Week 3 – SOFT7019 lab session

This week we will utilise an online C IDE called online gdb, please access it at <https://www.onlinegdb.com/>

In the top right corner, you will have the option to select the programming language, please select C.



# Exercise 1

**Looking at memory: variables of type int**

* Create an integer variable num1 and set it to 0xFFFF.
* Print the current value of num1 in decimal and hex
* Print out the number of bytes and bits for used for the num1 variable.
* Print out the maximum and minimum value that can be stored in a num1 as a decimal.

The output should look like this:

num1 has a value of 65535: FFFF

num1 uses 4 bytes: 32 bits

Max/Min value in num is 2147483647 / -2147483648

**Hints:**

1. Use a printf to print numl using either %d or %x
2. To find out the number of bytes used by a variable use the sizeof operator. ie sizeof(num1).You can print this using %d. Also remember there are 8 bits in a byte.

(iii)The constant INT\_MAX or INT\_MIN defines the max/min value that can be stored in an integer.

You can print this using %d or % x. You will need to include limits.h to use them.

# Exercise 2

**Reading an input and changing the case**

The user will input text consisting only of letters in lower or upper case, and spaces. Print the same text in the console but change the case of the letters.

Example:

Input: tHIs is A test

Output: ThiS IS a TEST

Hints:

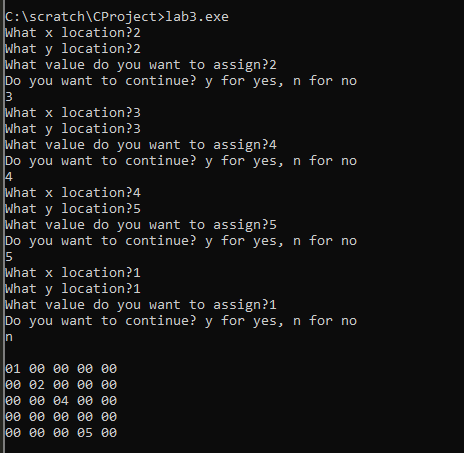
* use getchar() to read the input
* getchar() processes whatever is in the input buffer; you don’t need to read in one character at a time but can input the whole text at once (while loop)
* use the ASCII table to determine how you can change case
* characters that are not letters are printed without change
* stop processing input when you read ’\n’. (termination condition)

# Exercise 3

**Indexing and printing a 2D array**

* Generate a 5x5 2D array of integers and initialise all elements to 0
* Ask a user to input an x position, a y position, and a value
* Update the array at position [x][y] with the value
* After each loop ask the user if they want to continue
* Continue asking the user for positions and values until they request to stop
* Print the updated 2D array of integers

See this example:



# Appendix 1

The ASCII character encoding table.

