

```

// area_info[][0]=-1を無効エリアとして、有効なエリアのみを残してソートします
ABHB_sort_effective_area_info(&area_info_counter, area_info);

//*****これ以降が5_2変更追加分部*****

// 各エリア内の外周が円形であることを確認します
// エリアを囲んだ矩形の縦横の中心を円の中心座標、縦横長さの平均値を直径とします
// 円周上に青ピクセルが存在して、円周の外側に青ピクセルが無い場合、円形の物と判断しま
す。
int outer_pixel = 10; // 円の外側とする半径の差
int margin = 5; // 円周上の青ピクセル数をカウントする際の半径マージン
for (n = 0; n < area_info_counter; n++) {
    int area_w = area_info[n][7] - area_info[n][6] + 1; //①
    int area_h = area_info[n][5] - area_info[n][4] + 1; //②
    int diameter = (area_w + area_h) / 2; //③
    int radius = diameter / 2;
    int center_i = (area_info[n][4] + area_info[n][5]) / 2; //④
    int center_j = (area_info[n][6] + area_info[n][7]) / 2; //④
    int in_blue_counter = 0;
    int in_counter_all = 0;
    int out_blue_counter = 0;
    int out_counter_all = 0;
    for (i = center_i - radius - outer_pixel; i <= center_i + radius + outer_pixel;
i++) {
        for (j = center_j - radius - outer_pixel; j <= center_j + radius + outer_pixel;
j++) {
            if (0 <= i && i < height_pixel && 0 <= j && j < width_pixel) {
                int circle_value = (i - center_i) * (i - center_i) + (j - center_j) * (j -
center_j);
                if ((radius - margin) * (radius - margin) <= circle_value && circle_value <=
(radius * radius)) {
                    if (gBuff_blue[i][j] == 255) in_blue_counter++;
                    in_counter_all++;
                }
                if ((radius + outer_pixel - margin) * (radius + outer_pixel - margin) <=
circle_value &&
circle_value <= (radius + outer_pixel) * (radius + outer_pixel)) {
                    if (gBuff_blue[i][j] == 255) out_blue_counter++;
                    out_counter_all++;
                }
            }
        }
    }
    if (0 < in_counter_all && 0 < out_counter_all) {
        in_blue_counter = in_blue_counter * 100 / in_counter_all;
        out_blue_counter = out_blue_counter * 100 / out_counter_all;
    }

    // in_blue_counterが60%以上、out_blue_counterが10%以下の場合円形とします⑥
    char circle_message[64];
    if (60 <= in_blue_counter && out_blue_counter <= 10) {
        sprintf(circle_message, "円形です!");
    }
    else {
        sprintf(circle_message, "円形ではありません!");
    }

    printf(" %2d center=(%4d,%4d) area_w=%3d area_h=%3d diameter=%3d
in_blue_counter=%2d out_blue_counter=%2d %s¥n",
n, center_j, center_i, area_w, area_h, diameter, in_blue_counter,
out_blue_counter, circle_message);
}

```

```

// 直径の値をarea_info[][0]に、中心座標をarea_info[][2, 3]に記録します
area_info[n][0] = diameter;
area_info[n][2] = center_i;
area_info[n][3] = center_j;
}

// 直径から何円のコインかを特定して全コインの合計額を求めます⑦
int result_yen_all = 0;
int diameter_th[5] = { 219, 229, 237, 246, 264 }; // 1円玉と50円玉の直径中間、50と5
円の間・・・10と500円の間での配列
int yen[6] = { 1, 50, 5, 100, 10, 500 }; // 金額の配列
for (n = 0; n < area_info_counter; n++) {
    int k;
    int result_k = 5;
    for (k = 0; k < 5; k++) {
        if (area_info[n][0] < diameter_th[k]) {
            result_k = k;
            k = 5;
        }
    }
    result_yen_all = result_yen_all + yen[result_k];
    printf(" %2d center=(%4d,%4d) %3d円玉 合計金額:%4d円¥n",
        n, area_info[n][3], area_info[n][2], yen[result_k], result_yen_all);
}

free(area_info_p);
free(area_info);
free(gGray_p);
free(gGray);

return 0;
}

```