#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

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

if (argc != 2) {

printf("Usage: %s image\_file\_name\n", argv[0]);

return EXIT\_FAILURE;

}

//CV\_LOAD\_IMAGE\_COLOR = 1 forces the resultant IplImage to be colour.

//CV\_LOAD\_IMAGE\_GRAYSCALE = 0 forces a greyscale IplImage.

//CV\_LOAD\_IMAGE\_UNCHANGED = -1

IplImage\* Img1 = cvLoadImage(argv[1], CV\_LOAD\_IMAGE\_COLOR);

// Always check if the program can find a file

if (!Img1) {

printf("Error: fichero %s no leido\n", argv[1]);

return EXIT\_FAILURE;

}

// a visualization window is created with title 'image'

cvNamedWindow("color", CV\_WINDOW\_NORMAL);

// img is shown in 'image' window

cvShowImage("color", Img1);

cvWaitKey(0);

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

for (int fila = 0; fila < Img1->height; fila++) {

uchar \*pImg1 = Img1->imageData + (fila \* Img1->widthStep);

for (int columna = 0; columna < Img1->width; columna++) {

if(\*pImg1>0)\*pImg1 = \*pImg1-1; //B

pImg1++;

if(\*pImg1>0)\*pImg1 = \*pImg1-1; //G

pImg1++;

if(\*pImg1>0)\*pImg1 = \*pImg1-1; //R

pImg1++;

}

}

cvShowImage("color", Img1);

cvWaitKey(1);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}