Assignment Part 1 – Fortran

# Program Testing Procedure

Run program, observe console output. Every 3rd line should be "Fizz", every 5th line "Buzz", every 15th line "Fizz buzz".

Also make sure output formatting is correct (Fortran is finnicky with print formatting).

# Weekly Question

The most striking difference between Fortran and languages I’ve used before is the column-based syntax. This is due to Fortran’s origins with punch-cards; ancient technology obviously not relevant to any modern language. Most of the languages I use structure syntax with whitespace, braces, semicolons, etc. (Additionally, modern compilers’/interpreters’ greater capabilities compared to Fortran’s likely enable their handling of more flexible syntax.)  
Further exhibiting Fortran’s primitiveness are the PROGRAM, STOP, and END statements, which inform the compiler where the program starts and ends. In any modern language you would take it for granted that the compiler/interpreter could figure this information out for itself.

However, for the simple program we had to code, Fortran is not that different to something like C89, albeit more primitive. IF/ELSEIF/ELSE is the same as in modern procedural languages. There’s a DO loop, which is close enough to the modern “for” loop. The algebraic and boolean notation is largely the same as we use now (besides some differences in operator naming). The one function call to MOD() that I needed to use worked as I expected. Variables are declared with a type before use, the same as C.  
I suspect that Fortran’s difference would become even more apparent if I had to write a larger program, for example using subprograms.

# Reflection