project[5]: multidimensional arrays/images

Due Thursday, 3/10/2016, 12:59:59pm

Project Goals

The goals of this project are to:

1. Get you familiar with multidimensional arrays
2. Introduction to reading/writing files
3. Learn about images

Important Notes:

1. **Formatting:** Make sure that you follow the precise recommendations for the output content and formatting. Your assignment will be auto-graded and any change in formatting will result in a reduced grade.
2. **Comments:** Header comments are required on all files and recommended for the rest of the course. Points will be deducted if no header comments are included.

Background

For this assignment, we will be creating and editing images. Images are a two-dimensional array of pixels. In this case, we will be using PGM images. This is a much simpler image format than JPEG, GIF, or PNG. Here is an example PGM image:

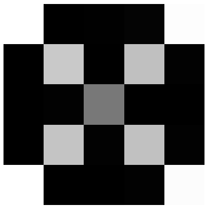
```
P2
# test.pgm
5 5
255
255 0 0 0 255
0 192 0 192 0
0 0 128 0 0
0 192 0 192 0
255 0 0 0 255
```

the first four lines are the image header:

```
P2 -> magic number
# test.pgm -> file name
5 5 -> size of the image (<width in pixels> <height in pixels>)
255 -> numbers of grey levels (in this case one pixel has 8 bits [0,255], you
could also enter 1 for a black and white image [0,1])
```

The next five lines are the image data itself, each number is a pixel value. If you take those 9 lines, and save it to a file (with a .pgm extension), you can open it as an image. You’ll see something like this:

t



You’ll see that 255 is white, 0 is black, and values in-between are varying shades of grey.

For more information on image file formats, see this article on Wikipedia (https://en.wikipedia.org/wiki/Image_file_formats)

Problem 1 (image_1.c)

1. Prompt a user to enter an integer and store it into a variable (let’s say: n).

```
2. If the integer is odd
        then create an all-black image of size n x n
        store this image in a multidimensional array
        print that image to the screen (for the above it would be 9 lines)
else
        print “You must enter an odd number”
        create a n+1 x n+1 size image (so 8 -> 9x9 image)
```

3. edit the array so that the image is now an X, where the X is in white on a black background, one pixel wide, going from each corner to the corner across.

4. print the image to the screen

So, if a user enters the 11 as the image size, the program will print the following:

```
P2
# test.pgm
11 11
255
255 0 0 0 0 0 0 0 0 0 255
0 255 0 0 0 0 0 0 0 255 0
0 0 255 0 0 0 0 0 255 0 0
0 0 0 255 0 0 0 255 0 0 0
0 0 0 0 255 0 255 0 0 0 0
0 0 0 0 0 255 0 0 0 0
0 0 0 0 255 0 255 0 0 0 0
0 0 0 255 0 0 0 255 0 0 0
0 0 255 0 0 0 0 0 255 0 0
0 255 0 0 0 0 0 0 0 255 0
255 0 0 0 0 0 0 0 0 0 255
```

Problem 2 (image_2.c)

1. Prompt a user to enter an integer and store it into a variable (let's say: n).

```
2. If the integer that divided by 3 would have a remainder of 2
        then create an all-black image of size n x n
        store this image in a multidimensional array
else
        create an image with dimensions add either 1 or 2 to make the dimensions when divided by 3 have
a remainder of 2 (so 9 -> 11x11 image, 16 -> 17x17 image)
```

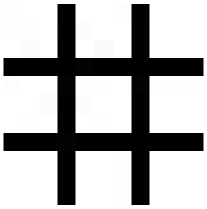
3. edit the array so that the image is now a black tic tac toe board on a black background, one pixel wide, with every cell of the board being equal.

4. print the image to the screen

So, if a user enters the 11 as the image size, the program will print the following (this is a black and white image):

```
P2
# image.pgm
11 11
1
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
1 1 1 0 1 1 1 0 1 1 1
```

and this would be the resultant image:



Challenge 1:

write the images to a file named image.pgm this is done by

// opening a file (using fopen)
// writing to the file (using fprintf, usage much like printf)
// closing the file (using fclose)

The following example shows the usage of writing to a file.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   FILE * fp;

   fp = fopen ("file.txt", "w+");
   fprintf(fp, "%s %s %s %d", "We", "are", "in", 2012);
   fclose(fp);

   return(0);
}
```

Submission details

The project needs to be submitted by Thursday, 3/10/2016, 12:59:59pm.

To submit your project, you will have to save your project files to an ECC machine using the Linux VM or the nomachine client:

- create a directory called “project5”
- save your *.c files in that directory
- save your description file into that directory
- DO THIS ONCE: Install the submission script ***<u>(don’t type the ‘>’ symbols)</u>***
  ```
  > cd ~
  > wget http://www.cse.unr.edu/~newellz2/submit
  > chmod +x ./submit
  ```
- TO Submit:
  ```
  > cd project5
  > ~/submit
  ```

The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses” constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include canceling a student’s enrollment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.