**Author:** Martin Tsvetomirov Marinov

**Matriculation number:** s0932707

**Name of the project:** Mosaic Image

**Description:**
The project uses an algorithm for creating mosaic images from an input image and a database of "tiles". If one looks at the mosaic image from distance it would appear as the original one, but on a closer inspection it could be noticed that it is actually composed of smaller images. To achieve this goal programmatically, a couple of auxiliary algorithms had to be written – for loading "bmp" image files and saving them to the disk, for manipulating pixels within an image, a colour filter which changes the "average colour" of an image, an alpha blending algorithm and an image resizing algorithm.

**How to use it:**
Just put a 24-bit bitmap image in the folder of the program, open "Mosaic.hs" with GHC and type in "main". After a while a file "output.bmp", containing the mosaic image, will appear in the folder. If the user wants to use his own tile images, he just needs to put them in the "tiles" folder and use the "analyze" command within "Mosaic.hs". If the user would like to change the default names of the files, he can edit "Settings.hs" with his favourite text editor and follow the instructions. There are advanced options for controlling the algorithm within the file such as defining the number of tiles within the new image, original tile size, changing the final blending with the original picture value, randomization factor, etc.
If the user wants to display the final image onto the screen instead of saving it to a file, he should use "MosaicDisplay.hs" which incorporates the same commands as "Mosaic.hs" except that instead of saving the output image to the screen, it shows it onto the monitor. Displaying images may not work on Linux machines (or Windows machines without GLUT library)
If you plan to use "Mosaic.hs" on Linux, you might need to recompile the default database with the "analyze" command.
If you want to manipulate large images you must use "Run.bat" on Windows or just call ghci with the command "+RTS -K100M" and open "Mosaic.hs" on other OSs. This command sets the stack size big enough to handle a large amount of data, but you have to be patient – bigger images mean more time for generating and saving.

**Implementation:**
The script downsizes the input image, so every pixel of the new smaller image can be substituted by a "tile". To find which tile corresponds to each pixel, the program reads the database which contains the tile file path and the "average" pixel represented as an integer colour from the palette. Using trees for optimized searching performance, it finds the closest possible match and then enhances the tile so it average colour would blend completely with the source pixel colour. Every tile is resized before being "drawn" into the new bitmap to save memory. After a tiled image has been generated, the original image is alpha blended so that there would be some details in the final mosaics that have been omitted during the downscaling of the original image.

**Problems:**
One of the major problems is the extremely slow writeFile function which is really painful to use on big files.

**Information taken from:**
Lots of tutorials over the net, and especially Wikipedia for the BMP file specifications