In this lab, we’ll practice constructing iterative statements in C, once more conditioning the motions of Scribbler2 robots on readings from their sensors.

**Pairs**

Here are the pairs for today’s lab. Drivers are on the left.

- Tanner Tufto and Cory McCartan
- Lilly Webster and Colin Greenman
- Ella Nicolson and Erhaan Ahmad
- Mattori Birnbaum and Ying Zhang
- Josh Lavinand Tyler Williams
- Jong Hoon Bae and Anna Blindermann
- Saung Thuya and Lex Martin
- Jae Hong Shin and Faizaan Ali
- Yuyin Sun and Eli Salm
- Dennis Chan and Gemma Nash
- Wyatt Heritage and Sanjay Sudhir
- Sophie Gaschott and Zachary Susag

**Exercises**

Exercise 000: In exercise 101 of the lab on conditionals, you wrote a program that defined a *turn_towards_light* function and called it fifteen times in a row. Revise that program so that it contains a single call in the body of a *for*-loop that repeats that call fifteen times.

Exercise 001: Write, compile, and test a program that causes the Scribbler2 robot to produce nine successive one-second tones, the first one at a pitch of 256 hertz, and each subsequent one at a pitch one-and-a-half times as high as its predecessor. Instead of writing out nine calls to *rBeep*, use a *for*-loop containing an updating assignment to a pitch variable followed by a single call to *rBeep*.

Exercise 010: A programmer who wanted to use C’s “compound assignment” expressions might write the updating assignment in the preceding exercise thus:

```
pitch *= 3/2;
```

Explain why this assignment would not have the desired effect, and explain what its actual effect would be.

Exercise 011: Write, compile, and test a program that causes the Scribbler2 robot to blink its fluke-end LED lights in a particular pattern: successive groups of 1, 2, 3, 4 quarter-second flashes, with the groups separated by one-second pauses (during which the lights are off) and the individual flashes within each group separated by quarter-second pauses. The effect should be: blink; blink, blink; blink, blink, blink; blink, blink, blink, blink.

Exercise 100: Let’s try to get a siren effect on the Scribbler2 robot by having it produce a large number of very short tones, each just a little higher than the one before. We can use the *rBeep2* procedure to make two tones sound at the same time, with a fixed interval between them.
Suppose we start with two tones that are in the exact ratio 3:4, such as 600 hertz and 800 hertz. If we increase the pitch of the first one to 612 hertz and the pitch of the second to 816 hertz, and then to 624 and 832, and then to 636 and 848, adding 12 to the first pitch and 16 to the second each time, the ratio between the pitches will still be 3:4, and the interval will sound the same, although the tones themselves will be a little higher. If we repeat the process fifty times, both tones will be a full octave higher, but their ratio will still be 3:4.

Write, compile, and test a program that causes the Scribbler2 to produce these tones. Choose the duration of the tones so that running through the entire octave takes about fifteen seconds.

Exercise 101: Revise the program you wrote for the preceding exercise so that, after playing all the ascending-pitch tones, it runs through the same pitches again, but this time in descending order, then repeats the entire up-and-down process twice more.

Exercise 110: Revise the program you wrote for exercise 100 so that, instead of producing a fixed number of tones, it keeps track of the total duration of all the tones produced and exits from the loop when that total equals or exceeds twelve seconds.

Exercise 111: Write, compile, and test a program that runs through the integers from 0 to 99 and causes the Scribbler2 robot to emit a short tone for each integer: 600 hertz if the integer is divisible by 7, 800 hertz if either of its digits is 7, and 400 hertz if it has neither property. If it has both properties, the robot should play the 600 hertz tone and the 800 hertz tone at the same time.