// Gray to binary

unsigned char Gray;

//Perform conversions

Gray = fgetc(infptr);

do{

if (Gray > 127)

fputc(255,outfptr);

else

fputc(0,outfptr);

//Read next pixel to check if we have an End of File

Gray = fgetc(infptr);

} while (!feof(infptr));

// Negative

unsigned char Gray;

//Perform conversions

Gray = fgetc(infptr);

do{

fputc(255-Gray,outfptr);

//Read next pixel to check if we have an End of File

Gray = fgetc(infptr);

} while (!feof(infptr));

// Change output file to outfptrh

// Histogram

unsigned int histcount[256],Gray;

char string[10];

//Initialize histogram count to 0

for (int i=0;i<256;i++)

histcount[i] = 0;

//Perform conversions

Gray = fgetc(infptr);

do{

//Increment histogram for grayscale level

histcount[Gray]++;

//Read next pixel to check if we have an End of File

Gray = fgetc(infptr);

} while (!feof(infptr));

//Convert the histogram values from integers to string

for (int i=0;i<256;i++){

itoa (histcount[i],string,10);

fputs(string,outfptrh);

fputs("\n",outfptrh);

}

// RGB to Gray

unsigned char R,G,B,Gray;

char string[10];

//Perform conversions

R = fgetc(infptr);

do{

G = fgetc(infptr);

B = fgetc(infptr);

Gray = (R + G + B) / 3;

fputc(Gray,outfptr);

//Read next pixel to check if we have an End of File

R = fgetc(infptr);

} while (!feof(infptr));

// Mirror

unsigned char Gray;

unsigned char mat[512][512];

int ren, col;

//Perform conversions

for (ren = 0; ren < 512; ren++)

for (col = 0; col < 512; col++) {

Gray = fgetc(infptr);

mat[ren][col] = Gray;

}

for (ren = 0; ren < 512; ren++)

for (col = 0; col < 512; col++) {

//Invert columns

Gray = mat[ren][511-col];

fputc(Gray,outfptr);

}

// Reduction 50%

unsigned char Gray;

unsigned char mat[512][512];

int ren, col;

//Add \*.pgm Header to output file

fprintf(outfptr,"P5\n256 256\n255\n");

//Perform conversions

for (ren = 0; ren < 512; ren++)

for (col = 0; col < 512; col++) {

Gray = fgetc(infptr);

mat[ren][col] = Gray;

}

for (ren = 0; ren < 256; ren++)

for (col = 0; col < 256; col++) {

//Invert columns

Gray = mat[ren\*2][col\*2];

fputc(Gray,outfptr);

}