Tutorial 3

- I/O Tests
- Side effects
- Reading input
- Global variables & mutation

Reminder

- Make sure to thoroughly test your code. Passing the public tests does not mean you will get 100%.

q3ex: state manipulation

- Assignment 1 Q3 (a GOLD question):
  - Define a set of functions to manage a machine's supply of parts.
  - Frequent change of state: testing is *critical* to ensure correctness.

- q3ex
  - A pre-implemented, simplified version of Q3
  - Let's try it!
Side effects

There are 3 types of side effects in functions for now:

- print output
- read input
- mutate a global variable

// MANY SIDE EFFECTS

```c
int y = 2;

int main(void) {
    int x = read_int();
    printf("%d", y);
    if ((x = y - 1)) {
        printf("%d", x = x + 2);
    }
    if (x) {
        y *= y;
        printf("%d", y);
    }
    printf("\n");
}
```

Exercise: Take a running total of user input

- Define the following C function:

  // running_total(total) reads integers until failure and,
  // after each integer read, prints total + all integers
  // read so far (including the most recent one)
  // effects: reads input
  // may produce output

No global variables allowed!
Global variables & mutation

- Global variables are defined *outside* of functions (at the “top level”).
- A function that mutates a global variable *does* have a side effect.
- Even if a function does not have a side effect, its behaviour may depend on other mutable global variables.

Exercise: Managing a bank account

Implement a simple banking interface where we can:

- **deposit**: adds money (positive) to your bank balance
- **withdraw**: withdraws money (positive) from your bank account if this will not make it more negative than the overdraft limit, and returns true on success
- **get_balance**: gets the current bank balance
- **set_overdraft_limit**: sets a non-negative overdraft limit