### Advanced Exercises

### Note: Try to create user defined functions for some of these exercises and practice good naming conventions and correct use of indentation and code layout. Also remember to comment your code as this will help you to remember what the code actually does (Some of the code may be reusable in the future)

### Advanced Exercise 01

Write a program that generates into a result variable at least the first 10 numbers in the **Fibonacci** **sequence**, using only two other variables and addition (**and no iteration** if you know how). Verify that your code works using the debugger. Don't forget to comment your code, and document the process you went through to solve the problem and overcome any difficulties.

### Advanced Exercise 02

Write a program that requires the user to enter a name, a date, a place, an activity, an object, and a sensation. Create from this an interesting short story that is displayed to the user using this input. You will need to utilise C strings for this exercise,

### Advanced Exercise 03

Design and implement a program that:

1. Asks the user for two integers.
2. Asks the user for an arithmetic operation to perform on them, e.g. enter 1 to add the numbers, 2 to subtract, etc.
3. Output the result
4. Test the code with various inputs and the debugger to ensure it works correctly. Don't forget to test all possibilities

### Advanced Exercise 04

Design then write a program that implements a simple "choose your own adventure" style of game. The response to each question should lead to further questions depending on the choice, e.g. "Left or Right?", left might say "You are outside a castle, do you go in or walk on by?", right might say "A dragon blocks your path, do you try to slay it or put on your magic pants of invisibility?", etc. Test as usual.

### Advanced Exercise 05

Design and create a program that records two characters’ hit points (start with both having 100 health). In a loop, each player attacks the other for a random amount of damage – between 0 and 20. Display all this information, along with the new health values. Repeat the loop until one of the players is dead. Report on which player won, unless they both killed each other in which case report a draw.

### Advanced Exercise 06

Design and implement a program that initialises an integer array with four numbers between 0 and 9 for a PIN code. The program should then ask the user for four numbers (either individually or more advanced in one go), and check if the correct PIN is entered.

### Advanced Exercise 07

The strcmp() function returns zero if the compared strings are equal, but it also returns other values if they are not. Investigate how these return values work, then design and implement a program that asks for several strings to be input by the user. The program should then sort the strings into alphabetical order, and then output the sorted strings.