

Student Name: Francisco Gallardo  
University ID: u5296823

I completed the following programs for Introduction to Computer Systems Assignment 1:

- Command-line BMP header parser
- Threshold filter
- Greyscale filter EXTENSION
- Inverted colours filter (negative filter effect) EXTENSION

The experience of coding in C programming language made me more proficient in incorporating core Software Systems concepts. Even though this Assignment might be an easy task for advanced programmers, I really enjoyed working on it, because I implemented basic C-programming language skills, and learned a bit more about the foundation of a functional programming language. When I finished my program I realized new ideas came to my mind, and wanted to do more options and new functions to modify images using C programming language.

For my next programs I would like to spend more time blueprinting and designing “how” the program is actually going to work, because I think it is crucial to be prepared in case something doesn’t work as expected. Next time I would like to implement C functions to make my code much simpler, shorter and readable, but this implies that I have to obtain much more experience. I learned that in order to achieve better results, it is essential to have a deep knowledge of the pre-built functions that a specific Programming language like C can offer. I also realized that the best way to master any programming language and its functions is reading relevant bibliography and keep practicing. I would also like to learn how to use an IDE for C, but I am still deciding which one is the most appropriate for an apprentice like me.

My programs work in the following way:

\*All the source image files must be contained in the same folder as the program executable file.

1. Command-line BMP header parser:

This program reads bytes located in the 18<sup>th</sup>, 19<sup>th</sup>, 22<sup>th</sup> and 23<sup>th</sup> positions of the header of a .bmp image file, converts them into actual data, and displays the results on command prompt.

**USAGE:** The user types two arguments in the command prompt: The name of the program and the name of the image to be measured by the program.

INPUT	OUTPUT
./bmpedit cup.bmp	Image width: 413 Image height: 413

## 2. Threshold filter:

This program copies a full coloured image and make a new file containing a black and white version of it, using a threshold value given by the user.

To do this, the program reads all the bytes from the source file and puts them into a new target file using the `fgetc` and `fputc` functions inside a while loop. After this, it makes an average of every three Bytes corresponding to the base colours Red, Green and Blue contained in every pixel. Divides this averaged value by 765 (to convert this average in a range between 0 and 1). Then, depending on the threshold value given, it re-assigns each byte to a new value using a do-while loop: 255 (white) if the average is above the threshold value, and 000 (black) if it is below the threshold value. Each time a byte is reassigned with a new value, it is also copied to the target file.


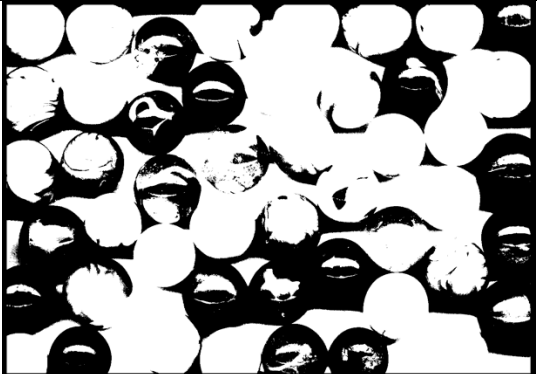
### USAGE:

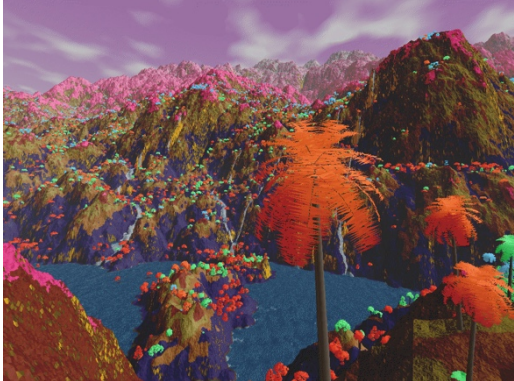

If the user types four arguments:

The name of the program + “-t” + the threshold value (a decimal between 0 and 1) + the name of the source image file. The program creates an output file named “out.bmp”.

If the user types five arguments:



The name of the program + “-t” + the threshold value (a decimal between 0 and 1 + the name of the target file + the name of the source image file. The program uses the third argument to create a file with that name.

4 Arguments	
INPUT	OUTPUT
<code>./bmpedit -t 0.5 marbles.bmp</code>	<code>out.bmp</code>
	

5 Arguments	
INPUT	OUTPUT
<code>./bmpedit -t 0.5 fgc.bmp land.bmp</code>	<code>fgc.bmp</code>
	



### 3. Grayscale filter EXTENSION:

**USAGE:** This program copies a coloured image and makes a new file containing a greyscale version of it. To achieve this, the program reads all the bytes from the source file and puts them into a new target file using the `fgetc` and `fputc` functions inside a while loop. After this, it makes an average of every three Bytes corresponding to the base colours Red, Green and Blue contained in every pixel and re-writes this averaged value in the target file image data using a do-while loop.

INPUT	OUTPUT
<code>./bmpedit -g tulips.bmp</code>	<code>out.bmp</code>
	

#### 4. Inverted colours filter (Negative effect) EXTENSION

**USAGE:** This program copies a coloured image and makes a new file containing an image with an *inverted colours effect* version of the original.

INPUT	OUTPUT
<code>./bmpedit -n flowers.bmp</code>	<code>out.bmp</code>
	

#### Testing

Sometimes I tested my programs in separate .txt files. I ran small pieces of code in separate files and copy-pasted the working part of the code in my original file as soon as I achieved the expected results. I also found useful to save a copy of my original program as soon as I made a big advance on this Assignment. I tested my program with several kinds of images with different sizes and the program always works fine.

#### Why do I deserve a good mark?

Even though I am not a professional programmer yet, I believe I deserve a good mark, because I worked very hard for this Assignment, and I think I achieved the expected results for this task. Nevertheless, it is not a good idea to conform. This assignment made me comprehend how actually computer systems read instructions from text files and how they think and behave. I think Software Engineers always have new things to learn and it is not a good idea to conform with what we already know. Specially in Software and Technology, there are always new things to discover.

Where did I get information to build this piece of software?

- I found very detailed and accurate information of the BMP file header structure, in an extremely interesting book named Image Processing in C by Dwayne Phillips [1].
- I also got relevant web resources on image processing in C language in the website “Poesia Binaria”, created by Gaspar Fernandez [2].
- To copy all the bytes from one source file into a target file using fgetc, fputc and while loop funtions, I used some code from the Programming Simplified website [3].
- I found valuable information with examples on how to use the ftell function in the Tutorials Point website [4].
- To open an external .txt file and print out its contents for the -h option, I used code provided from TylarBen in StackOverflow.com [5].

#### References:

- [1] Phillips, D. (2000) Image Processing in C. 2nd Edition edn. Lawrence, Kansas 66046-0127: R & D Publications.
- [2] Gaspar Fernández. 2011. Poesia Binaria. [ONLINE] Available at: <http://totaki.com/poesiabinaria/2011/06/leyendo-archivos-de-imagen-en-formato-bmp-en-c/>. [Accessed 23 March 16].
- [3] Jonny (no date) Programming Simplified - C program to copy files. Available at: <http://www.programmingsimplified.com/c-program-copy-file> (Accessed: 23 April 2016).
- [4] Aydar, A. (no date) C library function - ftell() Tutorials Point. Available at: [http://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_ftell.htm](http://www.tutorialspoint.com/c_standard_library/c_function_ftell.htm) (Accessed: 21 March 2016).
- [5] TylarBen. 2012. Opening a file from command line arguments in C. [ONLINE] Available at: <http://stackoverflow.com/questions/9449295/opening-a-file-from-command-line-arguments-in-c>. [Accessed 31 March 16].