WEEK 4

#include "helpers.h"

#include<math.h>

// Convert image to grayscale

void grayscale(int height, int width, RGBTRIPLE image[height][width])

{

for(int row =0; row<height; row++){

for (int col=0; col<width; col++){

int avg= round((image[row][col].rgbtRed+image[row][col].rgbtGreen+image[row][col].rgbtBlue)/3.0);

image[row][col].rgbtRed=image[row][col].rgbtGreen=image[row][col].rgbtBlue=avg;

}

}

return;

}

// Convert image to sepia

void sepia(int height, int width, RGBTRIPLE image[height][width])

{

int sRed, sGreen,sBlue;

for(int row =0; row<height; row++){

for (int col=0; col<width; col++){

sRed= round(0.393\*image[row][col].rgbtRed +0.769\*image[row][col].rgbtGreen+0.189\*image[row][col].rgbtBlue);

sGreen= round(0.349\*image[row][col].rgbtRed +0.686\*image[row][col].rgbtGreen+0.168\*image[row][col].rgbtBlue);

sBlue= round(0.272\*image[row][col].rgbtRed +0.534\*image[row][col].rgbtGreen+0.131\*image[row][col].rgbtBlue);

image[row][col].rgbtRed= sRed >255? 255:sRed;

image[row][col].rgbtGreen= sGreen >255? 255:sGreen;

image[row][col].rgbtBlue= sBlue >255? 255:sBlue;

}

}

return;

}

// Reflect image horizontally

void reflect(int height, int width, RGBTRIPLE image[height][width])

{

RGBTRIPLE original[height][width];

for(int row =0; row<height; row++)

{

for (int col=0; col<width; col++)

{

original[row][col] =image[row][col];

}

}

for(int row =0; row<height; row++)

{

for (int col=0, swap= width-1; col<width; col++)

{

image[row][col] =original[row][swap];

swap --;

}

}

return;

}

// Blur image

void blur(int height, int width, RGBTRIPLE image[height][width])

{

RGBTRIPLE original[height][width];

for(int row =0; row<height; row++)

{

for (int col=0; col<width; col++)

{

original[row][col] =image[row][col];

}

}

int totar, totalg, totalb,arrow,arrcol,count;

int totalr=totalg=totalb=0;

for(int row =0; row<height; row++)

{

for (int col=0, swap= width-1; col<width; col++)

{

for (arrow =row-1; arrow<= row+1;arrow++){

for(arrcol = col-1; arrcol<=col +1; arrcol++){

if(arrow<width && arrow < height && arrcol >=0 && arrow >=0)

{

totalr += original[arrow][arrcol].rgbtRed;

totalg += original[arrow][arrcol].rgbtGreen;

totalb += original[arrow][arrcol].rgbtBlue;

count ++;

}

}

}

image[row][col].rgbtRed = round(totalr /count);

image[row][col].rgbtGreen=round(totalg/count);

image[row][col].rgbtBlue=round(totalb/count);

count=0;

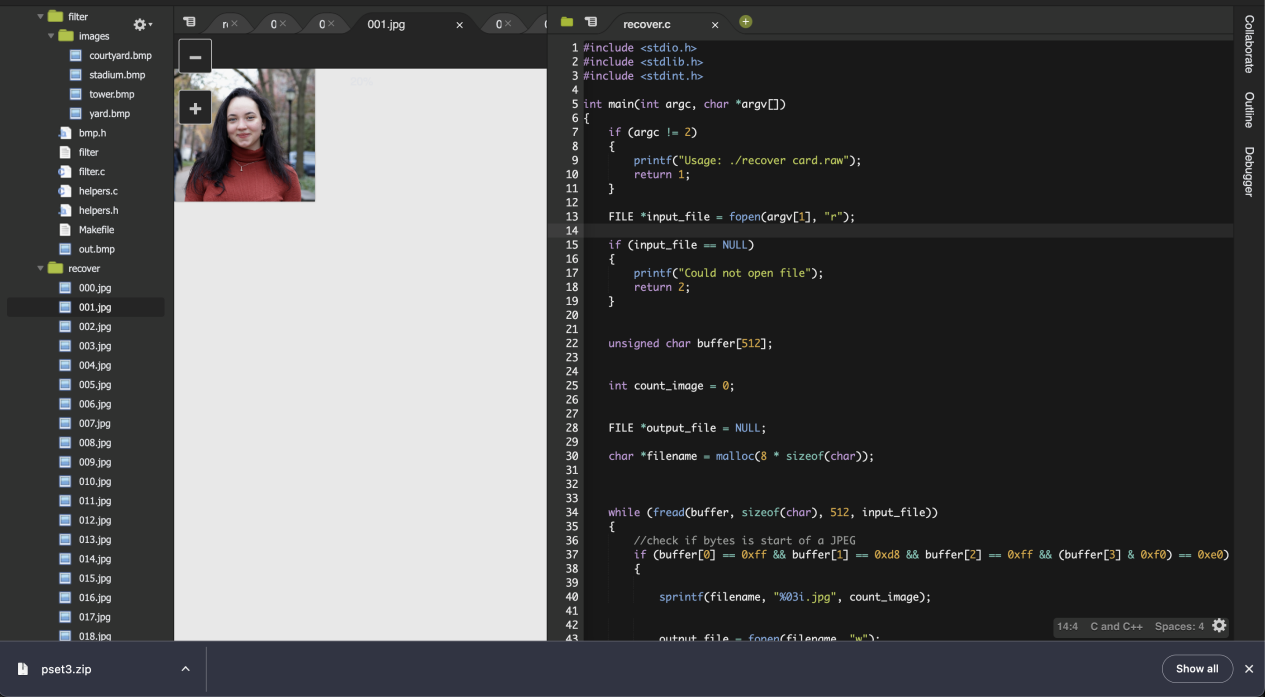
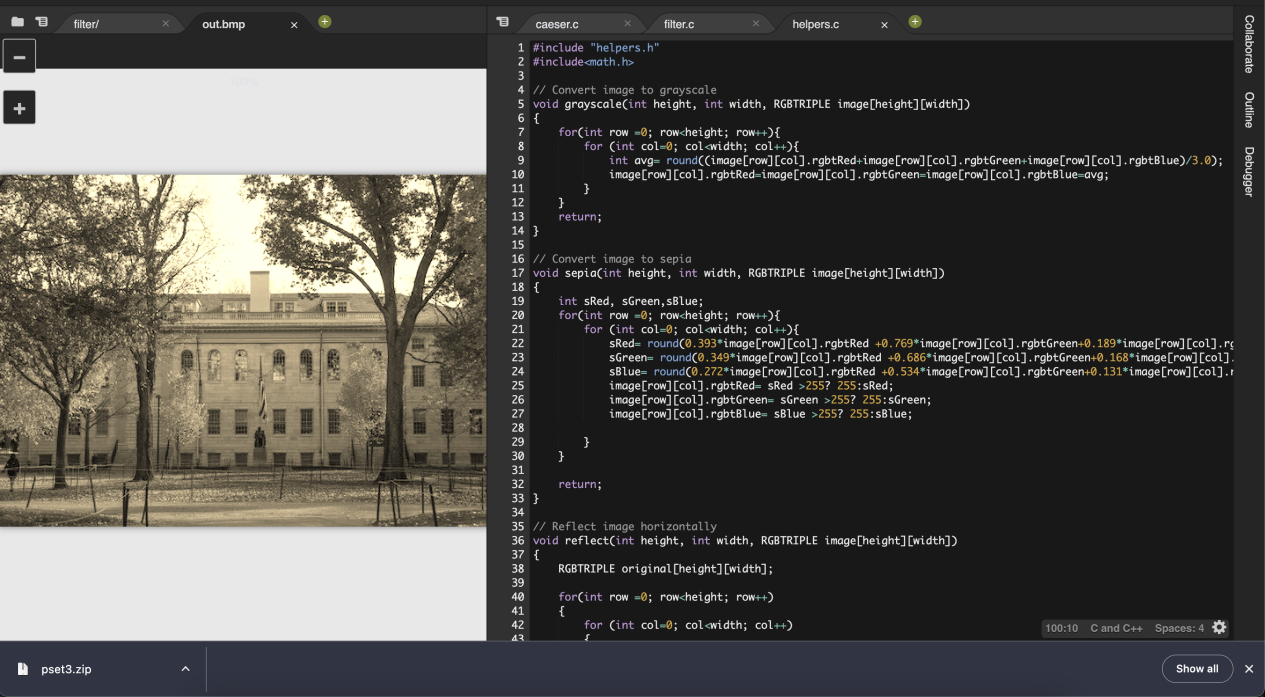
totalr=totalg=totalb=0;

}

}

return;

}



#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

int main(int argc, char \*argv[])

{

if (argc != 2)

{

printf("Usage: ./recover card.raw");

return 1;

}

FILE \*input\_file = fopen(argv[1], "r");

"

if (input\_file == NULL)

{

printf("Could not open file");

return 2;

}

unsigned char buffer[512];

int count\_image = 0;

FILE \*output\_file = NULL;

char \*filename = malloc(8 \* sizeof(char));

while (fread(buffer, sizeof(char), 512, input\_file))

{

if (buffer[0] == 0xff && buffer[1] == 0xd8 && buffer[2] == 0xff && (buffer[3] & 0xf0) == 0xe0)

{

sprintf(filename, "%03i.jpg", count\_image);

output\_file = fopen(filename, "w");

count\_image++;

}

if (output\_file != NULL)

{

fwrite(buffer, sizeof(char), 512, output\_file);

}

}

free(filename);

fclose(output\_file);

fclose(input\_file);

return 0;

}

