Assignment 1 Guidelines

Assignment 1 is due on Tuesday September 29.

- You are allowed to use standard library functions. The assignment should be submitted through **"Moodle"** as a tar file containing the source code for each question as well as a readme file. There should also be a makefile to compile the programs. The submission folder could look something like:

  readme.txt
  carbon.c
  ackermann.c
  newton.c
  makefile

- Any compilation error or warning will result in a mark deduction appropriate to the severity of the error. There will be some marks allocated for style and documentation, but the majority will come from the execution of the programs.

- Each file should have a comment at the beginning containing your name, id, the date, and the assignment name.

- For the readme file, it should contain the following:

  - name, id and assignment number
  - A brief description of how to run each program, e.g. any command line arguments needed.
  - Sample output demonstrating the proper functioning of the program.

- For each of the C files, any function should have a brief comment describing its purpose. Similarly, any section of code where it is not easily apparent what the code does should also have a short comment.

- For Question 3 of Assignment 1, the command *time* can be used to track the system time usage of any program. You should use this to measure the execution time of your program with different arguments.

Coding Style Guidelines

Part of your mark will reflect coding style. Although there is no single correct coding style, you should strive to make your code as readable and internally consistent as possible. Such programs are easier to test, debug, maintain and modify. Please consider the following points when writing code:

- All source files should have a comment header describing the purpose of the module/program; similarly, all sub-routines should have a comment header clearly describing its function, input, output and error conditions.

- Identifiers (variables, procedure names, etc.) should have descriptive names (readable) except loop counter, index etc.

- Literal constants should be avoided, use named constants for counting limits, array sizes and so forth.

- Indentation and use of whitespace should be consistent throughout the code.

- Ideally, a subroutine is a small block of code that performs a single function. If a function is getting too long or is doing too many things you should probably separate it into several separate subroutines.

- Commenting should be appropriate (you would not comment an increment statement as what it is doing is obvious)