### Exercise 1

Write a program where two goroutines pass an integer back and forth ten times. Display when each goroutine receives the integer. Increment the integer with each pass. Once the integer equals ten, terminate the program cleanly.

### Exercise 2

Write a program that uses a fan out pattern to generate 100 random numbers concurrently. Have each goroutine generate a single random number and return that number to the main goroutine over a buffered channel. Set the size of the buffer channel so no send ever blocks. Don't allocate more buffers than you need. Have the main goroutine display each random number it receives and then terminate the program.

### Exercise 3

Write a program that generates up to 100 random numbers concurrently. Do not send all 100 values so the number of sends/receives is unknown.

### Exercise 4

Write a program that generates up to 100 random numbers concurrently using a worker pool. Reject even values. Instruct the workers to shutdown with 100 odd numbers have been collected.