**Project Approach**

While trying to understand how to write and what to write for our week 5 discussion post, I ended up going ahead and starting on project three. As a lot of the course content for week five and week six went hand in hand. To be able to understand the some of these concepts, I thought working with project three for a hands-on approach would allow me to get a better understanding of what week five’s discussion post was asking. It also allowed me to get a head start of project three.

Reviewing the content for week six gave me a better grasp of how to construct the interpreter, and how the semantic rules are structured in the bison file (parser.y). I think the videos in the content for week six is where I got a lot of help for project three, which ultimately helped me complete the discussion post for week five. While following along with the course content and applying the concepts to project three, made me really want to finish project three before writing my discussion post (at least finish enough to compile my program and receive a calculated result). Once I got project three able to read and compiled a program that met discussion five’s instructions, I started further work on project three.

I ran into a few issues (will discuss further in the lessons learned section) that required a little more time and research from outside sources. However, for the most part I was able to get most of the information from our course content, mainly in week five and six.

**Test Cases**

Test Case 1:

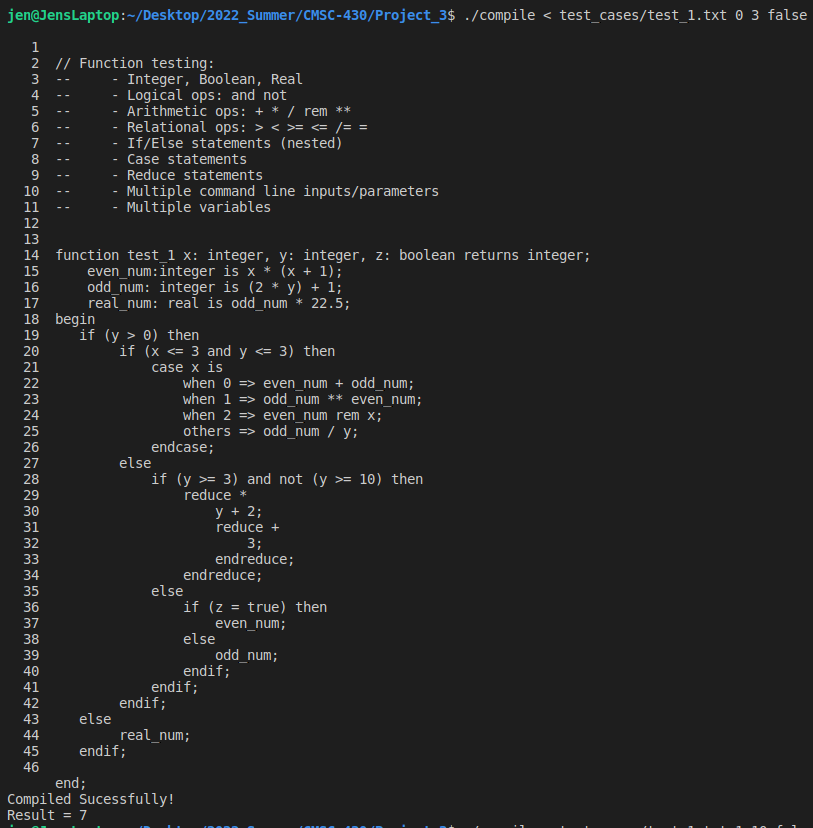
Testing for this test case will cover:

* Integer, Boolean, Real
* Logical ops: and not
* Arithmetic ops: + \* / rem \*\*
* Relational ops: > < >= <= /= =
* If/Else statements (nested)
* Case statements
* Reduce statements
* Multiple command line inputs/parameters
* Multiple variables

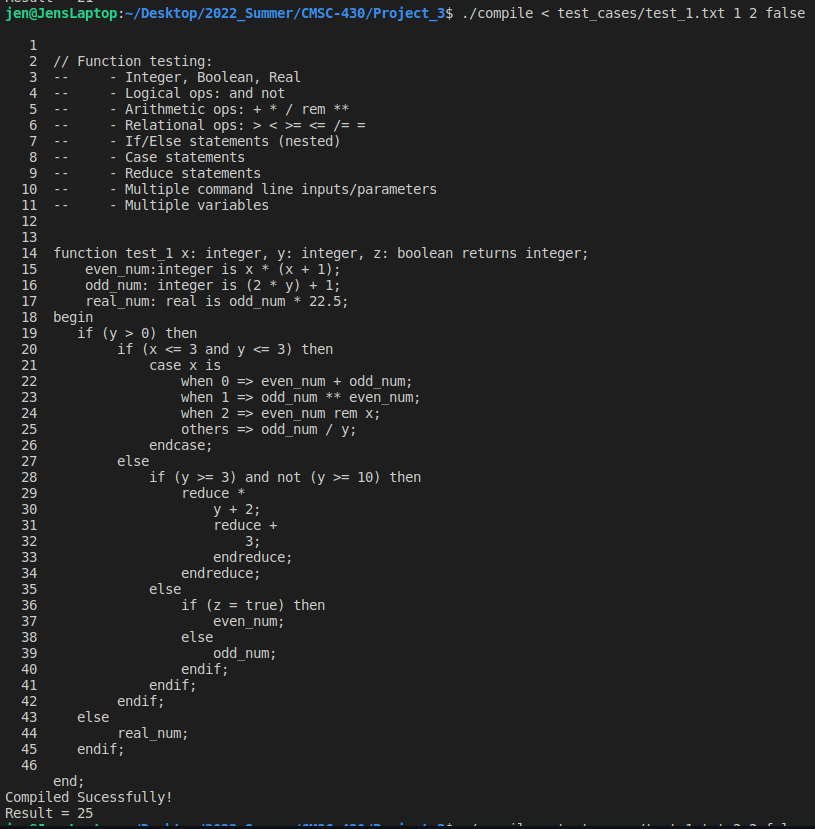
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Num.** | **x** | **y** | **z** | **Result** |
| Test 1 | 0 | 3 | false | 7 |
| Test 2 | 1 | 2 | false | 25 |
| Test 3 | 2 | 2 | false | 0 |
| Test 4 | 3 | 1 | false | 3 |
| Test 5 | 4 | 4 | false | 18 |
| Test 6 | 5 | 11 | true | 30 |
| Test 7 | 5 | 11 | false | 23 |
| Test 8 | 6 | 0 | true | 22.5 |

Screen Shots for Test Case 1:

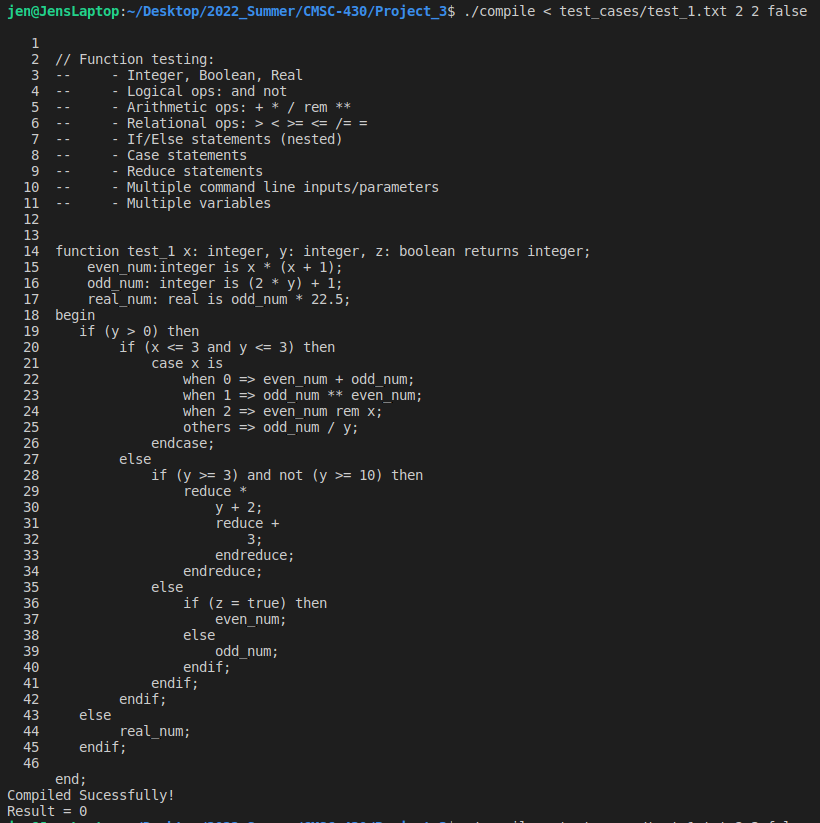
* Test 1



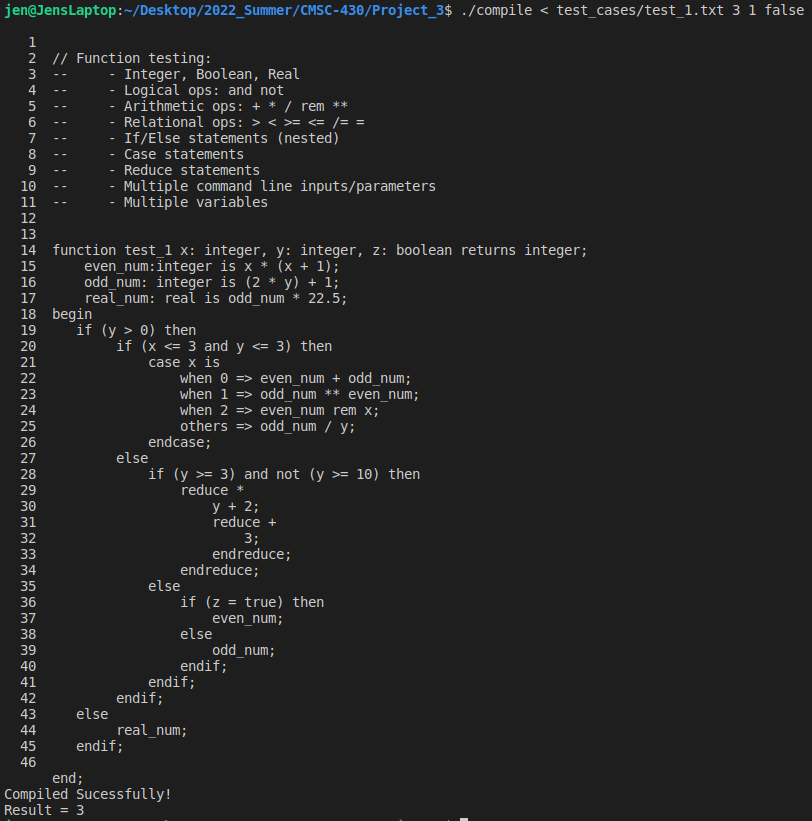
* Test 2



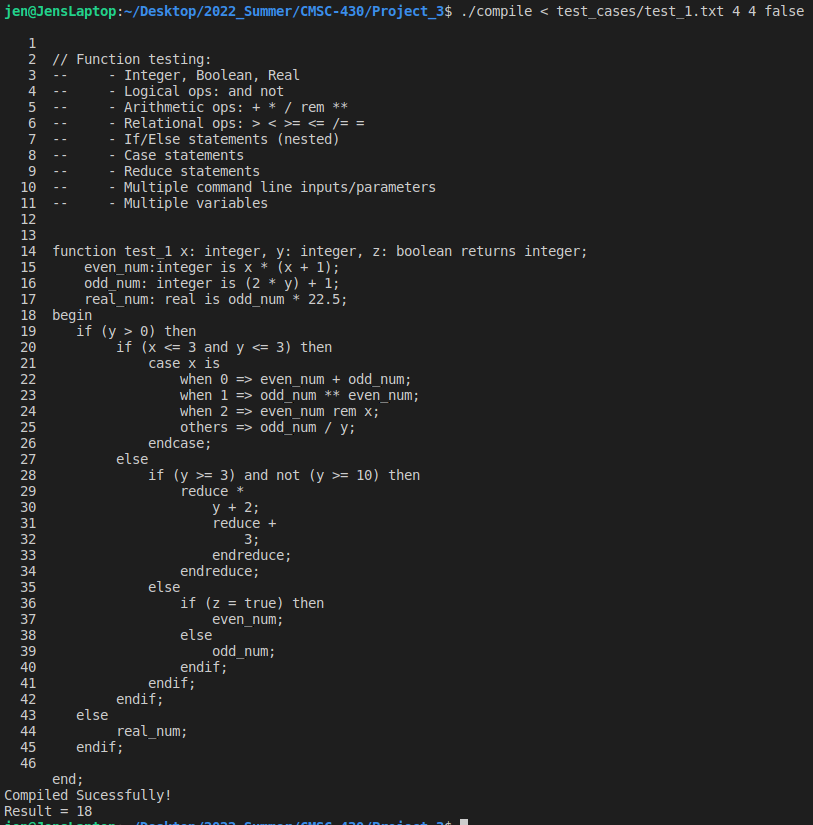
* Test 3



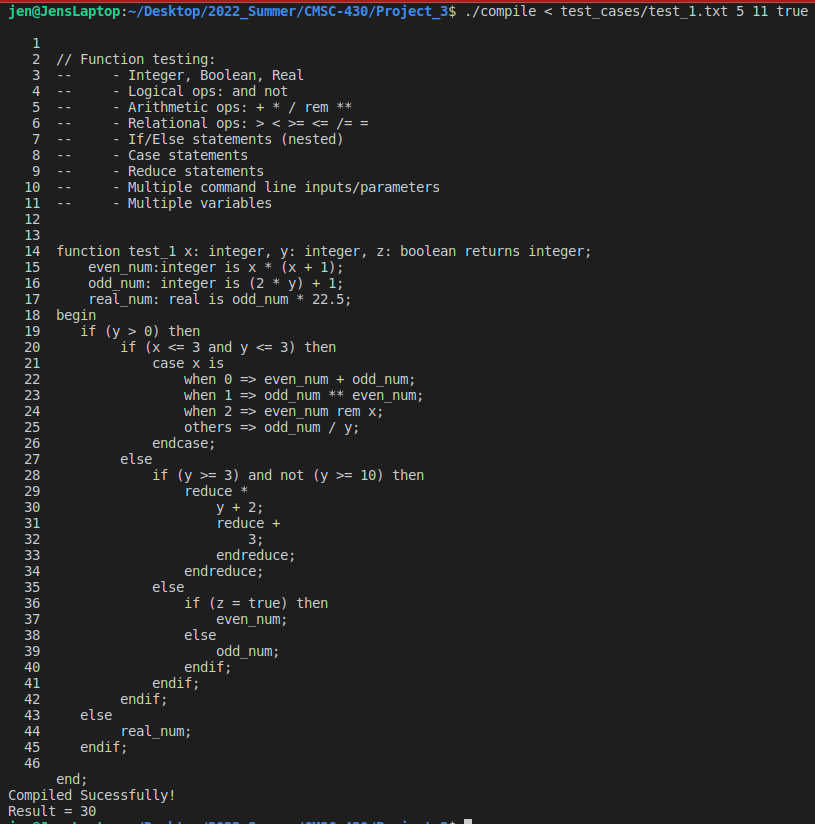
* Test 4



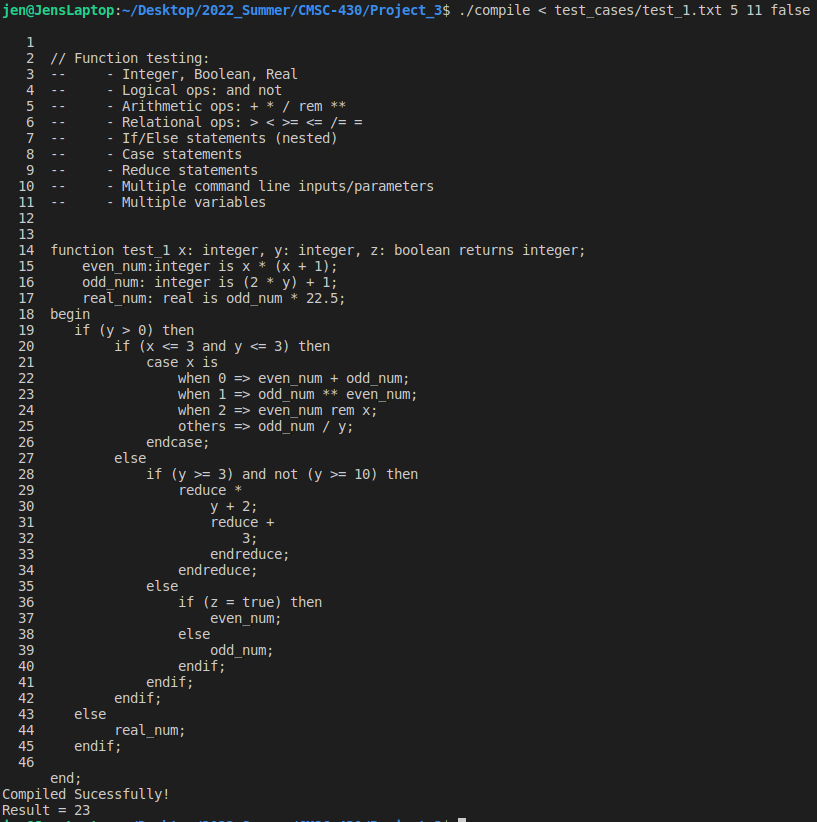
* Test 5



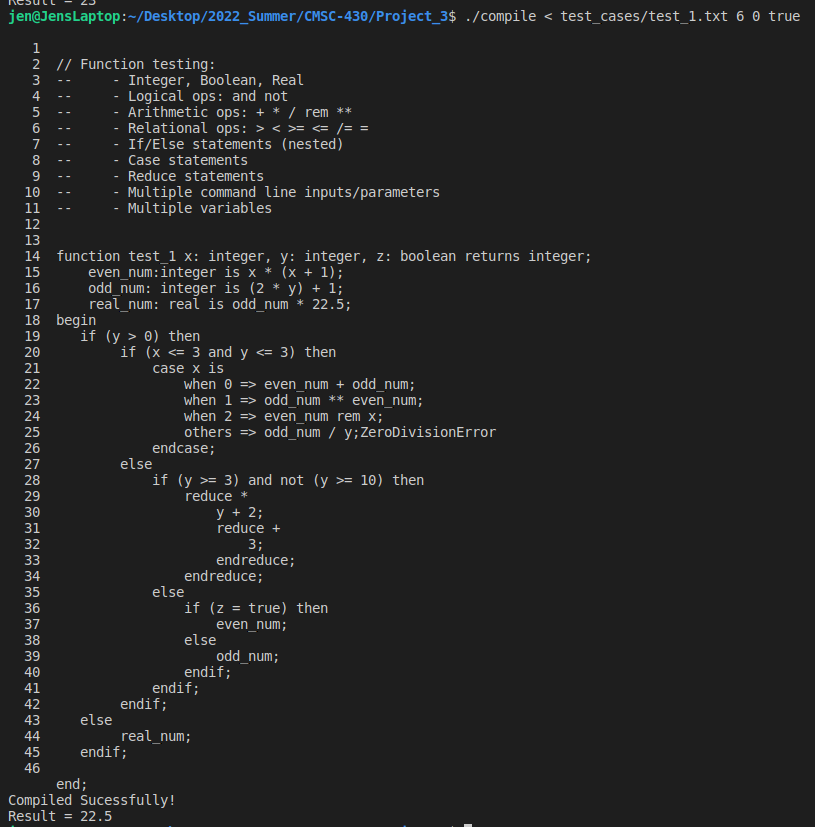
* Test 6



* Test 7



* Test 8



Test Case 2:

Testing for this test case will cover:

* Integer, Boolean
* Logical ops: and
* Relational ops: > <
* If/Else statements
* Case statements
* Command line inputs/parameters
* Multiple variables

|  |  |  |
| --- | --- | --- |
| **Test Num.** | **a** | **Result** |
| Test 1 | 0 | 0 |
| Test 2 | 1 | 0 |
| Test 3 | 2 | 1 |
| Test 4 | 3 | 0 |
| Test 5 | 4 | 0 |
| Test 6 | 5 | 0 |

Test Case 2 Screen Shots:

* + Test 1
  + Test 2
  + Test 3
  + Test 4
  + Test 5
  + Test 6

Test Case 3:

Testing for this test case will cover:

* Integer, Boolean, Real
* Logical ops: and not or
* Arithmetic ops: + - \* / rem \*\*
* Relational ops: > < >= <= /= =
* If/Else statements (nested)
* Case statements
* Reduce statements
* Multiple command line inputs/parameters
* Multiple variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Num.** | **a** | **b** | **c** | **t\_f** | **Result** |
| Test 1 | 5 | 10 | 18 | false | 9.3 |
| Test 2 | 20 | 0 | 0 | true | 7.5 |
| Test 3 | 0 | 3 | 7 | False | -27 |
| Test 4 | 5 | 0 | 50 | True | 5 |
| Test 5 | 5 | 0 | 50 | False | 1 |

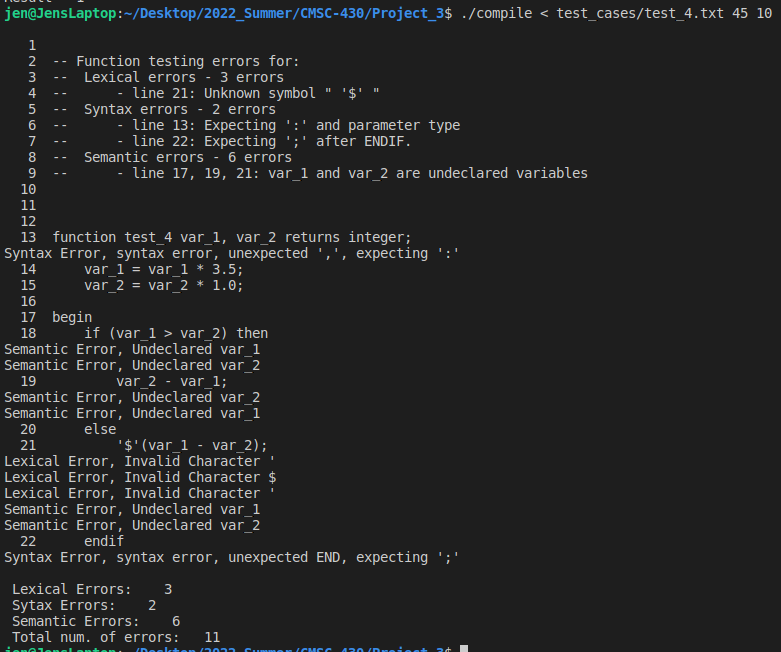
Test Case 3 Screen Shots:

* + Test 1
  + Test 2
  + Test 3
  + Test 4
  + Test 5

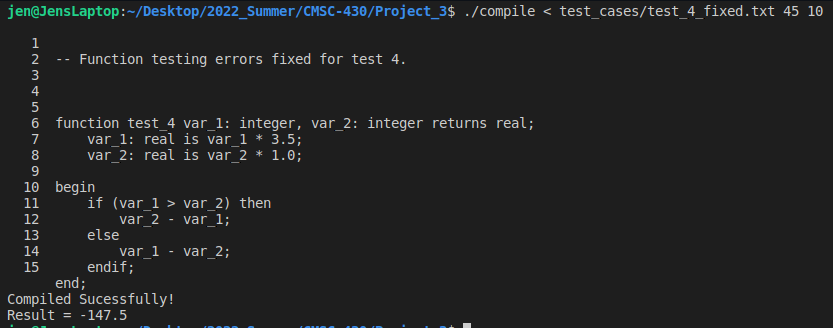
Test Case 4:

Testing for this test case will test errors for:

* Lexical errors - 3 errors
* line 21: Unknown symbol " '$' "
* Syntax errors - 2 errors
* line 13: Expecting ':' and parameter type
* line 22: Expecting ';' after ENDIF.
* Semantic errors - 6 errors
* line 17, 19, 21: var\_1 and var\_2 are undeclared variables



Test Case 4 (fixed errors)



**Lessons Learned**