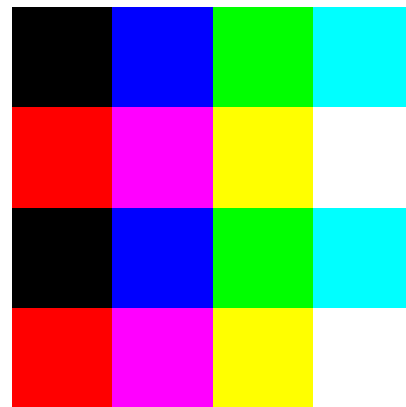
ファイルをオープンしました.

ファイルをクローズしました.

[添付図]



元の画像



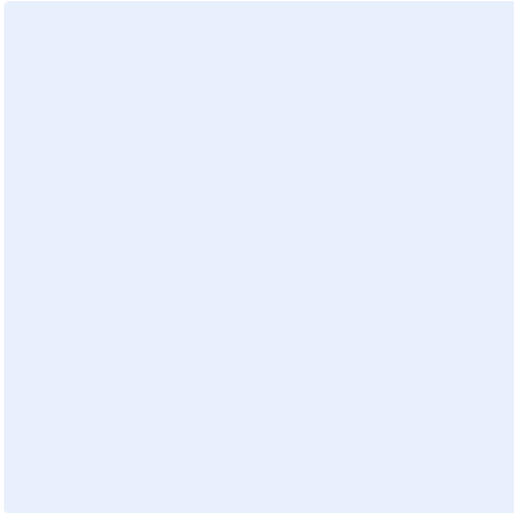
コピー後の画像

【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。】

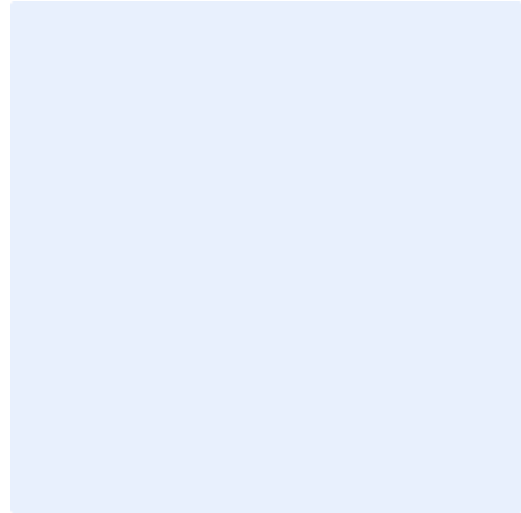
[プログラムリスト・コンパイル結果・実行結果]

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。