**3. Mostrar una nueva imagen cuya parte superior sea la parte inferior de fruits, y cuya parte inferior sea la parte superior. El resultado debe ser el que indica la figura 3.**

#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;

}

IplImage\* ImgMal =cvCreateImage(cvSize(Img1->width,Img1->height),Img1->depth,Img1->nChannels);

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

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

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

// img is shown in 'image' window

cvShowImage("color", Img1);

cvShowImage("mal", ImgMal);

cvWaitKey(0);

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

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

char \*pImgMal;

if(fila<Img1->height/2){

pImgMal=ImgMal->imageData+(fila\*ImgMal->widthStep)+((Img1->height/2)\*ImgMal->widthStep);

}else{

pImgMal=ImgMal->imageData+(fila\*ImgMal->widthStep)-((Img1->height/2)\*ImgMal->widthStep);

//pImgMal=ImgMal->imageData+(fila-(Img1->height/2)\*ImgMal->widthStep); ASI NO VA, ES CON LA LINEA DE ARRIBA

}

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

\*pImgMal=\*pImg1;

pImg1++;

pImgMal++;

\*pImgMal=\*pImg1;

pImg1++;

pImgMal++;

\*pImgMal=\*pImg1;

pImg1++;

pImgMal++;

}

cvShowImage("mal",ImgMal);

cvWaitKey(10);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}