

## 課題 9-b-1

```
void processing(void) {
    int x, y, z, yellow[256] = {0}, max = 0, maxcollor, normalyellow[256];

    for (z = 0; z < 256; z++) {
        for (x = 0; x < height; x++) {
            for (y = 0; y < width; y++) {
                if (z == imgin[0][x][y]) {
                    yellow[z]++;
                }
            }
        }
    }

    printf("¥n<最大度数>¥n");
    for (z = 0; z < 256; z++) {
        if (max < yellow[z]) {
            max = yellow[z];
            maxcollor = z;
        }
    }

    printf("画素値¥t%d : ¥t%d ¥n", maxcollor, max);
    for (z = 0; z < 256; z++) {
        normalyellow[z] = (int)((double)yellow[z] / (double)max * 100 + 0.5);
    }

    printf("¥n<度数表>¥n");
    for (z = 0; z < 256; z++) {
        printf("画素値¥t%d : ¥t%d --> ¥t%d ¥n", z, yellow[z], normalyellow[z]);
    }

    // ヒストグラム付き画像を出力するよ
    for (x = 0; x < height; x++) {
        for (y = 0; y < width; y++) {
            imgout[0][x][y] = imgin[0][x][y];
            imgout[1][x][y] = imgin[1][x][y];
            imgout[2][x][y] = imgin[2][x][y];
        }
    }
}
```

```

if (y < 256 && x > height - 100) {
    imgout[0][x][y] = 0;
    imgout[1][x][y] = 128;
    imgout[2][x][y] = 128;
    if (normalyellow[y] >= height - x) {
        imgout[0][x][y] = 0;
        imgout[1][x][y] = 128;
        imgout[2][x][y] = 255;
    }
}
}
}
}

printf("%n 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");
}

```



元画像



変換画像

## 課題 9-b-2

```
void processing(void) {
    int x, y, z, max = 0, min = 256;

    for (x = 0; x < height; x++) {
        for (y = 0; y < width; y++) {
            imgout[0][x][y] = imgin[0][x][y];
            imgout[1][x][y] = imgin[1][x][y];
            imgout[2][x][y] = imgin[2][x][y];
            if (max < imgout[0][x][y]) {
                max = imgout[0][x][y];
            } else if (min > imgout[0][x][y]) {
                min = imgout[0][x][y];
            }
        }
    }

    for (x = 0; x < height; x++) {
        for (y = 0; y < width; y++) {
            imgout[0][x][y] = imgin[0][x][y];
            if (imgout[0][x][y] < min && 0 <= imgout[0][x][y]) {
                imgout[0][x][y] = 0;
            } else if (max < imgout[0][x][y]) {
                imgout[0][x][y] = 255;
            } else {
                imgout[0][x][y] = (255 / (max - min)) * (imgout[0][x][y] - min);
            }
        }
    }

    printf("%n 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");
}
```



元画像



変換画像