**COMP10050 Group BO Assignment 2**

**Group BO:**

Martina D’Argenzio 22789139

Jack Dunne 22483576

https://csgitlab.ucd.ie/jack-dunne626/bo-assignment-2.git

# How is the Gantt chart displayed?

The Gantt chart is displayed partially using file handling, and mostly by standard output. Firstly, the shell is cleared (regardless of OS). A file “format” is opened to later retrieve the length of the topmost row. Then, to get the length of the row dividers, the first cell is initialized as empty, with spaces padding the rest of the cell. The number of spaces is found by finding the length of the longest task name. The topmost row of months is then created. The length of the dividers is found by finding the length of the topmost row (in characters) via the file handling. Using this, dividers of a dynamically suitable size can be created. Tasks are then printed in a similar manner.

**How is the task mapped to the struct?**

scanf into a struct array with a given format in struct.h. There is no need to worry about buffer overflow and security in such a low-level program.

**How were tasks edited?**

scanf, overwriting the task selected by the user. The program searches for a task with the input given and then edits the struct of which it belongs if it exists.

**How was a circular dependency search implemented?**

Two circular dependency searches were implemented recursively and are called when the user enters “test” into the console upon viewing their Gantt chart. The first function, printDependencies, was used at the beginning of testing any one specified task. If a task was visited more than once, another function named “circularCheck” would be called. If a dependent task existed such that it had no dependencies, it would return a success. Otherwise, it would fail.

**What’s your ASCII art about?**

Our ASCII art is the CLion logo because it’s the IDE both of us used.