Practical 8

In this prac you will use basic types and pointer types, write functions, use loops and if statements.
You should not make use of any global variables in this prac.

Part 1:
Define an automatic variable in main called *array*. It should be an array of **int8_t**'s and be initialised to the 40 values as specified in the assignment on Vula.
(sometimes copying text from this .pdf doesn't work well)

Define two int8_t's called *min* and *max*.
They should both be initialised to the starting value of the array.

Part 2:
Declare and define a function called:
find_min_max

The function should take as arguments:

1. a pointer to the start of an array of int8_t's, called *array*
2. the length of the array, called *length*. The type should be uint32_t.
3. a pointer to a int8_t, called *max_ptr*
4. a pointer to a int8_t, called *min_ptr*

The arguments should be defined in that order to keep the automarker happy.
The function should return void.

Part 3:
Implement the find_min_max function.
That is, write code which will:

- iterate through all elements of the array
- compare each element to the data pointed to by min_ptr and max_ptr
- if the array element is larger or smaller than the data at max_ptr or min_ptr respectively, update the data pointed to by the pointer to hold the new value

The end result of this should be that the data at max_ptr takes on the largest value in the array, while the data at min_ptr takes on the smallest value in the array.

In main, call the find_min_max function such that once it has executed the variables min and max hold the minimum and maximum values in the array.

Part 4:
In the infinite loop, toggle between displaying the the min and max value found on the LEDs.
There should be a 1 second delay in between displaying each pattern.

You’re advised to implement a function called delay which causes a 1 second delay when called by running a long but finite loop.
It’s very difficult to calculate the length of a delay loop in C. You’d have better luck trying something out and tuning it on a scope.
Suggestion: start with 80 000 loop iterations.

Marks:
Part 1: 2
Part 2: 2
Part 3: 3
Part 4: 3