2017-2018学年第1学期

**实 验 报 告**



* 课程名称:编程语言原理与编译
* 实验项目:期末大作业
* 专业班级：计算1501
* 学生学号：31501081，31501079
* 学生姓名：许佳晨，施怡昕
* 实验指导教师: 郭鸣

[简介](#_Toc22308)

[语言教程](#_Toc22308)

[ 算数运算符](#_Toc20160)

[ 循环 5](#_Toc20160)

[语言手册 6](#_Toc10246)

[ 算数运算符 6](#_Toc20160)

[ 循环 7](#_Toc20160)

[项目计划](#_Toc10246) 8

[ 概述 8](#_Toc20160)

[ 计划安排表 10](#_Toc14374)

[架构设计 11](#_Toc9768)

[测试 12](#_Toc21686)

[测试用例 12](#_Toc14784)

[测试代码 13](#_Toc26931)

[测试结果](#_Toc27923)

[心得收获](#_Toc24828)

[ 课程体会与建议](#_Toc9238)

[项目体会 21](#_Toc26103)

[技术评价表 22](#_Toc1980)

[分工表 23](#_Toc4453)

[GITHUB提交 24](#_Toc4453)

简介

**背景**

**编译原理是计算机专业的一门重要专业课，旨在介绍编译程序构造的一般原理和基本方法。内容包括语言和文法、词法分析、语法分析、语法制导翻译、中间代码生成、存储管理、代码优化和目标代码生成。 编译原理是计算机专业设置的一门重要的专业课程。虽然只有少数人从事编译方面的工作，但是这门课在理论、技术、方法上都对学生提供了系统而有效的训练，有利于提高软件人员的素质和能力。因此本课程大作业就是使用F#语言来实现一个micro C语言。**

**目的与目标**

**我们组在这次大作业中主要完成的是：**

**1编译（编译并在内存中运行、编译保存并执行、编译保存执行并在控制台输出结果、检测语法错误）**

**2语法（SWITCH CASE判断结构、FOR循环结构、DO WHILE循环结构、?:表达式）**

**3词法（八进制、十六进制、++、--、+、-、\*、/、%）**

**4 语义（）**

**5测试（语法部分测试、词法部分测试、语义部分测试）**

6文档编写

语言教程

### 运算符

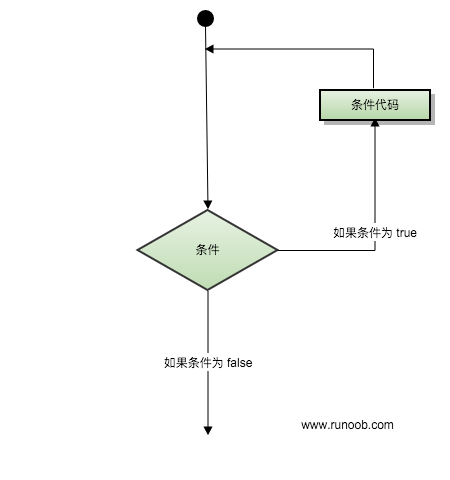
下表显示了支持的所有算术运算符。假设变量 **A** 的值为 10，变量 **B** 的值为 20，则：

|  |  |  |
| --- | --- | --- |
| 运算符 | 描述 | 实例 |
| + | 把两个操作数相加 | A + B 将得到 30 |
| - | 从第一个操作数中减去第二个操作数 | A - B 将得到 -10 |
| \* | 把两个操作数相乘 | A \* B 将得到 200 |
| / | 分子除以分母 | B / A 将得到 2 |
| % | 取模运算符，整除后的余数 | B % A 将得到 0 |
| ++ | 自增运算符，整数值增加 1 | A++ 将得到 11 |
| -- | 自减运算符，整数值减少 1 | A-- 将得到 9 |

### 循环

有的时候，我们可能需要多次执行同一块代码。一般情况下，语句是按顺序执行的：函数中的第一个语句先执行，接着是第二个语句，依此类推。

编程语言提供了更为复杂执行路径的多种控制结构。

循环语句允许我们多次执行一个语句或语句组，下面是大多数编程语言中循环语句的流程图：

## while 循环

while(condition)  
{  
 statement(s);  
}

在这里，**statement(s)** 可以是一个单独的语句，也可以是几个语句组成的代码块。**condition** 可以是任意的表达式，当为任意非零值时都为 true。当条件为 true 时执行循环。

当条件为 false 时，程序流将继续执行紧接着循环的下一条语句。

## 语言手册

### 算数运算符

Assumes that a=10 and b =20

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| + | Adds two operands | A + B will give 30 |
| - | Subtracts second operand from the first | A - B will give -10 |
| \* | Multiply both operands | A \* B will give 200 |
| / | Divide numerator by de-numerator | B / A will give 2 |
| % | Modulus Operator and remainder after integer division | B % A will give 0 |
| ++ | Increment operator increases integer value by one | A++ will give 11 |
| -- | Decrement operator decreases integer value by one | A-- will give 9 |

### 循环

|  |  |  |
| --- | --- | --- |
| Loop type | Description | Example |
| While循环 | 重复语句或语句组，直到给定的条件为真。测试前的状态执行循环体。 | int main(bool b) {  while (b)  break;  return 1;  }" |
| For 循环 | 执行的语句序列多次和缩写管理循环变量的代码。 | int n;  n=8;  int a;  for(a=0; a<n; ++a)  print a; |

## 

## 项目计划

### 概述

在一个月左右的时间里我们小组两人共同来完成大作业，大作业分为了很多小的模块，虽然有分工但是我们还是有彼此熟悉自己工作。虽然有一个月的时间，但是有很多时间是零散的，而且还有别的课程要完成作业，所以我们会利用我们平常的时间，而且能两个人一起的时间，这样不仅大家能一起讨论，增加团队间的交流也是很重要的，两个人一起完成能够提交效率，相互学习对方完成的部分，或者讨论自己或对方不懂的地方。

我们把这个项目分成了很多小模块，这样不仅能利用起来零碎的时间还可以每天都有一点进步，加快了完成作业的效率。本项目主要的模块有项目的设计、项目的编码和项目的测试，小组里的每个人都有参与了这三个部分，其中设计和编码阶段是比较难的部分。在设计的时候我们需要思考这些是不是我们有能力能够写出来的部分，或者这些是不是我们能在有限的时间里能够学习并且完成的地方，这就需要考虑到我们的设计的可行性，两个人列举出哪些地方是可行的，哪些又是我们需要舍弃的，当然这些东西是我们需要认真、严谨的研究过的。所以我们在设计阶段分配了大概2-3天的时间。后面我们决定加入一些语法和语义。

编码部分当然是其中重要的部分，所以花的时间比较长，编码的部分主要按照我们需要加的部分分为语法模块和语义模块，两个人分别完成一部分，我们在语法部分主要加入了FOR循环的结构DOWHILE循环的结构、SWITCH选择的结构等等。在语义的部分我们主要加入了类型检查，进行类型检查以后再去进行匹配。在编码的部分我们主要规划了2周左右的时间，但是没有顺利在预计时间里完成这个任务，因为这个作业对我们来说有难度，所以时间上超出了5天左右的时间。

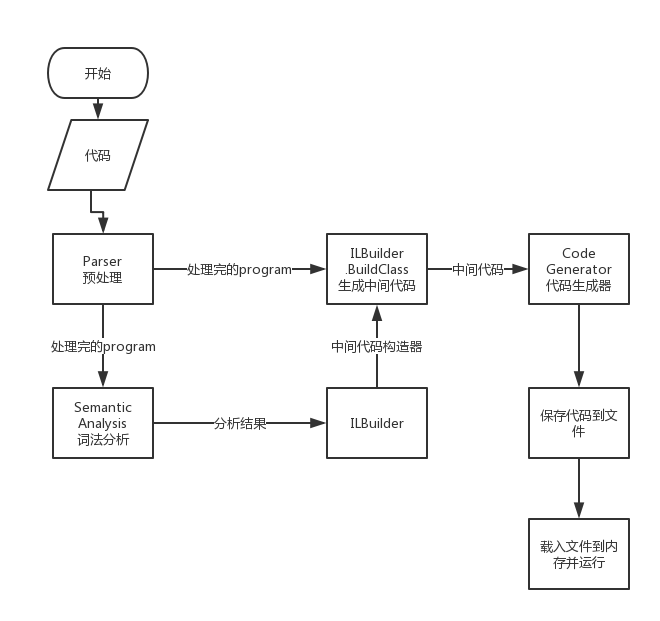
测试部分和最后的编写报告的部分是我们一起做的，测试的部分因为设计到编码的部分所以需要我们相互讨论得到，然后在进行册是，但是测试的部分不是特别的详细，我们只把一些完整的模块加入到其中，所以我们写的测试用例主要是SWITCH选择的结构、DOWHILE循环的结构、FOR循环的结构和FOR循环和SWITCH选择这两个相结合的结构，主要的测试报告后面还会写到。其中包含了测试结果来验证我们的编码部分。报告中完整的写到了我们所分的细小的模块、分工、测试等等。测试部分分配的时间为2天，我们也很抓紧时间在规定时间里完成了。编写报告的部分我们也顺利在3天时间里完成了。

虽然不是十分的顺利我们小组还是完成了编译原理的作业，这次完成的内容在整个课上的小组算不上比较好的，但是我们也是在很认真的去做自己的事情。我们小组两个的擅长的地方不同，但也是比较好的搭配，一个人会多在编码部分一点、一个人会报告中多加一点，基本来说还是较为平均的。在整个完成大作业的过程中要感谢老师和同学对我们的帮助，以及大家在较长时间里的努力。

### 计划安排表

|  |  |
| --- | --- |
| 项目计划 | 计划实施时间 |
| 设计模块 | 3天 |
| 编写语法分析代码 | 1周 |
| 编写语义部分代码 | 1周 |
| 编写词法部分代码 | 1周 |
| 测试代码 | 2天 |
| 修正代码 | 2天 |
| 书写报告 | 3天 |
| 大作业验收 | 1天 |

**架构设计**



## 测试

### 测试用例

|  |  |  |
| --- | --- | --- |
| 编号 | 测试项 | 测试用例 |
| 1 | 八进制 | test1.c |
| 2 | 十六进制 | test2.c |
| 3 | do while | test3.c |
| 4 | ?： | test4.c |
| 5 | switch case | test5.c |
| 6 | ++、--、+、-、\*、/、%、for、switch case | test6.c |

### 测试代码

test1.c

void main(int n){

n = 0o1000;

print n;

println;

}

test2.c

void main() {

n = 0x100;

print n;

println;

}

test3.c

void main(int a){

do{

print a;

--a;

}while(a>=0)

println ;

print a;

}

test4.c

void main(int n) {

int a;

for(a=0; a<n; ++a)

a%2==0?print a/2:print (a+2)/2;

println;

}

test5.c

void main(int n)

{

switch (n%3)

{

case 0: print 0;

case 1: print 1;

case 2: print 2;

}

println;

}

test6.c

int main(){

int a;

int b;

int i;

for (i = 0; i <= 6; ++i){

a = 10;

b = 6;

switch (i) {

case 0:

(a += b);

case 1:

(a -= b);

case 2:

(a \*= b);

case 3:

(a /= b);

case 4:

(a %= b);

case 5:

(++a);

case 6:

(--a);

}

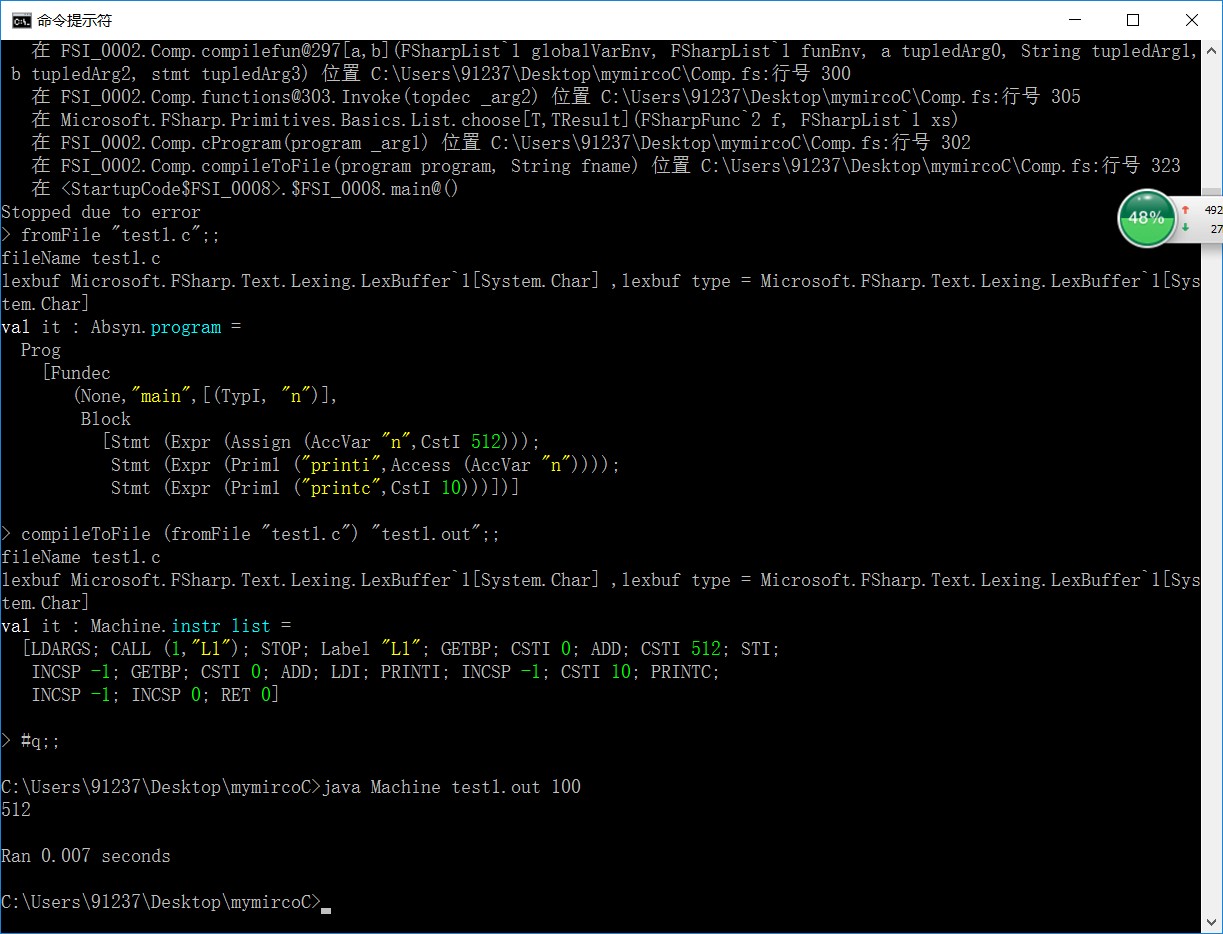
print a;

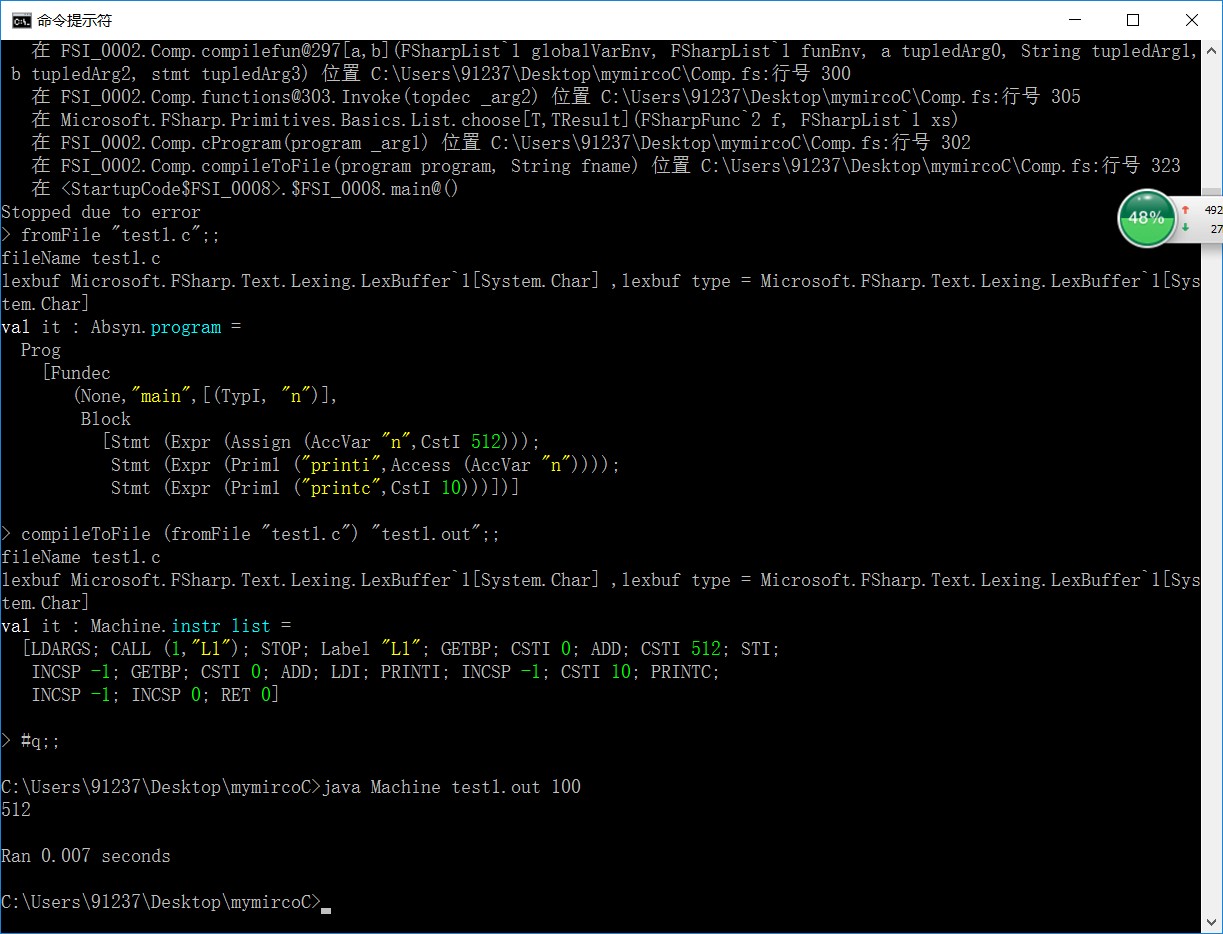
}

}

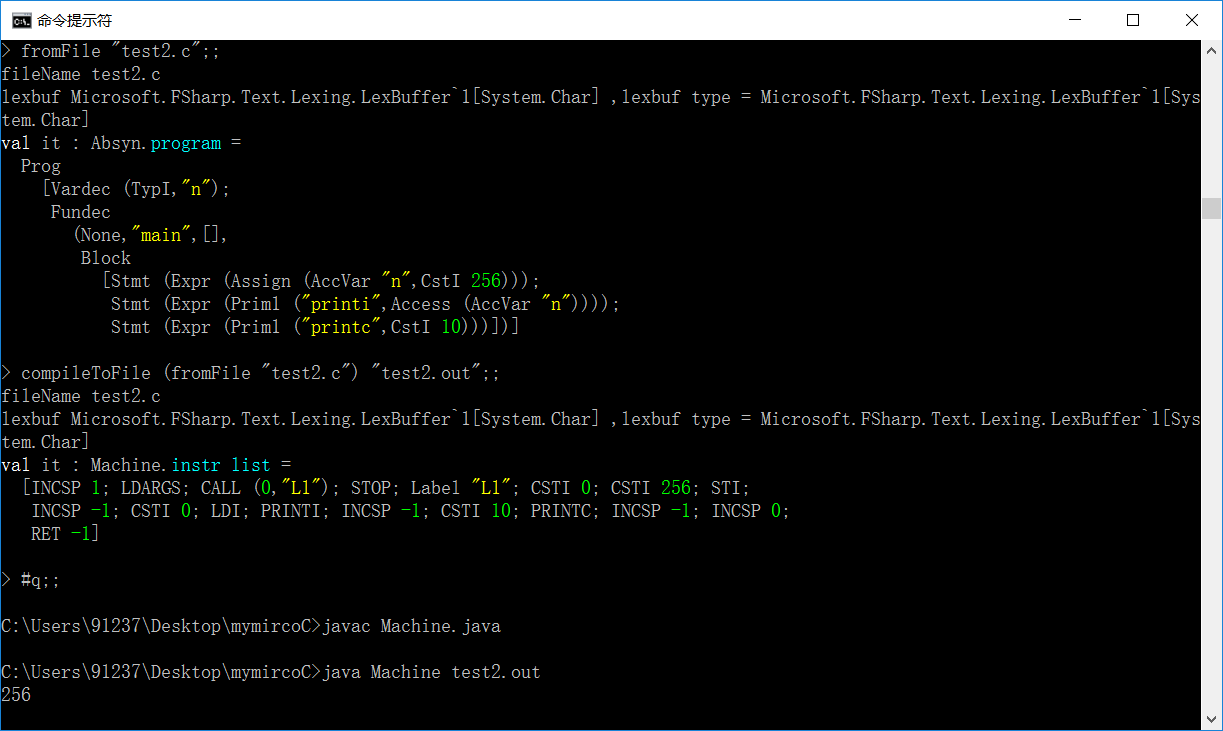
### 测试结果

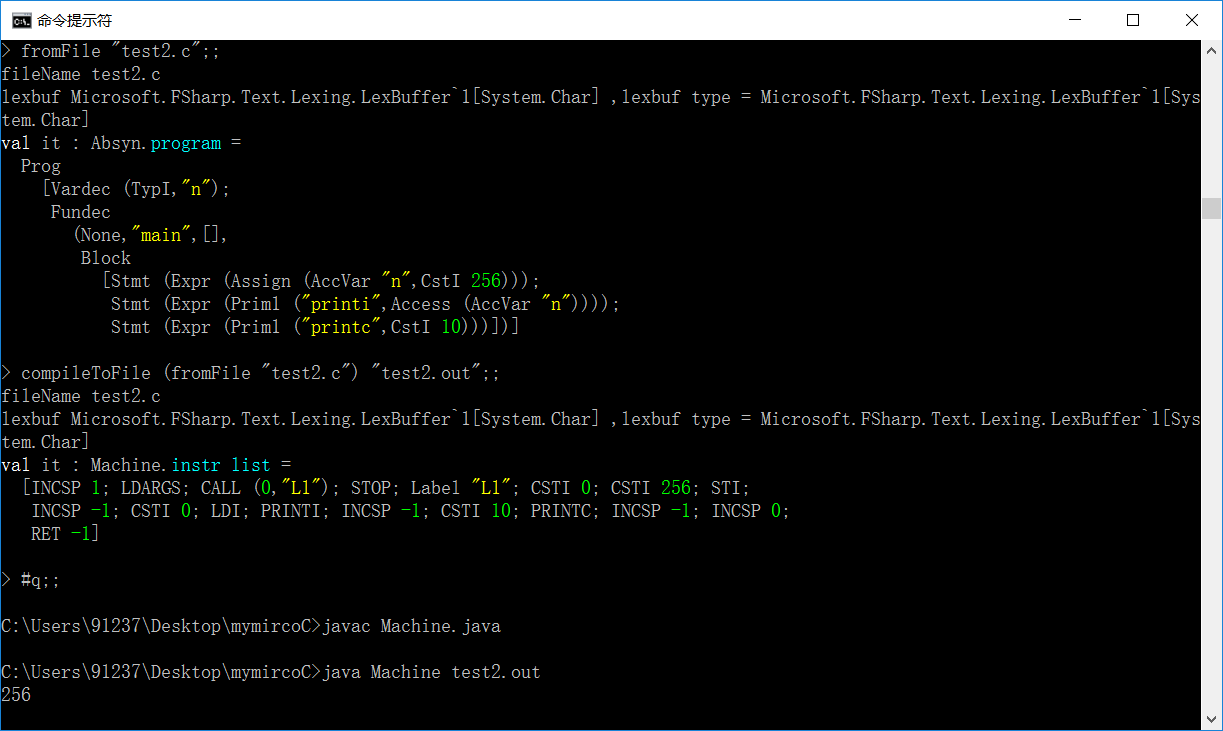
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 1 | 100 | 512 | 通过 |

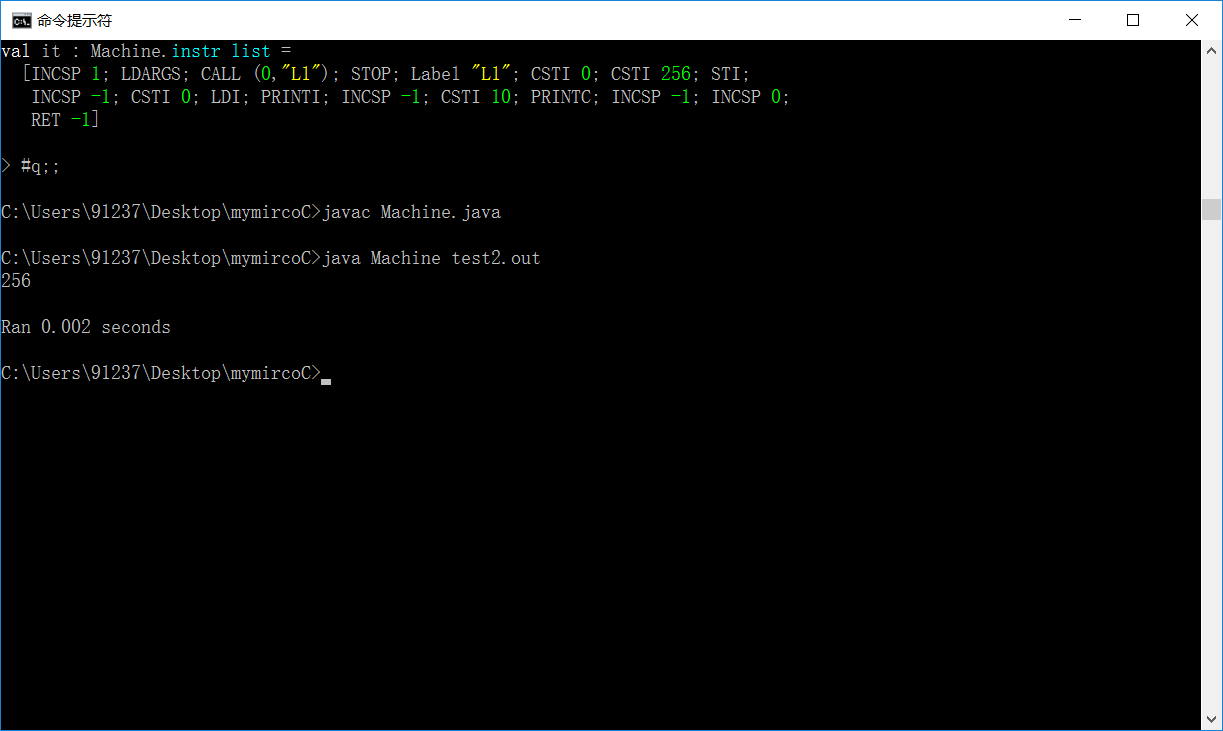




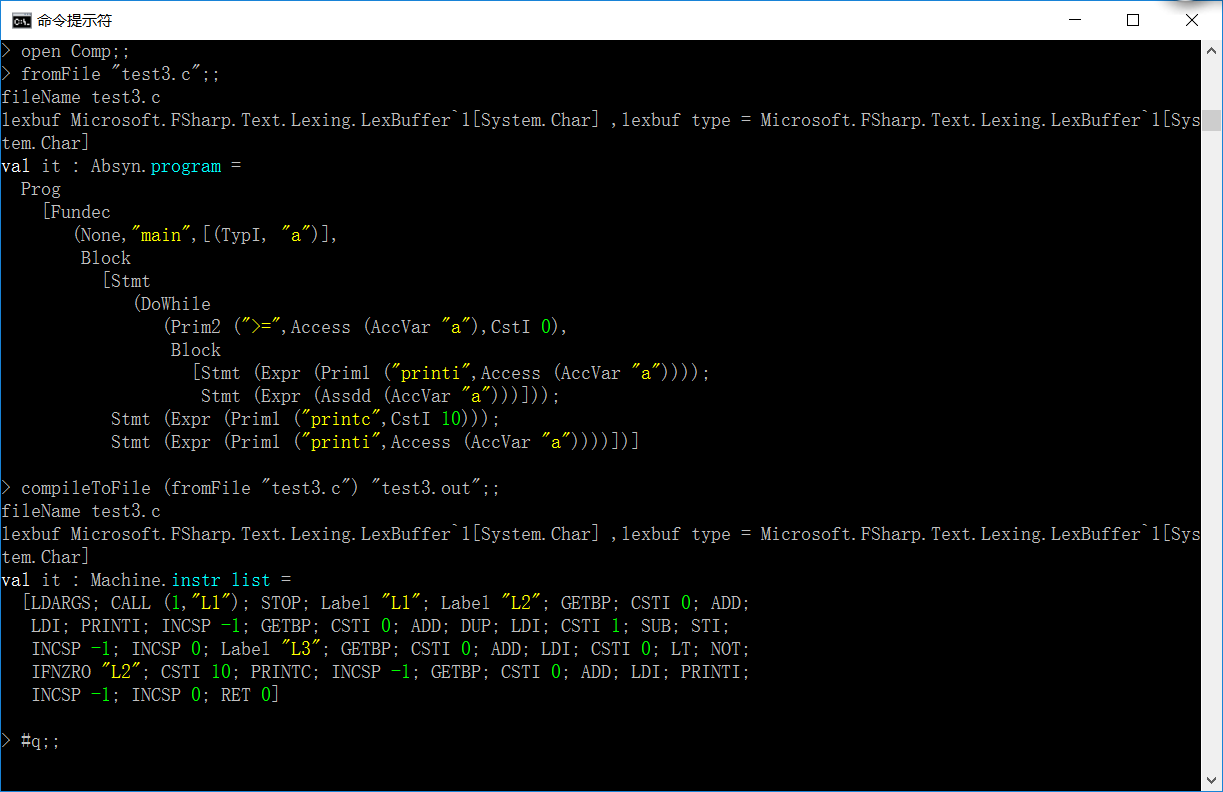
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 2 | 无 | 256 | 通过 |

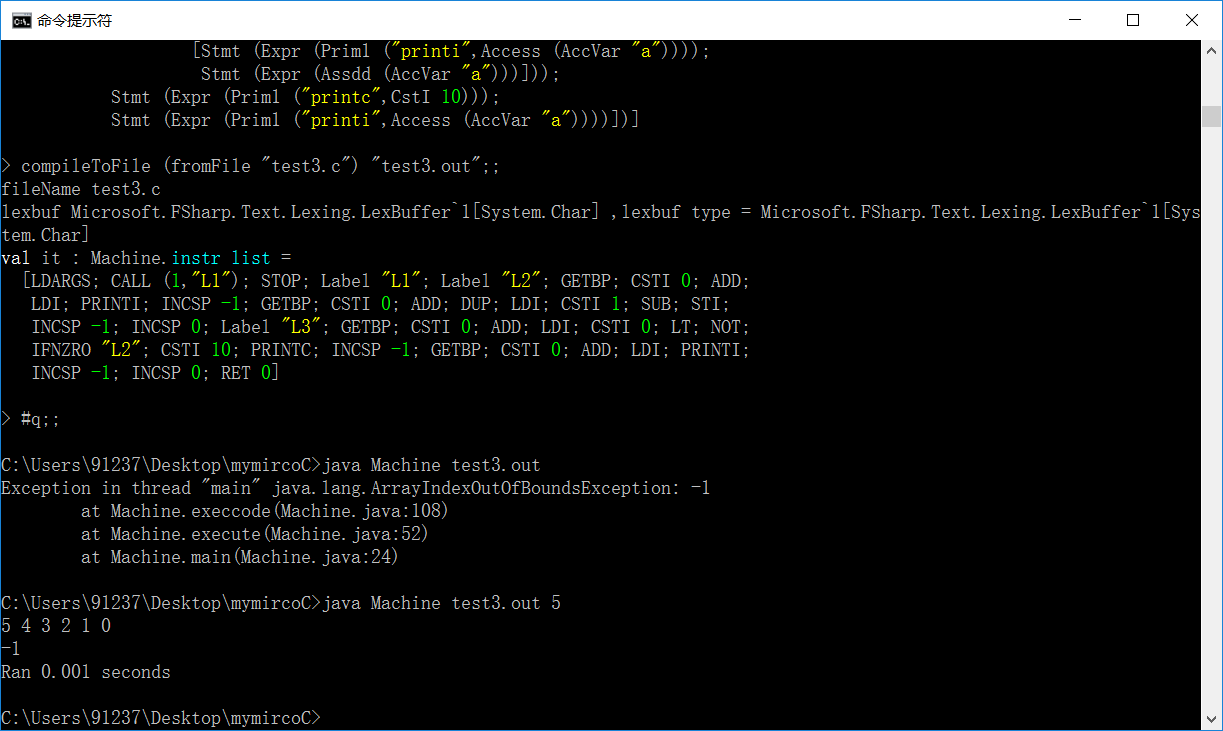


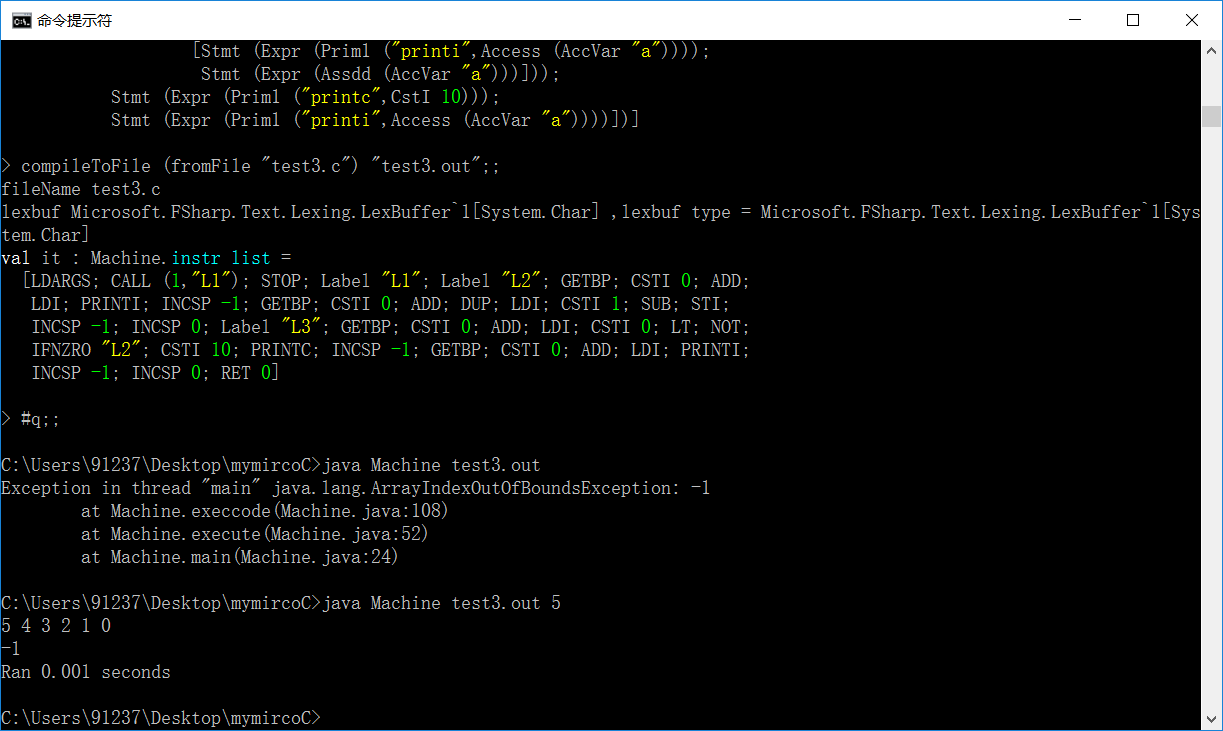




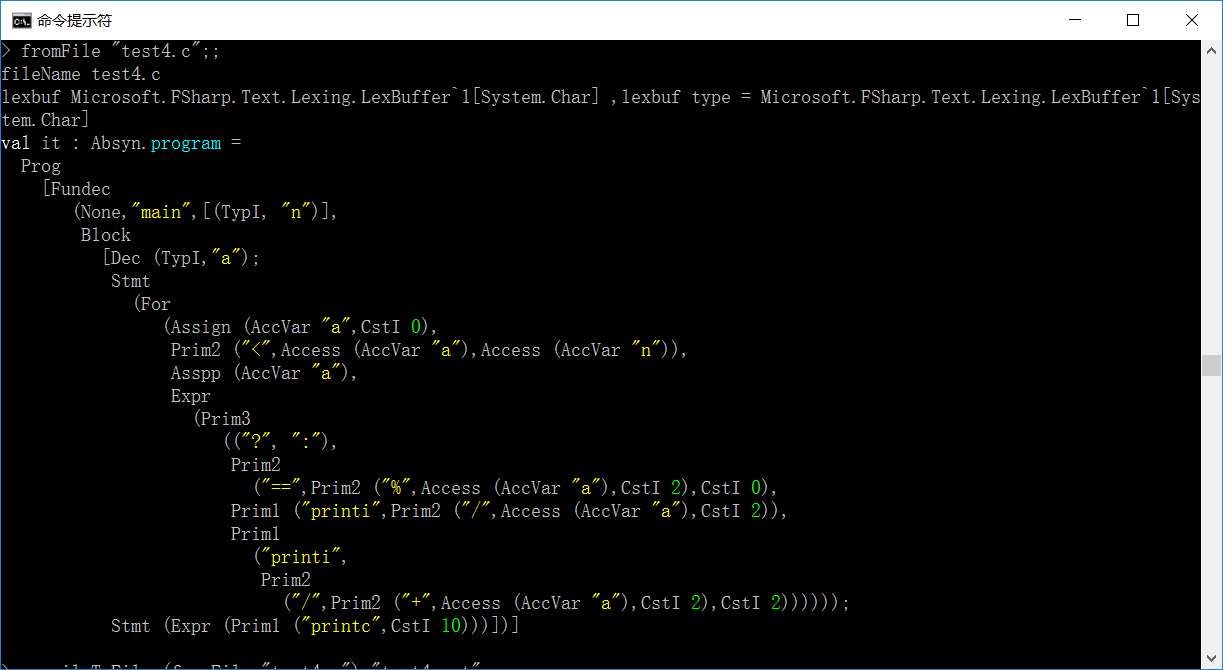
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 3 | 5 | 5 4 3 2 1 0 -1 | 通过 |

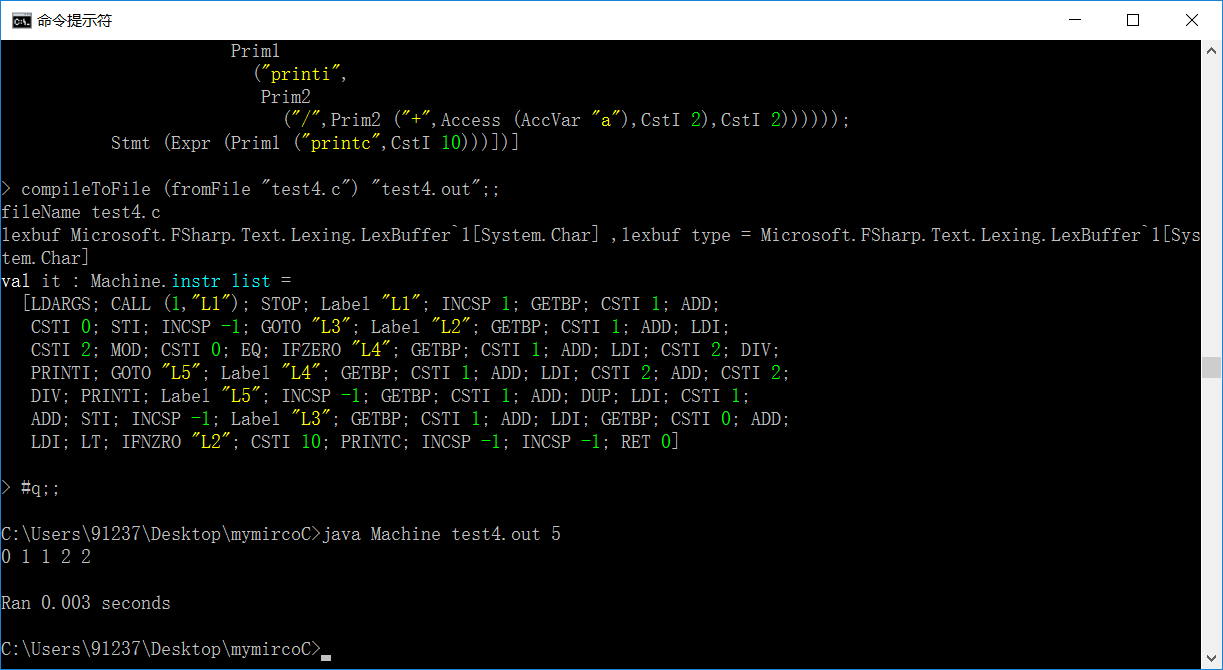


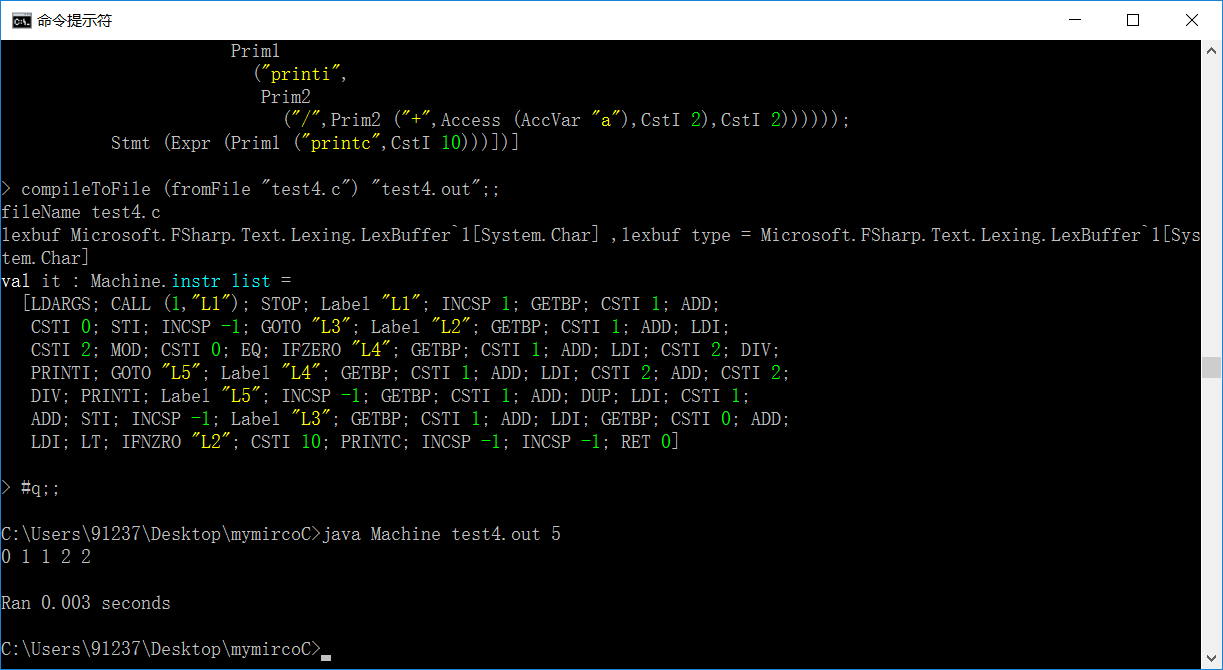




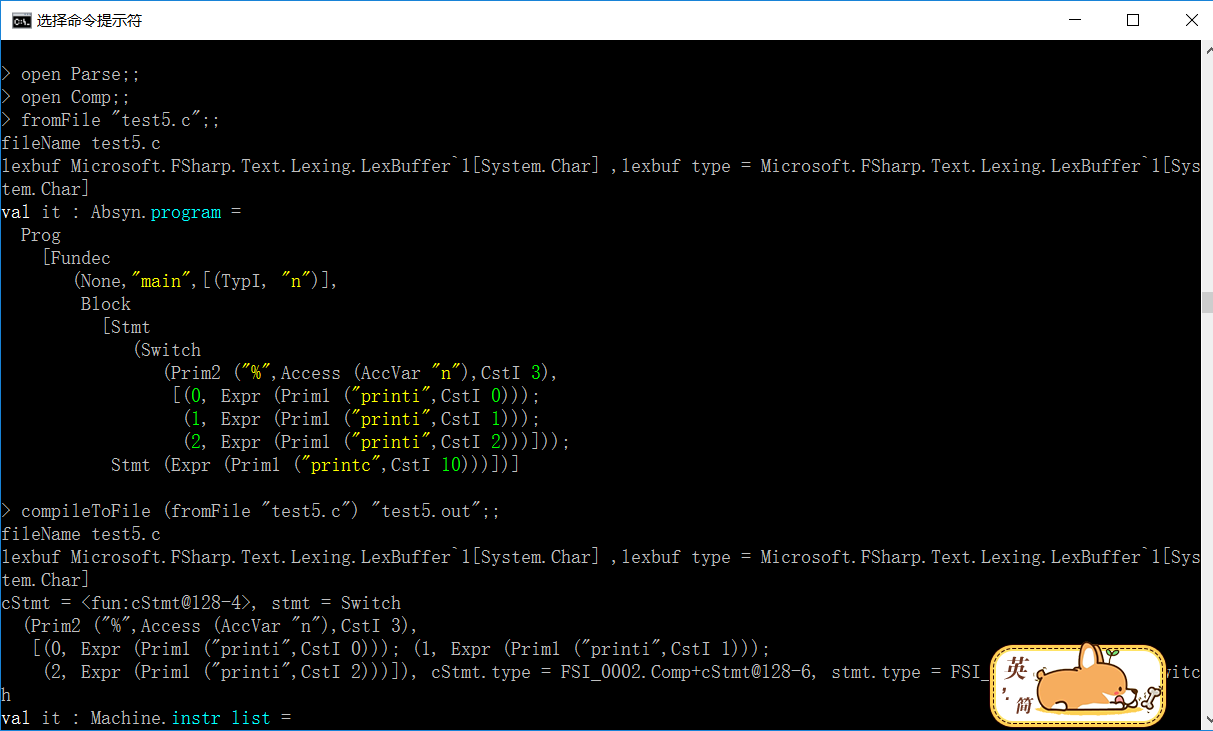
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 4 | 5 | 0 1 1 2 2 | 通过 |

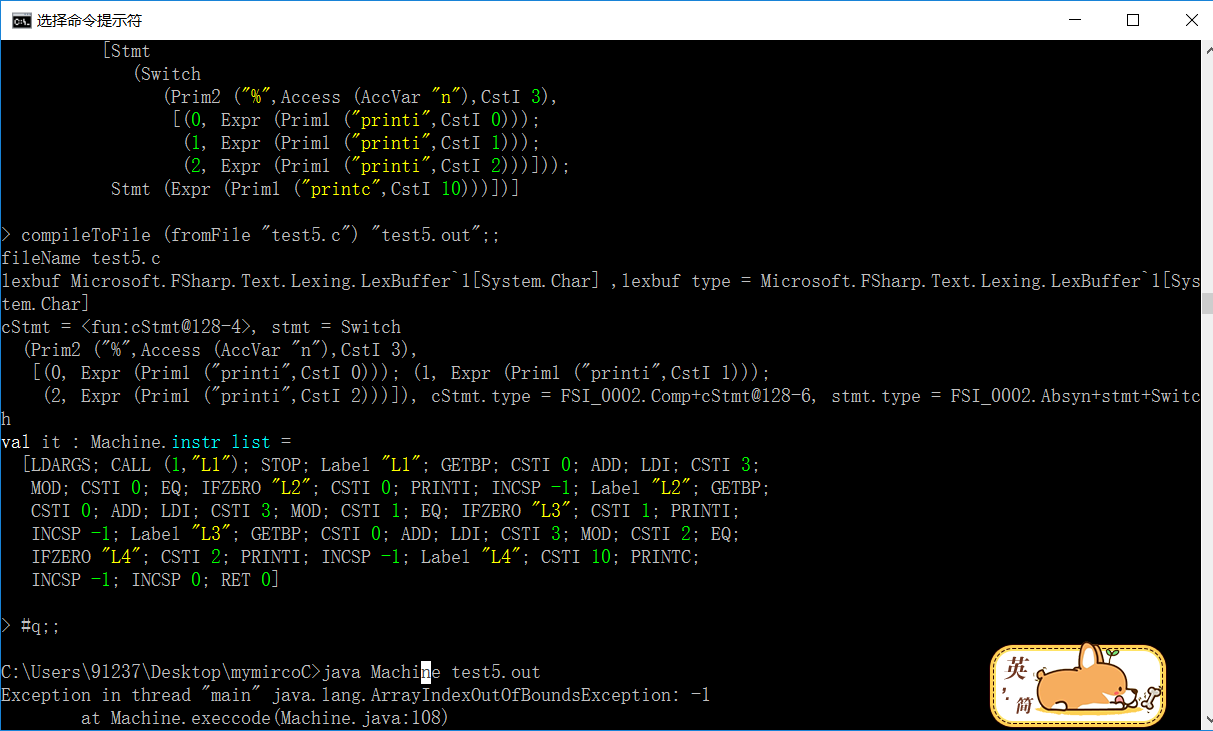


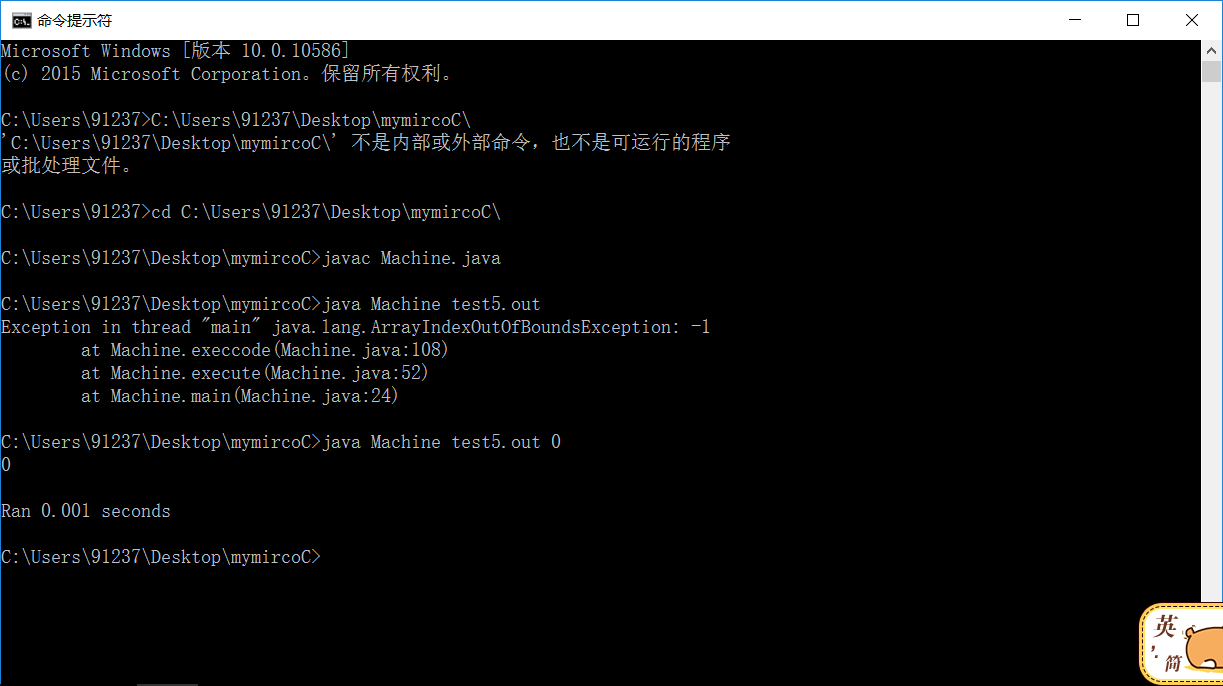




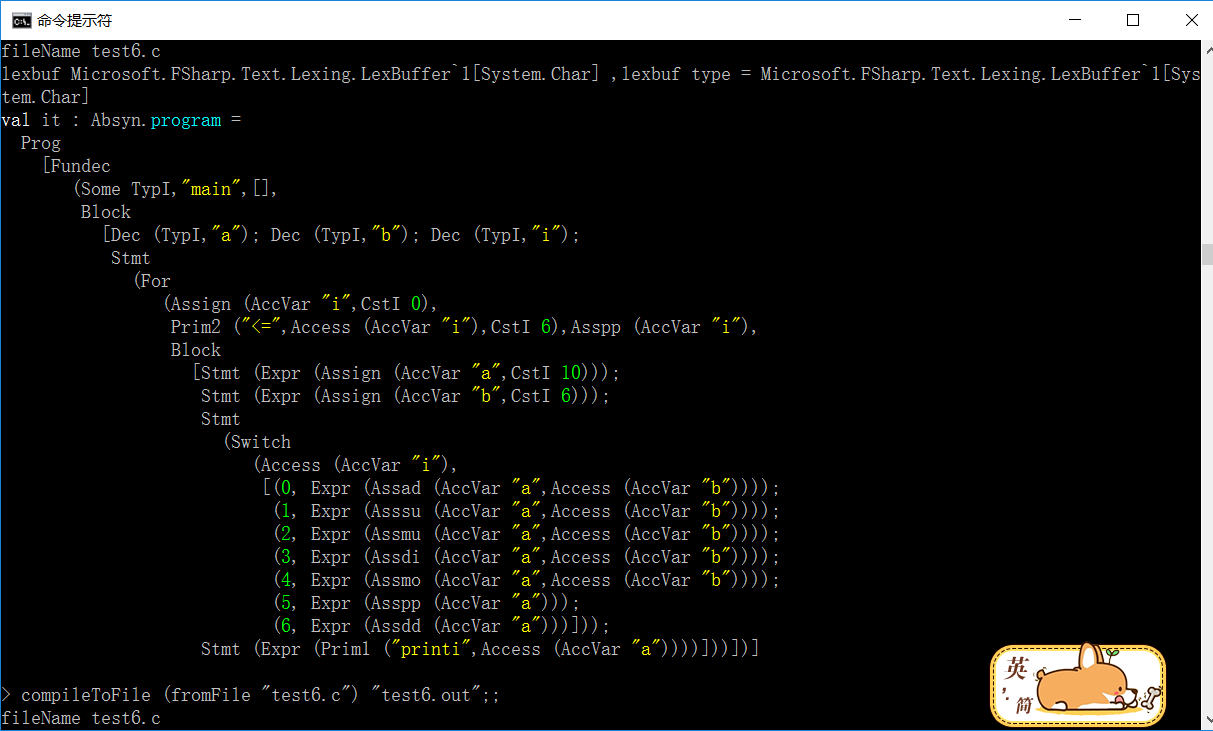
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 5 | 0 | 0 | 通过 |

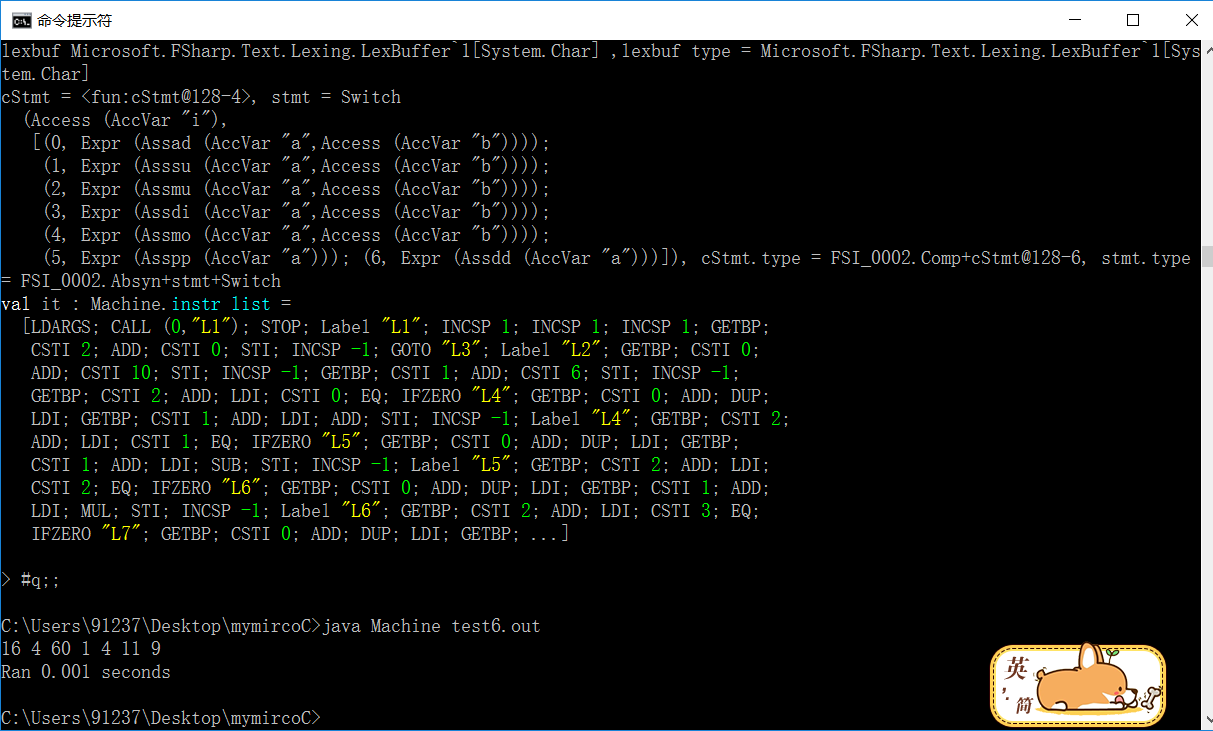


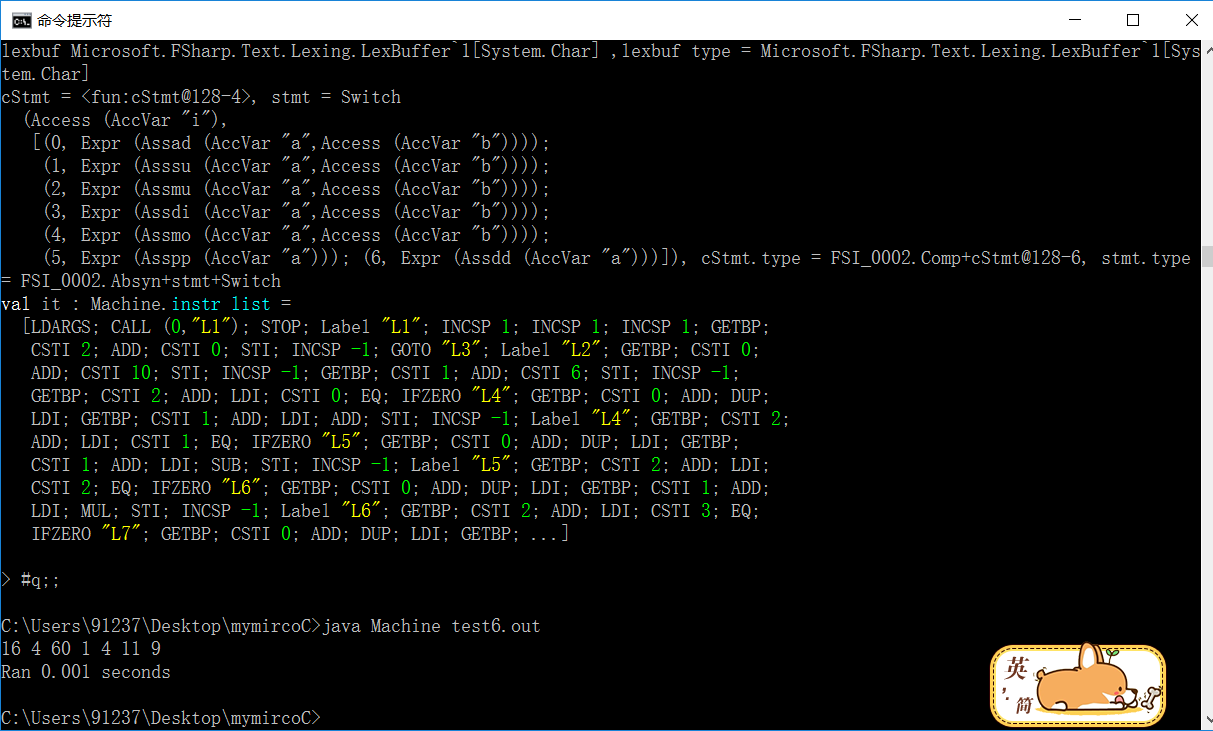




|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 输出 | 测试结果 |
| 6 | 无 | 16 4 60 1 4 11 9 | 通过 |







## 心得收获

### 课程体会与建议

许佳晨：

上完一个学期这门编译原理课程之后，感触很深:

平常上课认真听，也懂得了以前一点儿都不知道的东西，对编译原理也有了更深一步的了解和掌握。

重要的一点是：开学初拿到这本书的时候，感觉是比较高大上的，上课的时候会经常运用到它，但是每次上课以及课后作业的完成，基本都只需要老师的pdf和ppt只能搞定，所以最后就导致了又一本崭新的专业课本可以传承给下一届了。

对于期末的大作业，虽然有大约一个月的时间去完成它，但是还是感觉到了万分艰辛，因为最后我们使用了Fsharp语言实现的microC，尽管平常老师教的每个模块都很好，完成平时作业也没有什么特别大的问题，但是感觉最后大作业还是有一定的难度，所以还是需要付出一些时间来读一些文档和网上的示例代码，在日后的学习中，也要常常自学。

施怡昕：

我认为这门课是这个学期里比较难的课程了，要完全掌握对我来说是比较难的，所以在大作业中我也是尽量去完成自己能够完成的部分。在整个课程中；我认为老师很好的，讲课的时候很认真，还会给我们讲一些课外的拓展，每次都会有练习来巩固知识，但是这门课的难度有点高了，可能是上课没有完全认真的听讲，所以后面就一直跟不上，在后面的时候才慢慢把前面的知识点给补上来。

在这个课程中我们用到了JELAP、BABUN、FSHARP等等都是新学的，让我学到的东西变得多起来但是没有完整的掌握。还有老师在前几周的课程中，还让我们自己去学习了一些语言，当时我学的是PYTHON，因为有很多同学都已经接触过了，所以也希望自己也有一定的了解，当然还是希望自己能够在学习上坚持下来，不要只学一些浅的层面，比如在学习用了ANDROID STUDIO后可以做一个手机APP一样能用到实处。

## 项目体会

整个学期下来，感觉这门课是很难的，是需要话很多时间是认真学习的，而且知识点从浅入深，还在循环使用，如果其中一点没弄懂，后面就需要很多来填补。刚开始拿到大作业的时候也想了很久好做什么，最后还是选择了修改mirocC。老师本来的要求是要自己做一个编译器，但是能力有限只能自己会的做上去，我们在原有的mirocC的上加上了一些语法和语义，加上去的语法例如：DOWHILE语句、SWITCH语句、等等，还加上了一些加法、减法、乘法、除法的运算。虽然不是很多，但是我们也学习了很久，只要是能够自己完成这些，从慢慢开始不懂，然后能照着先前的学习，在写出自己东西。

在整个完成设计模块、编码模块、测试模块和编写文档模块的过程中，还是觉得每节课都是认真学习，不懂的及时搞懂才是最重要的。因为前面有些地方没有弄懂，后面就有觉得很难来做，所以以后在上课的时候还是要认真的听讲最好还是要课后能复习一下，多去把不会的问题去问老师和同学，才能真的在学到东西，为以后的学习打好基础。所以，大作业也是让我们在复习前面的知识来完成，让我们更加掌握一个知识点也是在做整个项目中的重要体会。所以这个大作业也是很必要的。

我们认为在团队的合作也是其中重要的部分，两个一起做的话，不懂的地方可以相互讨论，还可以相互学习对方会的东西，在整体上做一个补充在设计阶段、测试阶段等等的地方都是需要讨论的，有些时候也会有些争论，所以这时候需要很好的去沟通，任务才能继续下去，这也促进了我们的大作业顺利的完成了。

## 技术评价表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 一般 | 中 | 良 | 优 |
| 编译: 编译并在内存中运行 |  |  |  | O |
| 编译: 编译保存并执行 |  |  |  | O |
| 编译: 编译保存执行 |  |  |  | O |
| 编译: 在控制台输出结果 |  |  |  |  |
| 编译: 检测语法错误 |  |  |  | O |
| 语法: SWITCH CASE |  |  |  | O |
| 语法: FOR |  |  | O |  |
| 语法: DO WHILE |  |  |  | O |
| 语法: ?:表达式 |  |  |  | O |
| 词法: 八进制 |  |  | O |  |
| 词法: 十六进制 |  |  |  | O |
| 词法: ++、--、+、-、\*、/、% |  |  |  | O |
| 语义： |  |  | O |  |
| 测试：语法部分 |  |  |  | O |
| 测试：词法部分 |  |  |  | O |
| 测试：语义部分 |  |  | O |  |
| 文档：编写文档 |  |  |  | O |
| 文档：课程体会 |  |  |  | O |

## 分工表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 班级 | 学号 | 任务 | 比重 |
| 许佳晨 | 计算1501 | 31501081 | 编码、测试、文档 | 0.95 |
| 施怡昕 | 计算1501 | 31501079 | 编码、测试、文档 | 0.95 |

## GITHUB提交

