```
// FILE: threshold1_0a.c - Works!
// DATE: 02/21/20 08:34
// AUTH: P.Oh
// DESC: Output is threshold of Input image
// REFS: negative2_0.c

#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 RAW image file 256 x 256 size
    // Open file for binary reading
    // Assumes RAW file in same directory as this C-program
    ifile = fopen("cameraMan.raw", "rb"); // read binary file

    // 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_out(struct Image *Out) {
    FILE *ofile;
    int i;

    // Open (or create) binary file for writing
    ofile = fopen("thresholdOutput.raw", "wb");
    // Output 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

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

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

 thresholdValue = 50;

 for(i=0; i<In->Rows; ++i) {
    for(j=0; j<In->Cols; ++j) {
        val = *(In->Data + i*In->Rows + j);
        if(val < thresholdValue) {
            val = 0;
        }
        else {
            val = 255;
        }
```

```
        tmp = Out->Data + i*Out->Rows + j;
        *tmp = (unsigned char)val;
    };
 };
} // end Img_threshold

int main() {

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

  // Assumes RAW image is 256-by-256 bytes
  In.Rows = Out.Rows = 256;
  In.Cols = Out.Cols = 256;
  // Allocate memory for images
  In.Data = (unsigned char *)calloc(In.Rows, In.Cols);
  Out.Data = (unsigned char *)calloc(Out.Rows, Out.Cols);

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

} // end of main
```