

```

/*****/
/* 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

#define KEYSTONE_RATIO 0.145 //
#define PINCUSHION_RATIO 0.013
#define PAI 3.1415

// 台形歪を補正します
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) {

    int i, j, ii, jj;

    // 台形画像を補正します①
    for (i = 0; i < height_pixel; i++) {
        ii = height_pixel - int((double(height_pixel) - i) * (1.0 - KEYSTONE_RATIO *
            ((double(height_pixel) - i) / height_pixel)));
        for (j = width_pixel / 2; 0 <= j; j--) {
            jj = width_pixel / 2 - int((width_pixel / 2.0 - j) * (1.0 - KEYSTONE_RATIO *
                ((double(height_pixel) - i) / height_pixel)));
            if (0 <= ii && ii < height_pixel && 0 <= jj && jj < width_pixel) {
                gBuff_red[i][j] = gRed[ii][jj];
                gBuff_green[i][j] = gGreen[ii][jj];
                gBuff_blue[i][j] = gBlue[ii][jj];
            }
        }
    }
    for (j = width_pixel / 2; j < width_pixel; j++) {
        jj = width_pixel / 2 + int((j - width_pixel / 2.0) * (1.0 - KEYSTONE_RATIO *
            ((double(height_pixel) - i) / height_pixel)));
        if (0 <= ii && ii < height_pixel && 0 <= jj && jj < width_pixel) {
            gBuff_red[i][j] = gRed[ii][jj];
            gBuff_green[i][j] = gGreen[ii][jj];
            gBuff_blue[i][j] = gBlue[ii][jj];
        }
    }
}

// カレントディレクトリに台形補正後の画像をダンプします
char dump_file_name[256];
sprintf(dump_file_name, "keyston.bmp");
ABHB_bmp_read_write(1, dump_file_name, gBuff_red, gBuff_green, gBuff_blue,
&width_pixel, &height_pixel);

// 残った糸巻き歪を補正してgBuff_green[][]に結果を格納します
for (i = 0; i < height_pixel; i++) {
    double comp_ratio = cos((fabs(height_pixel / 2.0 - i) / (height_pixel / 2.0)) * PAI
/ 2.0); //②
    for (j = width_pixel / 2; 0 <= j; j--) {
        gBuff_green[i][j] = 0;
        jj = width_pixel / 2 - int((width_pixel / 2.0 - j) * (1.0 - PINCUSHION_RATIO *
comp_ratio));
        if (0 <= jj && jj < width_pixel) {
            gBuff_green[i][j] = gBuff_red[i][jj];
        }
    }
}

```

```

    }
  }
  for (j = width_pixel / 2; j < width_pixel; j++) {
    gBuff_green[i][j] = 0;
    jj = width_pixel / 2 + int((j - width_pixel / 2.0) * (1.0 - PINCUSHION_RATIO *
comp_ratio));
    if (0 <= jj && jj < width_pixel) {
      gBuff_green[i][j] = gBuff_red[i][jj];
    }
  }
}

// gBuff_green[][]をgBuff_red[][]とgBuff_blue[][]にコピーします
for (i = 0; i < height_pixel; i++) {
  for (j = 0; j < width_pixel; j++) {
    gBuff_red[i][j] = gBuff_green[i][j];
    gBuff_blue[i][j] = gBuff_green[i][j];
  }
}

return 0;
}

```