

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
void GetData();
int TransformTo10hex(int a,int b,int c,int d);
void Processing();
void PutData();
void rgbToYbr();
void ybrToRgb();
void turn();
void copy();
int roundx(double value);
int width, height,cpw,cph;
unsigned char image[3][512][512], header[54],timage[3][512][512];
double ybr[3][512][512], rgb[3][512][512];
int rybr[3][512][512], rRgb[3][512][512],cpybr[3][512][512],cprgb[3][512][512];
int main() {
    GetData();
    //turn();
    rgbToYbr();
    copy();
    ybrToRgb();
    PutData();
    return 0;
}
void GetData() {
    int c, numOfB, input, i = 0,size;
    char fName[100];
    FILE* fp;

    printf("ファイルめいを入力してください：");
    //scanf("%s", fName);
    //fp = fopen(fName, "r+");
    fp = fopen("pepper.bmp", "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 < 54; i++)
{
    header[i] = fgetc(fp);
    if (feof(fp))
    {
        break;
    }
}
height = TransformTo10hex(header[22], header[23], header[24], header[25]);
width = TransformTo10hex(header[18], header[19], header[20], header[21]);
for ( int y = 0; y < height; y++)
{
    for (int x = 0; x < width; x++)
    {
        for (int z = 0; z < 3; z++)
        {
            image[z][x][y] = fgetc(fp);
            //printf("%d,%d,%d=%.2d\n", z, x, y, image[z][x][y]);
        }
    }
}
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]);
}
printf("\n\n<挿入ビット数>\n%dbite", (size+54)%4);
fclose(fp);
printf("\nファイルをクローズしました.\n");
}

void Processing(){
    printf("\n<入力信号(RGB)>");
    for (int z = 2; z >= 0; z--)
    {
        switch(z) {
            case 2:printf("\nR:\n");
                break;
            case 1:printf("\nG:\n");
                break;
            case 0:printf("\nB:\n");

```

```

    }
    for (int y = height-1 ; y >= 0 ; y--)
    {
        for (int x = 0; x < width; x++)
        {
            printf("%.2x ", image[z][x][y]);

            }
        printf("\n");
    }
}
}

void PutData()
{
    char cpName[100];
    FILE* fp;
    //printf("\n<入力信号(RGB)>");
    //for (int z = 2; z >= 0; z--)
    //{
    //    switch (z) {
    //        case 2:printf("\nR:\n");
    //            break;
    //        case 1:printf("\nG:\n");
    //            break;
    //        case 0:printf("\nB:\n");
    //    }
    //    for (int y = height - 1; y >= 0; y--)
    //    {
    //        for (int x = 0; x < width; x++)
    //        {
    //            printf("%.2x ", rRgb[z][x][y]);

    //        }
    //        printf("\n");
    //    }
    //}
    //printf("\n出力ファイル名を入力して下さい:");
    //scanf("%s", cpName);

```

```

//fp = fopen(cpName, "w+");
fp = fopen("testimgCP.bmp", "w+");
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 (int i=0 ; i < 54; i++)
{
    fputc(header[i],fp);
    if (feof(fp))
    {
        break;
    }
}
int i=0;
for (int y = 0; y < height; y++)
{
    for (int x = 0; x < width; x++)
    {
        for (int z = 0; z < 3; z++)
        {
            fputc(rRgb[z][x][y],fp);
            cprgb[z][x][y]=rRgb[z][x][y];
            //printf("%d,%d,%d=%.2d\n", z, x, y, image[z][x][y]);
        }
    }
}
fclose(fp);
printf("\nファイルをクローズしました.\n");

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

```

```

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] = image[2][x][y] * a[i][0] + image[1][x][y] * a[i][1] +
image[0][x][y] * a[i][2];
            i++;

            rybr[k][x][y] = roundx(ybr[k][x][y]);
            if (k != 2)
                rybr[k][x][y] += 128;
            if (rybr[k][x][y] > 255)
                rybr[k][x][y] = 255;
            if (rybr[k][x][y] < 0)
                rybr[k][x][y] = 0;
        }
    }
}
//printf("\n<入力信号(YCbCr)>");
//for (int z = 2; z >= 0; z--)
//{
//    switch (z) {
//        case 2:printf("\nY:\n");
//            break;
//        case 1:printf("\nCb:\n");
//            break;
//        case 0:printf("\nCr:\n");
//    }
//    for (int y = height - 1; y >= 0; y--)
//    {
//        for (int x = 0; x < width; x++)
//        {
//            printf("%.2x ", rybr[z][x][y]);
//        }
//    }
}

```

```

    //    printf("\n");
    // }
    //}
}
//int round(double value) {
//    int tmp,ans;
//    tmp = (int)(value * 10);
//    if (tmp > 0 && tmp % 10 >= 5)
//        tmp += 10;
//    if (tmp < 0 && tmp % 10 <= -5)
//        tmp -= 10;
//    ans = tmp / 10;
//    if (ans < 0)
//        ans = 0;
//    if (ans > 255)
//        ans = 255;
//    return ans;
//}
int roundx(double value ) {
    if (value > 0)
        value += 0.5;
    if (value < 0)
        value -= 0.5;
    return value;
}
void ybrToRgb() {
    double b[3][3] = { 1,0,1.402,1,-0.3441,-0.7141,1,1.772,0 };
    int j = 0;
    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--) {
                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++;
                rRgb[k][x][y] = roundx(rgb[k][x][y]);
            }
        }
    }
}

```

```

        if (rRgb[k][x][y] > 255)
            rRgb[k][x][y] = 255;
        if (rRgb[k][x][y] < 0)
            rRgb[k][x][y] = 0;
    }
}
}
}
void turn() {
    for (int y = 0; y < height; y++)
    {
        int i = 0;
        for (int x = width-1; x > 0; x--)
        {
            for (int z = 0; z < 3; z++)
            {
                timage[z][i][y] = image[z][x][y];
                //printf("%d,%d,%d=%.2d\n", z, x, y, image[z][x][y]);
            }
            i++;
        }
    }
}
void copy() {
    for (int y = 0; y < height; y++)
    {
        for (int x = 0; x < width; x++)
        {
            for (int z = 0; z < 3; z++)
            {
                if (z!=2)
                {
                    ybr[z][x][y]=0;
                }else{
                    ybr[z][x][y] = rybr[z][x][y];
                    //printf("%d,%d,%d=%.2d\n", z, x, y, image[z][x][y]);
                }
            }
        }
    }
}

```

RGB

0 1 2

}
}
}

