

リスト4.txt

```
/* main_process */
// gRed, gGreen, gBlueの2次元配列にクエリ画像のRGB情報が格納されています
// 同じサイズのgBuff_red, gBuff_green, gBuff_blueに結果画像情報を格納してリターンします
// 平均化グレースケール化とNTSCグレースケールを生成します
int main_process(unsigned char** gRed, unsigned char** gGreen, unsigned char** gBlue,
                 unsigned char** gBuff_red, unsigned char** gBuff_green, unsigned char** gBuff_blue,
                 int width_pixel, int height_pixel) {

    // mode=0ではNTSCグレースケール、mode=1ではRGB平均グレースケールとします
    int mode = 0;

    int i, j;

    if (mode == 0) {
        printf(" NTSCグレースケールを生成\n");

        // NTSCグレースケール画像を生成します
        for (i = 0; i < height_pixel; i++) {
            for (j = 0; j < width_pixel; j++) {
                gBuff_red[i][j] = (gRed[i][j] * 2 + gGreen[i][j] * 4 + gBlue[i][j]) / 7;
                gBuff_green[i][j] = gBuff_red[i][j];
                gBuff_blue[i][j] = gBuff_red[i][j];
            }
        }
    }
    else {
        printf(" RGB平均グレースケールを生成\n");

        // 平均グレースケール画像を生成します
        for (i = 0; i < height_pixel; i++) {
            for (j = 0; j < width_pixel; j++) {
                gBuff_red[i][j] = (gRed[i][j] + gGreen[i][j] + gBlue[i][j]) / 3;
                gBuff_green[i][j] = gBuff_red[i][j];
                gBuff_blue[i][j] = gBuff_red[i][j];
            }
        }
    }

    return 0;
}
```