```
// FILE: whiteBox1_0.c
// DATE: 02/27/20 11:42
// AUTH: P.Oh
// DESC: Draw white box on gray RAW 256x256 image

#include<stdlib.h>
#include<stdio.h>
#include<memory.h>


struct Image {
    int Rows;             // image's number of rows
    int Cols;             // image's number of columns
    unsigned char *Data;  // pointer to image data
}; // end of struct Image

void Img_in(struct Image *Img) {
    FILE *ifile;
    int i;

    // NB: Assumes already know that using a RAW image file 256 x 256 size
    // Open file for binary reading
    ifile = fopen("cameraMan.raw", "rb"); // rb: reading text or binary files

    // Read directly into the image array
    for(i=0; i < Img->Rows; ++i)
        fread(Img->Data + i*Img->Cols, Img->Cols, 1, ifile);

    fclose(ifile);

} // end Img_in

void Img_DrawBox(struct Image *In, struct Image *Out) {

 long i, j;
 int val;
 unsigned char *tmp;

 // original example: 29, 11, 10, 25
 int boxColWidth = 75; // 29; // [pix] hence 14 pixels to the left and right of
center
 int boxRowHeight = 50; // 11; // [pix] hence 5 pixels upwards and downwards from
center
 int boxRowCenter = 128; // 10;
 int boxColCenter = 128; // 25;
 int boxTopLeftRowCorner, boxTopLeftColCorner;
 int boxTopRightRowCorner, boxTopRightColCorner;
 int boxBottomLeftRowCorner, boxBottomLeftColCorner;
 int boxBottomRightRowCorner, boxBottomRightColCorner;

 boxTopLeftRowCorner = boxRowCenter - ((boxRowHeight-1)/2); // 10 - (11-1)/2 = 5
NB: minus one because don't count center pixel
 boxTopLeftColCorner = boxColCenter - ((boxColWidth-1)/2);
 printf("Box Top Left corner is (%d, %d)\n", boxTopLeftRowCorner,
boxTopLeftColCorner);

 boxTopRightRowCorner = boxRowCenter - ((boxRowHeight-1)/2);
 boxTopRightColCorner = boxColCenter + ((boxColWidth-1)/2);
 printf("Box Top Right corner is (%d, %d)\n", boxTopRightRowCorner,
boxTopRightColCorner);
 boxBottomLeftRowCorner = boxRowCenter + ((boxRowHeight-1)/2);
 boxBottomLeftColCorner = boxColCenter - ((boxColWidth-1)/2);
 printf("Box Bottom Left corner is (%d, %d)\n", boxBottomLeftRowCorner,
```

```
boxBottomLeftColCorner);
 boxBottomRightRowCorner = boxRowCenter + ((boxRowHeight-1)/2);
 boxBottomRightColCorner = boxColCenter + ((boxColWidth-1)/2);
 printf("Box Bottom Right corner is (%d, %d)\n", boxBottomRightRowCorner,
boxBottomRightColCorner);

 for(i=0; i<In->Rows; ++i) {
    for(j=0; j<In->Cols; ++j) {
       val = *(In->Data + i*In->Rows + j);
       if( (i==boxTopLeftRowCorner || i==boxBottomLeftRowCorner) ) {
          // OK, we're on box's top or bottom row
          if( (j>=boxTopLeftColCorner && j<=boxTopRightColCorner) ||
(j>=boxBottomLeftColCorner && j<=boxBottomRightColCorner) ){
             // Draw top OR bottom line
             val = 255; // make row white between left and right side
          }; // otherwise just keep the original value of val
       }; // end if that checks for box's top or bottom row

       if( j==boxTopLeftColCorner || j==boxTopRightColCorner ) {
         // OK, we're on left or right side
         if( (i>=boxTopLeftRowCorner && i<=boxBottomLeftRowCorner) ||
(i>=boxTopRightRowCorner && i<=boxBottomRightRowCorner) ) {
           // Draw left OR right line
           val = 255; // make column white between top and bottom row
         };
       }; // end if that checks for box's left or right side
       tmp = Out->Data + i*Out->Rows + j;
       *tmp = (unsigned char)val;
    };
 };
} // end Img_threshold

void Img_out(struct Image *Out) {
    FILE *ofile;
    int i;

    // open (or create) a file for writing
    // ofile = fopen("OUTPUT.RAW", "wb");
    ofile = fopen("cameraManWithWhiteBox.raw", "wb");
    // out the image by rows
    for(i=0; i < Out->Rows; ++i)
        fwrite(Out->Data + i*Out->Cols, Out->Cols, 1, ofile);

    fclose(ofile);
} // end Img_out

int main() {

  FILE *ofile;
  struct Image In, Out; // Declare input and output images

  // Initialize image parameters and allocate memory
  In.Rows = Out.Rows = 256;
  In.Cols = Out.Cols = 256;
  In.Data = (unsigned char *)calloc(In.Rows, In.Cols);
  Out.Data = (unsigned char *)calloc(Out.Rows, Out.Cols);
  ofile = fopen("cameraManWithWhiteBox.raw", "wb");

  Img_in(&In);
  Img_DrawBox(&In, &Out);
  Img_out(&Out);

} // end of main
```