

BIL105E - Introduction to Scientific and Engineering Computing

Homework-4

Assignment Date : 22.04.2010
Due Date : 20.05.2010 at 22:00

- Submit two files to Ninova. File names must be your İTÜ student number.
 - 1) Your C source file (Example: 040090123.c)
 - 2) Your MS Word report file (Example: 040090123.doc)
- You should use the standard report format (Yazılım Ödevleri Klavuzu) which is available at Ninova. *Flowchart is not required for this homework, but an outline of algorithm (pseudocode) is required in the report.*
- Make sure gcc compiles your code successfully on Unix/Linux.
- Cheating is unacceptable and subject to disciplinary actions. All submitted programs will be cross-checked by using an automatic detection system.
- Late submissions through email are not accepted.

IMPORTANT

The following information must be added at the beginning of your C source code, otherwise 10 points will be deducted as penalty from your homework grade.

```
/*  
Course      : Bill105e  
Term       : 2010 Spring  
Homework   : #4  
Student Name : Aaa Bbb  
Student Number : 123456789  
*/
```

HOMEWORK DEFINITION

Write a C program to read a color image file in PPM (Portable Pixel Map) format, then generate 6 new image files that are described below, also in PPM format.

Note that, this homework is not about displaying the image, but it is only about processing the image.

INPUT

First, your program should ask user to enter the name of image file, as shown below:

```
Enter name of input image file :  ari.ppm
```

Then the program should read all informations from that PPM file.

(Your C executable program and the image file must be in the same directory.)

Attention: In this homework, input/output redirection method should not be used.

You must use file functions such as fopen, fscanf, fprintf, and so on.

Standard Format of PPM Image Files

- PPM files are color image files which can be viewed by the *iview* program. (This freeware program has been made available at Ninova.)
- PPM files are text files; when you open a PPM file with a text editor such as Notepad, for example the following information will be shown.

Example PPM Image File

	P3
Dimensions →	201 238
Max color →	255
Red, Green, and Blue values of first pixel →	236 105 138

```
140 138 138 137 141 140 141 140 140
140 142 142 144 162 220 251 255 255 255 255
215 214 214 213 212 210 209 209 207 206 203 202
200 199 200 198 197 197 196 196 195 194 192 191
191 189 186 184 184 184 184 184 184 183 182 182
181 179 176 176 175 174 173 172 171 172 171 171
171 169 168 250 251 252 253 253 253 253 253
254 253 253 253 253 253 253 253 252 252 251 251
253 253 252 251 251 251 251 251 252 251 250 249
249 249 250 251 250 250 250 249 248 247 246 246
.....
.....
```

- The first line contains the file format information. It must be P3 for all PPM files.
- The second line contains the Dimensions (Columns and Rows) of the image.
- The third line contains the Maximum Color value (usually 255).
- The other numbers are the pixel values of the image, from-left-to-right and from-top-to-down order.

- **A pixel consists of three numbers: Red, Green, and Blue.**
- RGB numbers are separated with a blank space. The count of these RGB numbers in the file must be exactly equal to Columns * Rows * 3.

RGB Examples:

- (236 105 138) means the pixel is a mixture of red, green, and blue
- (255 255 255) means the pixel is white
- (0 0 0) means the pixel is black
- (255 0 0) means the pixel is red
- (0 255 0) means the pixel is green
- (0 0 255) means the pixel is blue
- (74 74 74) means the pixel is a tone of gray

An image can be represented with a matrix. Each cell in matrix represents a pixel that actually contains three numbers. For example the cell a contains a_{Red} , a_{Green} , a_{Blue} values.

Original Image (INPUT)

a	b	c	d
e	f	g	h
i	j	k	l



OUTPUTS

Your program should create all of the following 6 image files in PPM format.

1. Mirror.ppm : Mirror image thru right vertical axis
2. Rotate.ppm : Left rotated image by 90 degree
3. Negative.ppm : Negative image
4. Gray.ppm : Gray image
5. Small.ppm : Two times smaller image
6. Big.ppm : Two times bigger image

At the end, your program should display a message on screen as shown below:

**6 output image files have been generated.
PROGRAM ENDED.**

The following examples explain how to perform the required image processing operations.

1. Mirror Image

Thru right vertical axis.

<i>d</i>	<i>c</i>	<i>b</i>	<i>a</i>
<i>h</i>	<i>g</i>	<i>f</i>	<i>e</i>
<i>l</i>	<i>k</i>	<i>j</i>	<i>i</i>



2. Rotated Image

Left by 90 degree.

<i>d</i>	<i>h</i>	<i>l</i>
<i>c</i>	<i>g</i>	<i>k</i>
<i>b</i>	<i>f</i>	<i>j</i>
<i>a</i>	<i>e</i>	<i>i</i>



3. Negative Image

To find new Red, Green, Blue values,
subtract old Red, Green, Blue values
from Max Color (255).

<i>Max - a</i>	<i>Max - b</i>	<i>Max - c</i>	<i>Max - d</i>
<i>Max - e</i>	<i>Max - f</i>	<i>Max - g</i>	<i>Max - h</i>
<i>Max - i</i>	<i>Max - j</i>	<i>Max - k</i>	<i>Max - l</i>



4. Gray Image

$$\text{Avg} = (\text{Red} + \text{Green} + \text{Blue}) / 3$$

The new Red, Green, and Blue will be same as Avg.

<i>Avg of a</i>	<i>Avg of b</i>	<i>Avg of c</i>	<i>Avg of d</i>
<i>Avg of e</i>	<i>Avg of f</i>	<i>Avg of g</i>	<i>Avg of h</i>
<i>Avg of i</i>	<i>Avg of j</i>	<i>Avg of k</i>	<i>Avg of l</i>



5. Small Image

Skip every other column and row pixels.

<i>a</i>	<i>c</i>
<i>i</i>	<i>k</i>



6. Big Image

Repeat every column and row pixels twice.

<i>a</i>	<i>a</i>	<i>b</i>	<i>b</i>	<i>c</i>	<i>c</i>	<i>d</i>	<i>d</i>
<i>a</i>	<i>a</i>	<i>b</i>	<i>b</i>	<i>c</i>	<i>c</i>	<i>d</i>	<i>d</i>
<i>e</i>	<i>e</i>	<i>f</i>	<i>f</i>	<i>g</i>	<i>g</i>	<i>h</i>	<i>h</i>
<i>e</i>	<i>e</i>	<i>f</i>	<i>f</i>	<i>g</i>	<i>g</i>	<i>h</i>	<i>h</i>
<i>i</i>	<i>i</i>	<i>j</i>	<i>j</i>	<i>k</i>	<i>k</i>	<i>l</i>	<i>l</i>
<i>i</i>	<i>i</i>	<i>j</i>	<i>j</i>	<i>k</i>	<i>k</i>	<i>l</i>	<i>l</i>

