Test scripts for CSE201A

Please merge your homework with the repepctive test script repository. You should include a *Makefile* such that running make in the root directory of your submission produces a file in the root directory. The file, when executed, should read the strings via stdin and output the required content via stdout.

These test scripts are prepared based on Sohum Banerjea's work.

Instructions

Assignment 1

Assignment 1: ARITH was implemented in Golang. No additional dependencies are required since all parsing is done using handwritten parser combinators.

The go installation instructions can be followed from https://golang.org/doc/install

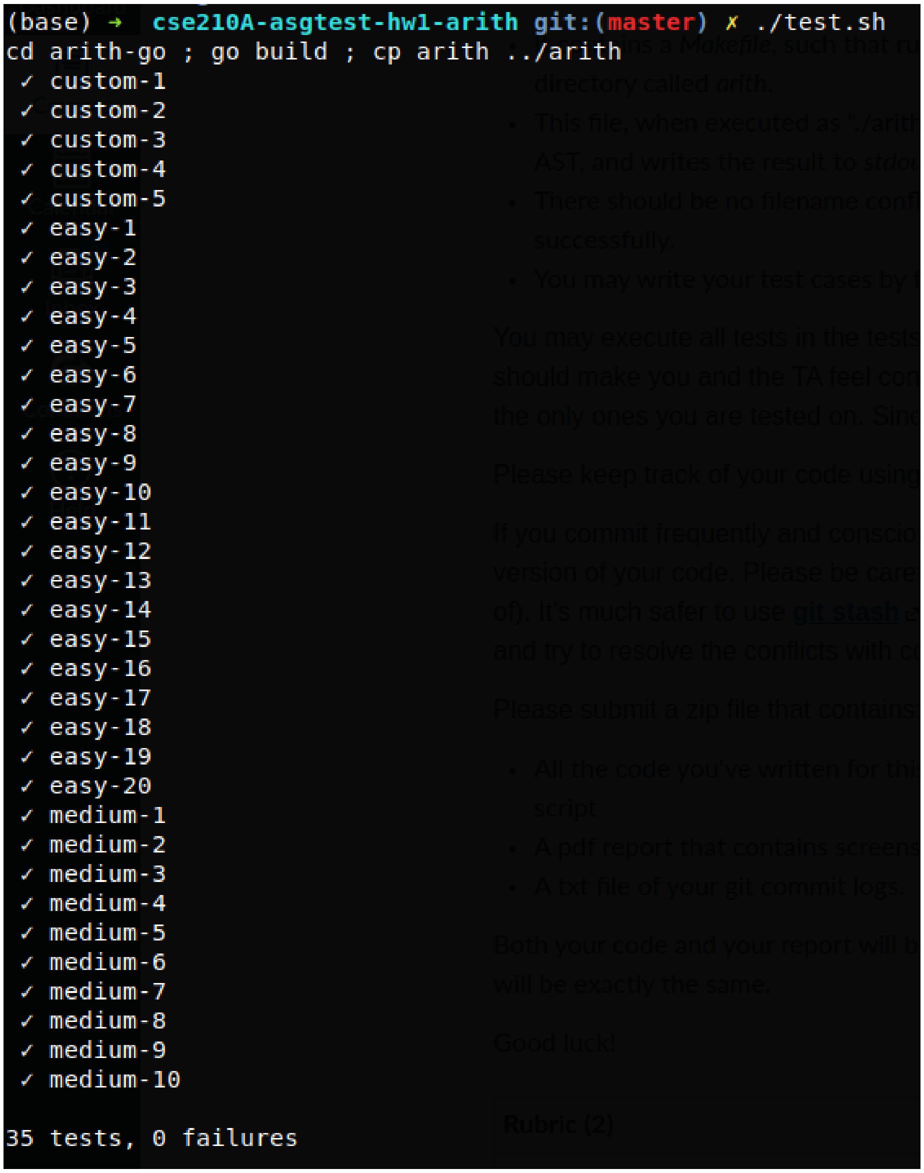
To build the executable, simply run

```
make
```

The implementation of ARITH was largely inspired by Armin Heller's posts: * https://medium.com/@armin.heller/parser-combinator-gotchas-2792deac4531 * https://medium.com/@armin.heller/parser-combinator-gotchas-2792deac4531

The key differences are that the original blog post evaluates the expressions inline instead of building a parse tree and in the assignment, some additional combinators are implemented.

Test Case Coverage



```
(base) ➜  cse210A-asgtest-hw1-arith git:(master) ✗ ./test.sh
cd arith-go ; go build ; cp arith ../arith
 ✓ custom-1
 ✓ custom-2
 ✓ custom-3
 ✓ custom-4
 ✓ custom-5
 ✓ easy-1
 ✓ easy-2
 ✓ easy-3
 ✓ easy-4
 ✓ easy-5
 ✓ easy-6
 ✓ easy-7
 ✓ easy-8
 ✓ easy-9
 ✓ easy-10
 ✓ easy-11
 ✓ easy-12
 ✓ easy-13
 ✓ easy-14
 ✓ easy-15
 ✓ easy-16
 ✓ easy-17
 ✓ easy-18
 ✓ easy-19
 ✓ easy-20
 ✓ medium-1
 ✓ medium-2
 ✓ medium-3
 ✓ medium-4
 ✓ medium-5
 ✓ medium-6
 ✓ medium-7
 ✓ medium-8
 ✓ medium-9
 ✓ medium-10

35 tests, 0 failures
```

Figure 1: Test cases