Lab 3 - Exercises

- Please use the notes as a reference for answering these questions.
- The purpose of these exercises is that you understand the notes!

You can also use online resources such as:

https://www.programiz.com/c-programming/c-arrays https://www.w3schools.com/cpp/cpp_arrays.asp

I have given quite heavy hints for this as I know that some of you haven't coded much or used C++. Please ask for help where needed!

Exercise 1: Array Size in Memory

Question: Create an array of type `int` that stores 10 values. What is the size of this in memory?

Hints:

- Use the notes to see how to create an array!
- Use the `sizeof` operator to determine the size of the array in memory. e.g if you have an array called `myArray` then `sizeof(myArray)` will give you it's size.
    - An array of 5 integers will occupy `4 * 5 = 20` bytes in memory. Your code should output the right value for 10 integers.

Code Hint:

```
#include <iostream>

int main() {
    int i = 32;

    std::cout << "Size of the array in memory: " << sizeof(i) << std::endl;

    return 0;
}
```

Exercise 2: Printing Array Elements

Question: Print out each element in from the integer array you created in exercise 1.

Hints:

- Use a `for` loop to iterate through the elements of the array.
- Print each element in the loop.

Code Hint:

```
#include <iostream>

int main() {
    int myArray[3] = {1,2,3}; // Create an array of 3 integers
    // Print each element in the array
    for (int i = 0; i < 2; i++) {
        std::cout << "Element " << i << ": " << myArray[i] << std::endl;
    }

    return 0;
}
```

Exercise 3: Updating Array Element

Question: Update the 3rd element of the array to be 42 that you created in exercise 1.

Hints:

- Use the array index (remember that indices start from `0`) to update the desired element.

Code Hint:

```
#include <iostream>

int main() {
    int myArray[3] = {1,2,3}; // Create an array of 3 integers

    std::cout << myArray[1] << "\n";

    myArray[1] = 12; // Update the 2nd item in the array to 12

    std::cout << myArray[1] << std::endl;

    return 0;
}
```

Exercise 4: Array Memory Addresses

Question: Find out the base address of the array and then work out the addresses of the 3rd, 5th, and 8th elements of the array. Draw this as a picture in memory.

Hints:

- Use the address-of operator `&` to get the base memory address of the array.
- Use this base address to calculate the addresses of the above elements. **See this weeks notes!**

Code Hint:

```
#include <iostream>

int main() {
    int myArray[3] = {1,2,3}; // Create an array of 3 integers

    // Calculate the base address of the array
    int* baseAddress = &myArray[0];

    // Print the addresses
    std::cout << "Base Address: " << baseAddress << std::endl;

    return 0;
}
```

Exercise 5: Creating a String with Char Array

Question: Create a string using a char `array`. Create the same string using the `string` from the string library. Print both of these strings.

Hints:

- You should find examples for this in the notes.

Exercise 6: Memory Requirements of a 2D Array

Question: Create a 2D array of doubles with 3 rows and 4 columns and determine how much memory this requires. What is the base address? Put any values you like into the 2D array.

Hint

- You did something similar in exercise 1 using `sizeof`.

Code Hint:

```
#include <iostream>

int main() {
    int matrix[2][3] = {{3, 2, 4}, {4,2,1}}; // Create a 2D array of ints with 2 rows and three columns

    // Calculate the base address
    double* baseAddress = &matrix[0][0];

    std::cout << "Base Address: " << baseAddress << std::endl;

    return 0;
}
```

Exercise 7: Updating 2D Array Element

Question: Using the two-dimensional array you created in exercise 6. Update the 2nd row and 3rd column to be `3.142`.

Hint

- Indexing starts at `0`, so if you have an array called `myArray` then `myArray[0][2]` refers to row 1 and column 3.