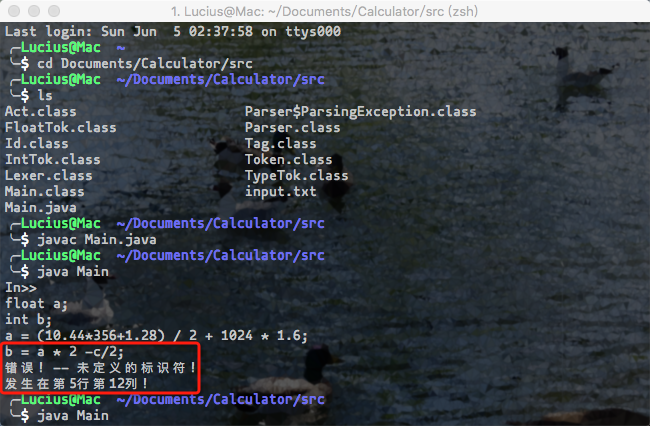
文法：

/\*\*  
 \* <text> --> <stmtList> .  
 \* <stmtList>--> <stmt>(； <stmt>)\*  
 \* <stmt> --> <declare> | <output> | <calc>  
 \* <declare> --> TYPE <varList>  
 \* <varList> --> ID(，ID)\*  
 \* <output> --> write (<calc>)  
 \* <calc> --> ID = <calc> | <expr>  
 \* <expr> --> <term> <expr\_>  
 \* <expr\_> --> (+|-) <term> <expr\_> | ε  
 \* <term> --> <factor> <term\_>  
 \* <term\_> --> (\*|/) <factor> <term\_> | ε  
 \* <factor> --> —<factor> | ID | INT | FLOAT | (<calc>)  
 \*\*/

测试：

（1）：



输入为：

float a;

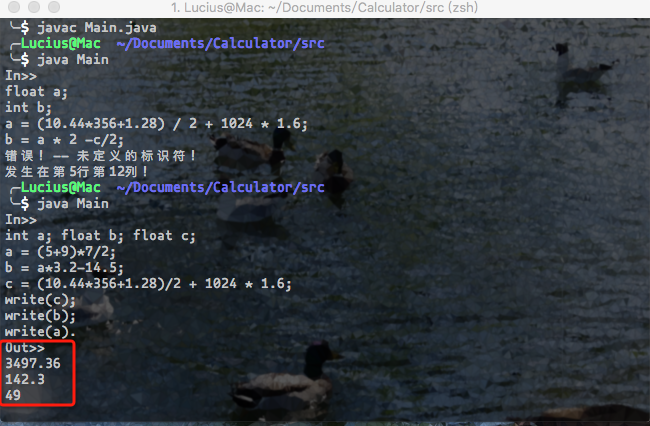
int b;

a = (10.44\*356+1.28) / 2 + 1025 \*1.6;

b = a\*2 – c/2;

此时立即报错，因为 c 没有声明，编译器准确给出了 c 的位置。

（2）：



输入：

int a; float b; float c;

a = (5+9)\*7/2;

b = a\*3.2-14.5;

c = (10.44\*356+1.28) / 2 + 1025 \*1.6;

write(c);

write(b);

write(a).

输出：

3497.36

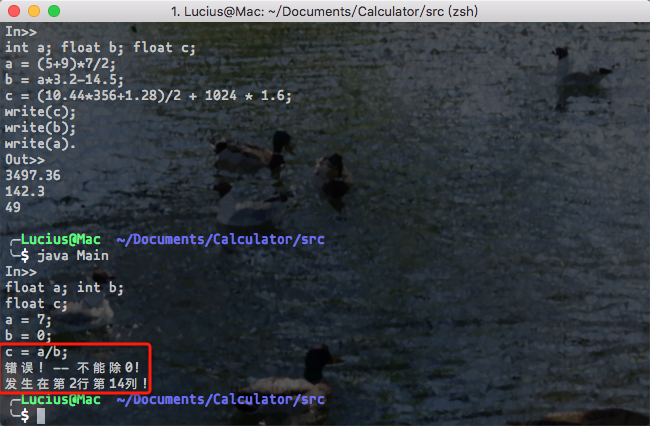
142.3

49

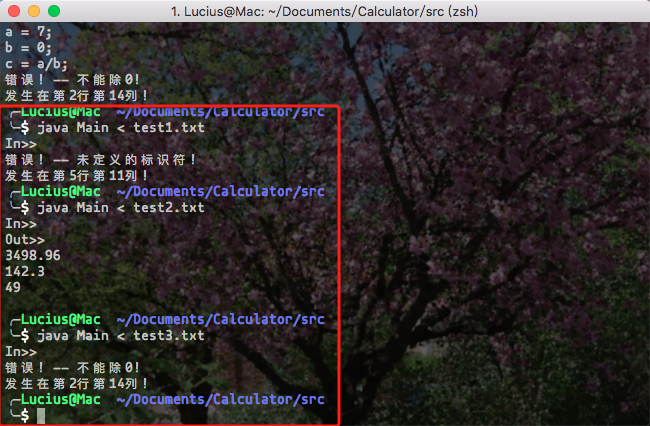
经验证，确实分别是 c, b, a 的值。

（3）：

除零错误：



以上三例都已经在 txt 文件中写好，放在 src/文件夹中。可以手动输入，也可以采用 编译后java Main < xxx.txt 的重定向输入方式。



基本上所有的报错错误都有涵盖。

值得一提的是，作业要求中的“浮点数可以转换成整数；整数不可以转换成浮点数”这一项无法做到，我认为这句话说反了。所以只能做到整数转浮点数（如 7 -> 7.0），而不能浮点数转整数（精度损失大）。