FizzBuzz

In this chapter we build the classic tech interview question: FizzBuzz.

The idea is simple. Write a program that enumerates the integers from 0 to some stopping value, perhaps 100.

For each integer:

- If it is a multiple of 3, print Fizz

- If it is a multiple of 5, print Buzz

- If it is a multiple of both 3 *and* 5, print FizzBuzz

- Otherwise, if none of the above applies, print the integer.

The interviewer's hope is that you get twisted in knots trying to navigate the case where the integer is a multiple of both 3 and 5. There are many ways to solve this challenge.

One way might be to test for being a multiple of 15 *first* and print FizzBuzz if true. Then test against 3 and then against 5.

Another way is to accumulate the correct out by testing against 3 and adding Fizz to a buffer. Then test against 5 and if appropriate append Buzz to the buffer. Either the buffer was empty, in which case you get Buzz alone - or it already contained Fizz in which case the buffer now contains FizzBuzz. Finally, if *anything* is in the buffer, cause the buffer to be printed and append a new line.

In C++, the buffer could be a C++ string or a stringstream. In C you might think that you must resort to using an array of `char` to act as the buffer, filling it with `strncpy` or some such nonsense.

But you don't have to bother! `printf` is a buffered output stream. It won't print anything until it encounters a new line character.

In this program, we'll use this to buffer up either Fizz, Buzz or both then as indicated above, we'll end with a new line and BAM - whatever was in the `printf` buffer gets sent to the console.

Here is a video where we walk through the process of writing FizzBuzz from scratch in ARM 64 bit assembly language.

Here is the source code.

The video is long but there is much benefit to be had by watching and listening to another person's process as they write the code. **AND especially** listening and watching to them debug when things go wrong!