

In this assignment, you will implement a simple programming language, we call, Flang for Float Language, that can perform simple calculations and display the results. Your parser will generate valid Java assembly code which can be linked to executable Java byte code using the Jasmin tool we introduced in class.

Part I: Design and parsing of the language.

```

prog
  : (statement NEWLINE)+ EOF
  ;
statement
  : 'let' ID 'be' expr
  | 'print' expr (COUNT times)?
  ;
expr
  : FLOAT
  | ID
  | '(' OP expr* ')'
  ;
OP
  : '+' | '-' | '*' | '/'
  ;
ID
  : LETTER (LETTER|'0'..'9')*
  ;
fragment LETTER
  : '$'
  | 'A' .. 'Z'
  | 'a' .. 'z'
  | '-'
  | '.'
  ;
FLOAT
  : ('0'..'9')+ '.' ('0'..'9')*
  ;
COUNT
  : ('0'..'9')
  ;

```

The semantics are to support full floating point arithmetics with trigonometric functions as well. Functions are called using prefix notation. Here are some examples of **expr**.

3.1415

// a constant, pi

```

(* 2.0 3.1415)           // twice of pi
(* pi (* 2.0 radius))    // perimeter of a circle
(* pi (* radius radius)) // area of a circle
(/ total count)         // average
(* total (/ 1.0 count))  // also the average
(+ 1.0 2.0 3.0 4.0 5.0) // + and * take all parameters
(* pi radius radius)     // also the area

```

Part II: Relevant Java instructions in assembly code

- fadd: adds two floating values on the stack, and replaces them with the result.
- fsub: subtracts the top value of the stack from the second top value of the stack, and replaces them with the result.
- fmul: multiplies the two floating values on the stack, and replaces them with the result.
- fdiv: divides the second top value with the top value, and replaces them with the result.

In addition, you need to use the standard assembly code fragments for invocation of `System.out.println(float)`, and looping.

Part III: Test flang code

```

let pi be 3.1415
let radius be 5.0
let diameter be (* 2.0 radius)
let area be (* pi radius radius)
let number-of-brothers 3.0
print area 2 times
print number-of-brothers
print (/ area number-of-brothers)

```

Here is another test flang code

```

let annual.income be 100000.0
let months.in.a.year be 12.0
print (/ annual.income months.in.a.year) 1 times

```

Part IV: Submission and automatic grading

This assignment is to be automarked completely.

You must submit your code with the file names as indicated.

1. Flang.g
2. Compiler.java

Your code will be compiled **automatically** by the following script.

```
export CLASSPATH=<<system wide antlr library>>:.
```

```
java org.antlr.Tool Flang.g
javac Compiler.java
```

Compiler.class is an executable Java class that accepts **one** parameter from command line, which is the Flang source code. The compiler **always** generates an assembly file called **a.j** regardless the Flang source code file name. The Java **executable** class that is described by **a.j** is **always** named **A**.

Your submission will be **automatically** executed by the following script:

```
export CLASSPATH=<<antlr library>>:.
java Compiler test.flang
java -jar jasmin.jar a.j
java A >> output 2>&1
diff output expected
```

Marking scheme:

If passes the marking scripts: 100%
Otherwise, the following break down applies:

Flang.g	30%
Compiler	30%
Correctness	40%