**1-)**A picture containing text

Description automatically generated

Text, letter

Description automatically generated

**2-)**

// gcc hw1.c -o hw1 img\_pro.c -lm

// ./ hw1 cathedral.pgm

#include "img\_pro.h"

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

{

unsigned char \*\*img , \*\*img2 , \*\*img3;

char \*pgm\_file;

int i , j , NC , NR , count = 0;

if(argc!=2)

{

printf("\nUsage: hw1 [Image file (\*.pgm)]\n");

printf("\nE.g. hw1 cathedral.pgm \n");

exit(-1);

}

pgm\_file = argv[1]; // 2rd input for which image should be corrected

img = pgm\_file\_to\_img(pgm\_file , &NC , &NR); // to save uncorrected pixel values

show\_pgm\_file(pgm\_file); // shows uncorrected image

img2 = alloc\_img(NC , NR); // to allocate space for corrected pixel values (same size with img)

img3 = alloc\_img(NC , NR); // to allocate space for corrected pixel values (same size with img)

//--------------PART - 1--------------

for(i = 0 ; i < NR ; i++) // for threshold "128" processed image (It can be maximum of 255)(BW)

{

for(j = 0 ; j<NC ; j++)

{

if(127 > img[i][j])

{

img2[i][j] = 0;

}

else if(255 > img[i][j] && 127 <= img[i][j])

{

img2[i][j] = 255;

}

}

}

img\_to\_pgm\_file(img2,"hw1.pgm",NC,NR); // Converting image(matrix) to Grayscale(actual image)

show\_pgm\_file("hw1.pgm"); // Displays BW image

//--------------PART - REST--------------

int l = 1; // label counter

long int label[NR][NC];

// CCL check for a , b , c , d , p

for(i = 1 ; i < NR - 1 ; i++)

{

for(j = 1 ; j < NC - 1 ; j++)

{

if(img2[i][j] == 255) // the case of, "p" == 1 on CCL

{

if(img2[i-1][j-1] == 255) // checking CCL for "a"

{

label[i][j] = l;

}

else if(img2[i-1][j] == 255) // checking CCL for "b"

{

label[i][j] = l;

}

else if(img2[i-1][j+1] == 255) // checking CCL for "c"

{

label[i][j] = l;

}

else if(img2[i][j-1] == 255) // checking CCL for "d"

{

label[i][j] = l;

}

else // if there is no neighbor, then increment the label counter

{

l++;

}

}

}

}

// 8 neighbor connection check

/\*

for(i = 1 ; i < NR - 1 ; i++)

{

for(j = 1 ; j < NC - 1 ; j++)

{

if(img2[i][j] == 255) // the case of, "p" == 1 on CCL

{

if(img2[i-1][j-1] == 255)

{

label[i][j] = l;

}

else if(img2[i-1][j] == 255)

{

label[i][j] = l;

}

else if(img2[i-1][j+1] == 255)

{

label[i][j] = l;

}

else if(img2[i][j-1] == 255)

{

label[i][j] = l;

}

else if(img2[i][j+1] == 255)

{

label[i][j] = l;

}

else if(img2[i+1][j-1] == 255)

{

label[i][j] = l;

}

else if(img2[i+1][j] == 255)

{

label[i][j] = l;

}

else if(img2[i+1][j+1] == 255)

{

label[i][j] = l;

}

else // if there is no neighbor, then increment the label counter

{

l++;

}

}

}

}

\*/

int counter = 0;

counter = 255 / l;

int incrementation = counter;

int val = counter;

for(i = 0 ; i < NR ; i++)

{

for(j = 0 ; j<NC ; j++)

{

img3[i][j] = label[i][j]; // assigning the labels on the image

}

}

img\_to\_pgm\_file(img3,"hw2.pgm",NC,NR); // Converting image(matrix) to Grayscale(actual image)

show\_pgm\_file("hw2.pgm"); // Displays the labels

printf("Total number of labels: %d\n\n", l);

free\_img(img);

free\_img(img2);

return(1);

}