Objectives: This laboratory aims to practice using arrays in C on Arduino.

Learning outcomes: Declare and initialize arrays and access array elements

Lab instructions

1. The lab is for individuals working alone and must be completed during the lab session
2. Create a new sketch for each major section in the lab.
3. Before you leave the lab, call the lab demonstrator to check what you have done for all the sections (this is why separate sketches are important). *Anything not checked by the demonstrator during lab will have to be assigned zero marks.*
4. Create a single plain text submission file (.txt) for the lab using a plain text editor (e.g. NotePad++). Copy all the sketches you write and any other answers required for the lab into the submission text file. *Name the file “108_Lab5_firstname_surname.txt”, using your actual name. Include your name and lab number at the top of the submission file also and clearly label everything in the file. Unclear submissions will have to be marked down or (in the worst case) not marked at all.*

Marking for lab/assignment

Most of the lab and assignment will be marked during lab sessions. It is essential that you get the demonstrator to confirm your progress during the lab (for the lab portion) and at the start of the next lab (for the assignment portion).

For all code sections, marks will be deducted for bad communication and style (e.g. missing or mismatching comments, poor variable names, bad indentation, inappropriate use of global variables, unnecessary code repetition, etc.) and incorrect behaviour, or failure to follow the requirements of the question.

General marks will also be lost if the submission document instructions are not followed.

1 Basic arrays – save sketch as “Lab5_BasicArrays”

Background: review notes 2.10 Arrays.

The objective of this first exercise is to practice declaring an array with initial values, iterating over an array to get the value of each element, and iterating over and array to set the value of each element.

You do not need to write your own helper functions for this first sketch.

Lab5_BasicArrays requirements:

- Download and extract the starter sketch on moodle to your sketchbook. This starter sketch helps with button click detection. You must use the readSwitchEvent function to detect button clicks.
- Declare a constant at the top of the file to use as the array size. You must use #define for this constant. For now set the array size to be 4.
- Inside the loop function declare an array of int values that will remember its element values from one run of the loop function to the next.
    - The array size should be specified using the size constant defined previously.
    - Initialize the array elements to be all zeros. You should do this by specifying an array initializer (see notes) and only specifying the value of the first element as 0. The compiler then automatically sets all remaining (unspecified) element values to zero also.
- Every time the loop function runs
    - Check if SW1 has been normally clicked and if it has, iterate over the array and print the current value of each element. (You will need to get the value of each array element.)

      Hints: the notes show how to iterate over an array using a for-loop. You want to simply Serial.print each element and then Serial.print(“, “) after each element (except perhaps the last one if you want to get it perfect). Finally, print a new line after all array elements have been printed.

    - Otherwise, if SW1 has been long or very-long clicked, iterate over the array and replace each element with a random number between 0-99 inclusive. (You will need to set the value of each array element.) Then iterate over the array again and print the latest values. (You will need to get the value of each array element.)

      Hint: Remember that random(N) selects random numbers in the range 0-(N-1) inclusive.

- Now change the value of your array size constant to 8. Everything should still work and the sketch output should appear as:

```
Lab5_BasicArray starting...

[Click] array is { 0, 0, 0, 0, 0, 0, 0, 0 }
[LongClick] modified array { 7, 49, 73, 58, 30, 72, 44, 78 }
[Click] array is { 7, 49, 73, 58, 30, 72, 44, 78 }
[Click] array is { 7, 49, 73, 58, 30, 72, 44, 78 }
[LongClick] modified array { 23, 9, 40, 65, 92, 42, 87, 3 }
```

✏ Copy the sketch into your answer document and demonstrate it.

2 arrays and functions – save sketch as “Lab5_ArraysAndFunctions”

Background: The objective of the second exercise is to practice using arrays with functions. We achieve this by moving some code that operates on the array out of the loop function into their own functions and then passing our array into those functions when we call them.

Lab5_ArraysAndFunctions requirements:

- Start with your solution to part 1.
- Move the code that iterates over an array and prints the value of each element into its own function (e.g. printArray). Then, replace the two places in the loop function that you have code to iterate over the array elements printing their values with a call printArray.
    - You will need to pass in both the array to operate on and the length of the array as parameters to the function. Refer to notes 2.10.
    - Note that the printArray function must not use the constant for the array size in its code – it must use a parameter passed in.

    (If it did use the constant directly, then printArray would only ever work with the specific array in this lab exercise which would not make it a very flexible/reusable function. Instead, when we write functions we want to make them general so that they can be reused in other programmes also. For example we might have a programme with multiple different size arrays and we would still like to be able to use our printArray function to print the contents of each of those arrays.)

- Move the code in the loop function that iterates over the array and sets each element value to a new random number into its own function (e.g. randomizeArrayValues). Then replace the code in the loop function with an appropriate call to the new randomizeArrayValues function.
    - Enhance the randomizeArrayValues function with extra parameters so that you can also specify the range of random values to use when modifying the elements.

✏ Copy the sketch into your answer document and demonstrate it.