

Table of Contents

Testing Strategy.....	1
SPUT Test Results	1
Turtle Program Output	9

Testing Strategy

I tried to test functions at the unit level, i.e. testing some of the most important individual functions against expected output, and at the integration level where I created a temporary file with an actual turtle script and tested parsing all combinations of legal and illegal commands.

I used the Sput test harness to write all my automated tests. Sput test functions are all contained in a single header file, Sput.h, which has been included in the source files of the program. The tests can be run at any time by running the program with TESTING as its first argument instead of the input file. This switches off SDL output and sends the Sput test results to stderr. SDL output can also be switched off at any time with a normal input file by passing SDL_OFF as the program's second argument.

SPUT Test Results

See below for the Sput test results (The number of tests run and passed is shown at the end):

== Entering suite #1, "Test parsing FD command with valid and invalid combinations" ==

```
[1:1] testParseFD:#1 "Testing FD command with valid
number 20, should return 0" pass
[1:2] testParseFD:#2 "Testing FD command with valid
number 20.5, should return 0" pass
[1:3] testParseFD:#3 "Testing FD command with valid
number -20, should return 0" pass
[1:4] testParseFD:#4 "Testing FD command with valid
number -20.5, should return 0" pass
[1:5] testParseFD:#5 "Testing FD command with valid
number 0, should return 0" pass
[1:6] testParseFD:#6 "Testing FD command with valid
variable A, should return 0" pass
[1:7] testParseFD:#7 "Testing FD command with valid
variable B, should return 0" pass
```

Error: FD expected a number or valid variable

[1:8] testParseFD:#8 "Testing FD command with invalid variable C, should return -1" pass

Error: FD expected a number or valid variable

[1:9] testParseFD:#9 "Testing FD command with invalid variable D, should return -1" pass

--> 9 check(s), 9 ok, 0 failed (0.00%)

== Entering suite #2, "Test parsing RT command with valid and invalid combinations" ==

[2:1] testParseRT:#1 "Testing RT command with valid number 20, should return 0" pass

[2:2] testParseRT:#2 "Testing RT command with valid number 20.5, should return 0" pass

[2:3] testParseRT:#3 "Testing RT command with valid number -20, should return 0" pass

[2:4] testParseRT:#4 "Testing RT command with valid number -20.5, should return 0" pass

[2:5] testParseRT:#5 "Testing RT command with valid number 0, should return 0" pass

[2:6] testParseRT:#6 "Testing RT command with valid variable A, should return 0" pass

[2:7] testParseRT:#7 "Testing RT command with valid variable B, should return 0" pass

Error: RT expected a number or valid variable

[2:8] testParseRT:#8 "Testing RT command with invalid variable C, should return -1" pass

Error: RT expected a number or valid variable

[2:9] testParseRT:#9 "Testing RT command with invalid variable D, should return -1" pass

--> 9 check(s), 9 ok, 0 failed (0.00%)

== Entering suite #3, "Test parsing LT command with valid and invalid combinations" ==

[3:1] testParseLT:#1 "Testing LT command with valid number 20, should return 0" pass

[3:2] testParseLT:#2 "Testing LT command with valid number 20.5, should return 0" pass

[3:3] testParseLT:#3 "Testing LT command with valid number -20, should return 0" pass

[3:4] testParseLT:#4 "Testing LT command with valid number -20.5, should return 0" pass

[3:5] testParseLT:#5 "Testing LT command with valid

number 0, should return 0" pass
[3:6] testParseLT:#6 "Testing LT command with valid
variable A, should return 0" pass
[3:7] testParseLT:#7 "Testing LT command with valid
variable B, should return 0" pass

Error: LT expected a number or valid variable

[3:8] testParseLT:#8 "Testing LT command with invalid
variable A, should return -1" pass

Error: LT expected a number or valid variable

[3:9] testParseLT:#9 "Testing LT command with invalid
variable Z, should return -1" pass

--> 9 check(s), 9 ok, 0 failed (0.00%)

== Entering suite #4, "Test executing RT Command" ==

[4:1] testExecuteRT:#1 "Testing turning right 40
degrees, turtle angle should be 40" pass
[4:2] testExecuteRT:#2 "Testing turning right 360
degrees, turtle angle should still be 40" pass
[4:3] testExecuteRT:#3 "Testing turning right 720
degrees, turtle angle should still be 40" pass
[4:4] testExecuteRT:#4 "Testing turning right 1080
degrees, turtle angle should still be 40" pass
[4:5] testExecuteRT:#5 "Testing turning right -20
degrees, turtle angle should still be 20" pass
[4:6] testExecuteRT:#6 "Testing turning right -360
degrees, turtle angle should still be 20" pass
[4:7] testExecuteRT:#7 "Testing turning right -720
degrees, turtle angle should still be 20" pass
[4:8] testExecuteRT:#8 "Testing turning right -1080
degrees, turtle angle should still be 20" pass

--> 8 check(s), 8 ok, 0 failed (0.00%)

== Entering suite #5, "Test executing LT Command" ==

[5:1] testExecuteLT:#1 "Testing turning left 40
degrees, turtle angle should be 320" pass
[5:2] testExecuteLT:#2 "Testing turning left 360
degrees, turtle angle should still be 320" pass
[5:3] testExecuteLT:#3 "Testing turning left 720
degrees, turtle angle should still be 320" pass
[5:4] testExecuteLT:#4 "Testing turning left 1080
degrees, turtle angle should still be 320" pass
[5:5] testExecuteLT:#5 "Testing turning left -20
degrees, turtle angle should now be 340" pass
[5:6] testExecuteLT:#6 "Testing turning left -360
degrees, turtle angle should still be 340" pass
[5:7] testExecuteLT:#7 "Testing turning left -720

degrees, turtle angle should still be 340" pass
[5:8] testExecuteLT:#8 "Testing turning left -1080
degrees, turtle angle should still be 340" pass

--> 8 check(s), 8 ok, 0 failed (0.00%)

== Entering suite #6, "Test Illegal Commands" ==

Error: Turtle file appears to be empty

[6:1] testParserIllegalCommands:#1 "Testing empty
turtle file, should return error" pass

Error: Expected final brace

[6:2] testParserIllegalCommands:#2 "Testing missing
final brace, should return error" pass

Error: Expected initial brace

[6:3] testParserIllegalCommands:#3 "Testing missing
initial brace, should return error" pass

Error: FD expected a number or valid variable

[6:4] testParserIllegalCommands:#4 "Testing FD
command with no argument, should return error" pass

Error: FD expected a number or valid variable

[6:5] testParserIllegalCommands:#5 "Testing FD
command with invalid argument, should return error"
pass

Error: LT expected a number or valid variable

[6:6] testParserIllegalCommands:#6 "Testing LT
command with no argument, should return error" pass

Error: LT expected a number or valid variable

[6:7] testParserIllegalCommands:#7 "Testing LT
command with invalid argument, should return error"
pass

Error: RT expected a number or valid variable

[6:8] testParserIllegalCommands:#8 "Testing RT
command with no argument, should return error" pass

Error: RT expected a number or valid variable

[6:9] testParserIllegalCommands:#9 "Testing RT

command with no argument, should return error" pass

Error: SET expected a valid variable name, i.e. an uppercase letter between A-Z

[6:10] testParserIllegalCommands:#10 "Testing SET command with no arguments, should return error" pass

Error: SET expected a valid variable name, i.e. an uppercase letter between A-Z

[6:11] testParserIllegalCommands:#11 "Testing SET command with invalid variable, should return error" pass

Error: SET expected an assignment "!=" operator

[6:12] testParserIllegalCommands:#12 "Testing SET command with valid variable but no assignment operator, should return error" pass

Error: SET expected a number or valid variable after the assignment "!=" operator

[6:13] testParserIllegalCommands:#13 "Testing SET command with valid variable and assignment operator but nothing after it, should return error" pass

Error: SET expected a number or valid variable after the assignment "!=" operator

[6:14] testParserIllegalCommands:#14 "Testing SET command with valid variable and assignment operator, but invalid number or variable after it, should return error" pass

Error: SET expected a semi-colon ";" after the RPN expression

[6:15] testParserIllegalCommands:#15 "Testing SET command with valid variable, assignment operator and number, but missing semi-colon, should return error" pass

Error: SET expected a valid RPN expression. Please check expression for errors

[6:16] testParserIllegalCommands:#16 "Testing SET command with valid variable, assignment operator and number and missing semi-colon, but illegal RPN expression, should return error" pass

Error: SET expected a valid RPN expression. Please check expression for errors

[6:17] testParserIllegalCommands:#17 "Testing SET command with another illegal RPN expression, should return error" pass

Error: SET expected a valid RPN expression. Please check expression for errors

[6:18] testParserIllegalCommands:#18 "Testing SET command with invalid variables in RPN expression, should return error" pass

Error: D0 expected a valid variable name, i.e. an uppercase letter between A-Z

[6:19] testParserIllegalCommands:#19 "Testing D0 command with no arguments, should return error" pass

Error: D0 expected a valid variable name, i.e. an uppercase letter between A-Z

[6:20] testParserIllegalCommands:#20 "Testing D0 command with invalid variable, should return error" pass

Error: D0 expected the FROM keyword after variable

[6:21] testParserIllegalCommands:#21 "Testing D0 command with valid variable but nothing after, should return error" pass

Error: D0 expected a valid integer after FROM keyword

[6:22] testParserIllegalCommands:#22 "Testing D0 command with valid variable and FROM keyword, but nothing after, should return error" pass

Error: D0 expected a valid integer after FROM keyword

[6:23] testParserIllegalCommands:#23 "Testing D0 command with valid variable and FROM keyword, but unrecognized symbol after, should return error" pass

Error: D0 expected a valid integer after FROM keyword

[6:24] testParserIllegalCommands:#24 "Testing D0 command with valid variable and FROM keyword, but invalid number after (i.e. non-integer), should return error" pass

Error: D0 expected the T0 keyword after first integer

[6:25] testParserIllegalCommands:#25 "Testing D0 command with valid variable, FROM keyword and valid

integer but nothing after, should return error" pass

Error: D0 expected the T0 keyword after first integer

[6:26] testParserIllegalCommands:#26 "Testing D0 command with valid variable, FROM keyword, valid integer and T0 keyword but nothing after, should return error" pass

Error: D0 expected a valid integer after T0 keyword

[6:27] testParserIllegalCommands:#27 "Testing D0 command with valid variable, FROM keyword, valid integer and T0 keyword, but unrecognized command after, should return error" pass

Error: D0 expected a valid integer after T0 keyword

[6:28] testParserIllegalCommands:#28 "Testing D0 command with valid variable, FROM keyword, integer and T0 keyword, but invalid integer after, should return error" pass

Error: Expected initial brace

[6:29] testParserIllegalCommands:#29 "Testing D0 command with valid variable, FROM keyword, valid integer, T0 keyword, valid integer, but missing initial brace, should return error" pass

Error: Expected final brace

[6:30] testParserIllegalCommands:#30 "Testing D0 command with valid variable, FROM keyword, valid integer, T0 keyword, valid integer and initial brace, but no final brace, should return error" pass

Error: Unrecognized command or symbol encountered

[6:31] testParserIllegalCommands:#31 "Testing valid do loop with unrecognized command, should return error" pass

Error: FD expected a number or valid variable

[6:32] testParserIllegalCommands:#32 "Testing valid do loop with recognized command but no argument, should return error" pass

Error: FD expected a number or valid variable

[6:33] testParserIllegalCommands:#33 "Testing valid do loop with recognized command but unrecognized argument, should return error" pass

Error: Expected initial brace

[6:34] testParserIllegalCommands:#34 "Testing invalid do loop (missing initial brace) with legal command, should return error" pass

Error: Expected final brace

[6:35] testParserIllegalCommands:#35 "Testing invalid do loop (missing final brace) with legal command, should return error" pass

--> 35 check(s), 35 ok, 0 failed (0.00%)

== Entering suite #7, "Test Legal Commands" ==

[7:1] testParserLegalCommands:#1 "Testing empty pair of braces, script still legal so should not return error" pass

[7:2] testParserLegalCommands:#2 "Testing FD command with valid number, should not return error" pass

[7:3] testParserLegalCommands:#3 "Testing FD command with valid variable, should not return error" pass

[7:4] testParserLegalCommands:#4 "Testing RT command with valid number, should not return error" pass

[7:5] testParserLegalCommands:#5 "Testing RT command with valid variable, should not return error" pass

[7:6] testParserLegalCommands:#6 "Testing LT command with valid number, should not return error" pass

[7:7] testParserLegalCommands:#7 "Testing LT command with valid variable, should not return error" pass

[7:8] testParserLegalCommands:#8 "Testing SET command with valid number, should not return error" pass

[7:9] testParserLegalCommands:#9 "Testing SET command with valid variable after assignment operator instead of number, should not return error" pass

[7:10] testParserLegalCommands:#10 "Testing SET command with valid RPN expression, should not return error" pass

[7:11] testParserLegalCommands:#11 "Testing SET command with another valid RPN expression, should not return error" pass

[7:12] testParserLegalCommands:#12 "Testing SET command with another valid RPN expression, should not return error" pass

[7:13] testParserLegalCommands:#13 "Testing SET command with another valid RPN expression using variable instead of number, should not return error" pass

[7:14] testParserLegalCommands:#14 "Testing D0 command with empty loop body, should not return error as still legal" pass

[7:15] testParserLegalCommands:#15 "Testing D0


```

command with FD command in loop body using number,
should not return error as still legal" pass
[7:16] testParserLegalCommands:#16 "Testing D0
command with FD command in loop body using loop
variable, should not return error" pass
[7:17] testParserLegalCommands:#17 "Testing D0
command with FD command in loop body using variable
assigned before loop, should not return error" pass
[7:18] testParserLegalCommands:#18 "Testing D0
command with FD command in loop body using variable
assigned inside loop body, should not return error"
pass

```

```

--> 18 check(s), 18 ok, 0 failed (0.00%)

```

```

==> 96 check(s) in 7 suite(s) finished after 0.00
second(s),
    96 succeeded, 0 failed (0.00%)

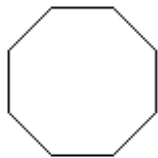
```

[SUCCESS]

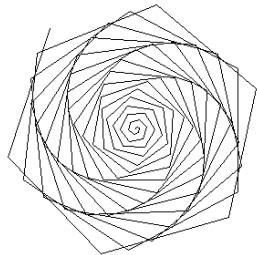
Program ended with exit code: 0

Turtle Program Output

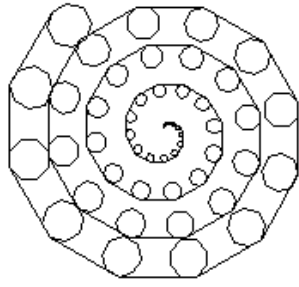
1. { DO A FROM 1 TO 8 { FD 30 LT 45 } }



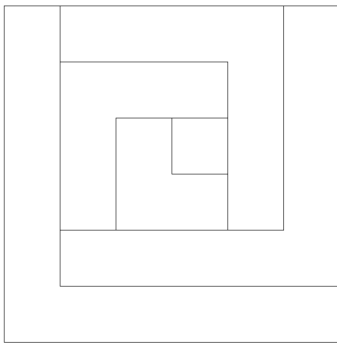
2. { DO A FROM 1 TO 100 { SET C := A 1.5 * ; FD C RT 62 } }



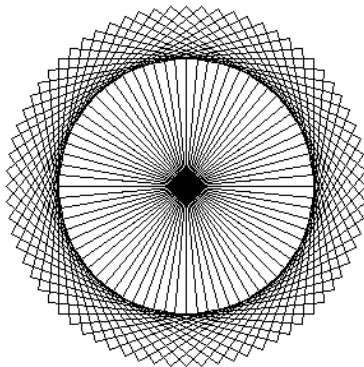
3. { DO A FROM 1 TO 50 { FD A RT 30 DO B FROM 1 TO 8 { SET C := A 5 / ; FD C RT 45 } } }



4. { DO A FROM 1 TO 7 { SET B := A 100 * ; FD B RT 90 DO C FROM 1 TO 4 { FD B RT 90 } } }



5. { DO A FROM 1 TO 100 { DO B FROM 1 TO 4 { FD 100 RT 90 } RT 5 } }



6. { RT 90 DO A FROM 1 TO 50 { SET B := A 10 * ; FD B RT 144 } }

