# Chessboard

This is a **pair** exercise and must be competed in your **tutorial** or **lab** with your partner.

In this activity you need to complete the function `drawChessboard` in the provided program.

Download `chessboard.c`, or copy it into your current directory on a CSE system by running

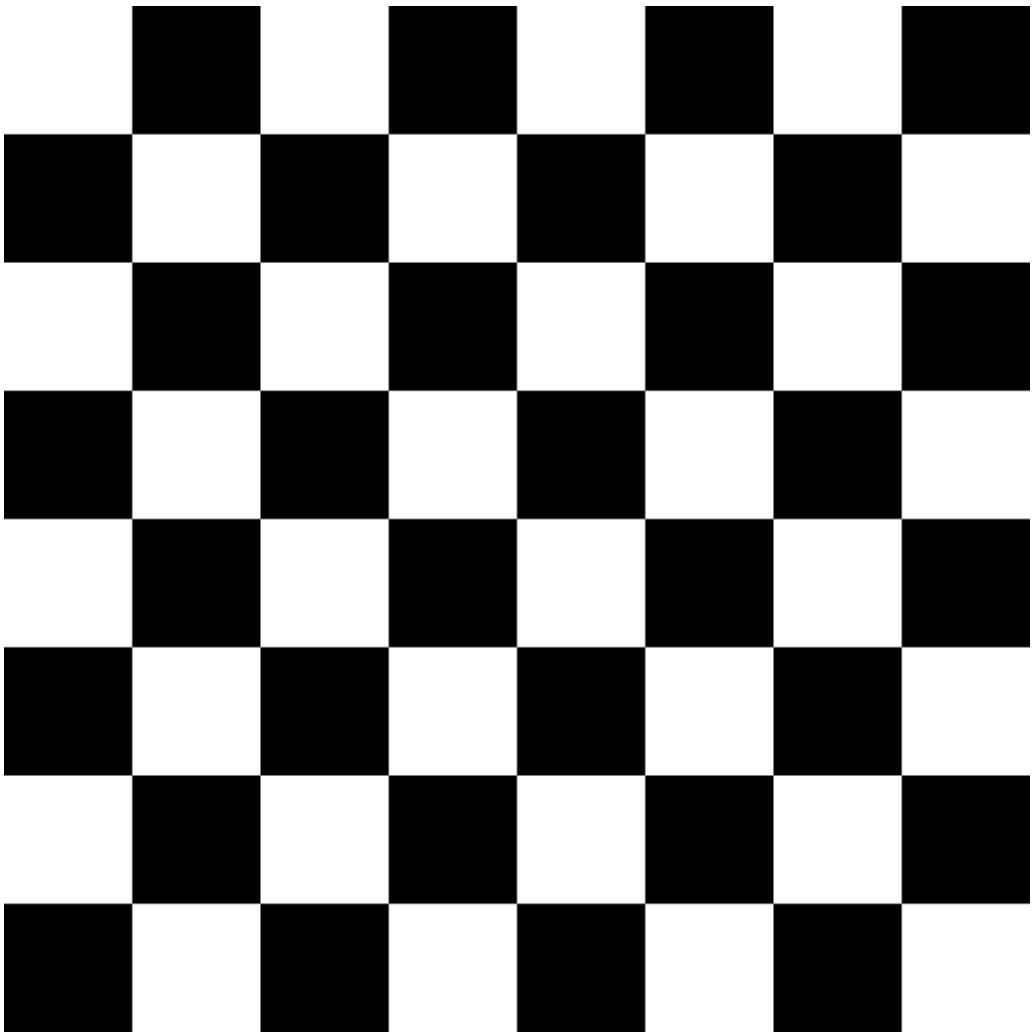
```
$ cp /web/cs1511/17s2/week06/files/chessboard.c .
```

You are provided with the following representation of a pixel:

```
typedef struct _pixel {
    unsigned char red;
    unsigned char green;
    unsigned char blue;
} pixel;
```

You need to complete the `drawChessboard` function so that it fills in its given 2-dimensional pixel buffer with a chessboard image of 8 columns and 8 rows with alternating white and black squares and a white square in the top left corner.

Your image should look identical to this one



> Remember, the pixel buffer is used `pixels[y][x]`, with rows first. The bottom-left corner of the image is the start of the pixels and corresponds with `pixels[0][0]`.

The program will send the data for the image to the screen output. To send it into a file instead run it using the following commands:

```
$ dcc -o chessboard chessboard.c
$ ./chessboard > chessboard.bmp
$ eog chessboard.bmp
```

To run some simple automated tests:

```
$ 1511 autotest chessboard
```

To run Styl-o-matic:

```
$ 1511 stylomatic chessboard.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk06_chessboard
```

Or, if you are working from home, upload the relevant file(s) to the `wk06_chessboard` activity on Give Online.