**BIL105E**

Introduction to Scientific and Engineering Computing

2010 Spring

**Report of HW4**

Date of Submission : 18.05. 2010

Student Name : ABDULLAH AYDEGER

Student Number : 040090533

Instructor : YAŞAR ERENLER

CRN : 20959

**1-Introduction**

The purpose of this homework is to develop to C program to read a color image file in PPM (portable pixel map) format, then generate 6 new image files. 6 new images:

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

**2-Development and Operating Environments**

MS Windows

The Dev-C++ environment has been used to write the source code, compile and run

the program.

Unix

The source code has been also copied to Unix, then compiled and tested with the

GNU C Compiler. The following is the commands used:

To compile : gcc hw4.c –o hw4.exe

To run : hw4.exe

**3-Data Structures and Variables**

struct {

int red //each res’s pixels red color

int gren //each res’s pixels green color

int blue; //each res’s pixels blue color

}res[boyut][boyut]; //struct’s name

int i; //loop counter

int j; //loop counter

char format[3]; //PPM format

char filename[20]; //filename that will be opened

int column; //count of columns of ppm file

int row; //count of rows of ppm file

int maxcolor; //maximum color value

FILE \*ptr; //ppm file pointer

FILE \*mirrorptr; //pointer of mirror image thru right vertical axis

FILE \*rotateptr; //pointer of left rotated image by 90 degree

FILE \*negativeptr; //pointer of negative image

FILE \*greyptr; //pointer of grey image

FILE \*smallptr; //pointer of 2 times smaller image

FILE \*bigptr; //pointer of 2 times bigger image

4-Program Flow

**Pseudocode**

Begin

Input filename

ptr = filename

Input ptr format, column, row, maxcolor

for j=0 to (row-1) step 1

Begin

for i=0 to (column-1) step 1

Input ptr : res[j][i].red, res[i][j].green, res[j][i].blue

End

Output mirrorptr format, column, row, maxcolor

for j=0 to row step 1

Begin

for i=column-1 to 0 step -1

Begin

Output mirrorptr res[j][i].red,res[j][i].green,res[j][i].blue

End

Output mirrorptr \n

End

Output rotateptr format, column, row, maxcolor

for j= column-1 to 0 step -1

Begin

for i=0 to (row-1) step 1

Begin

Output rotateptr res[i][j].red,res[i][j].green,res[i][j].blue

End

Output rotateptr \n

End

Output negativeptr format, column, row, maxcolor

for j=0 to (row-1) step 1

Begin

for i=0 to (column-1) step 1

Begin

Output negativeptr 255-res[j][i].red,255-res[j][i].green,255-res[j][i].blue

End

Output negativeptr \n

End

Outptr greyptr format,column,row,maxcolor

for j=0 to (row-1) step 1

Begin

for i=0 to (column-1) step 1

Begin

Output greyptr (res[j][i].red+res[j][i].green+res[j][i].blue)/3 , (res[j][i].red+res[j][i].green+res[j][i].blue)/3 , (res[j][i].red+res[j][i].green+res[j][i].blue)/3

End

Output greyptr \n

End

Output smallptr format,column/2,row/2,maxcolor

for j=0 to (row-1) step 2

Begin

for i=0 to (column-1) step 2

Begin

Output smallptr res[j][i].red,res[j][i].green,res[j][i].blue

End

Output smallptr \n

End

Output bigptr format,column\*2,row\*2,maxcolor

for j=0 to (row-1) step 1

Begin

for i=0 to (column-1) step 1

Begin

Output bigptr res[j][i].red,res[j][i].green,res[j][i].blue

Output bigptr res[j][i].red,res[j][i].green,res[j][i].blue

End

Output bigptr \n

for i=0 to (column-1) step 1

Begin

Output bigptr res[j][i].red,res[j][i].green,res[j][i].blue

Output bigptr res[j][i].red,res[j][i].green,res[j][i].blue

End

Output bigptr \n

Output stdout

6 output image files have been generated . \n

PROGRAM ENDED . \n\n

End

**5-Conclusion**

In this homework, I have learned the followings :

--I’ve learned to use file functions such as fopen, fscanf, fprintf.

--I’ve learned to use some complicated matrix

--I’ve learned to calculate transpose of matrix