**Tiny C解释器**

TinyC是一个简化版C语言的***解释器***。该解释器读入一个简化版C语言源文件(\*.txt)，对该文件进行解释执行。

TinyC的词法和语法与C语言相同，简化的部分包括：数据类型只包含了整数类型；语句只包含了变量声明语句、表达式和赋值语句、if语句、while语句，增加了一个show语句，用于输出表达式E的结果（对于int型，直接显示其值；对于char型，显示该字符）；除main函数外不支持函数，不支持全局变量。

TinyC使用的文法如下（为便于理解，我们这里部分非终结符用中文表示，用[]表示可选部分）：

程序体🡪main() {函数体}

函数体🡪变量声明语句 函数体

函数体🡪赋值语句 函数体

函数体🡪show(E); 函数体

函数体🡪if (B) {函数体} [else {函数体}] 函数体

函数体🡪while(B) {函数体} 函数体

函数体🡪ε

变量声明语句🡪类型 id [=E] id\_list ;

id\_list🡪, id [=E] id\_list | ε

类型🡪int | char

赋值语句🡪id=表达式;

E 🡪 TEE1

E1 🡪 +TEE1 | -TEE1 | ε

TE🡪FTE1

TE1🡪\*FTE1 | /FTE1 | ε

F🡪id | num | (E)

B🡪TBB1

B1🡪‘||’TBB1 | ε

TB🡪FBTB1

TB1🡪&&FBTB1 | ε

FB🡪 E>[=] E | E <[=] E | E== E | E!=E | E | !B | TRUE | FALSE

1. **题目要求**

**1、实验内容：**

（1）理解代码，观察词法分析器输出的记号流和语法分析器输出的产生式。尝试编写一些源程序进行测试，检查并修改你发现的错误。你也可以改进TinyC，但只应当使它更贴近标准C语言，而不是相反。

（2）程序中，虽然在语义分析部分包含了char型数据的处理，但词法分析和语法分析并不完整，请完善代码，使之支持char型数据。

（3）在表达式计算中，允许逻辑表达式参与算术表达式计算。例如 a+(b>1)

（4）添加continue和break语句

**2、编程语言及环境：**

标准C语言

1. **课时安排**

实验共计30学时，时间分配可参考如下，但不强制要求：

* 实验内容（1），4学时。
* 实验内容（2），8学时。
* 实验内容（3），8学时。
* 实验内容（4），10学时。

1. **实验报告内容及评价标准**

整个实验最终提交一个实验报告、一份最终的“工程源代码”。

提交报告截至日期：最后一次实验后一周。例如，最后一次实验是6月5日，则提交报告截至日期为6月12日24:00。每迟交1天，扣1分。

**1. 实验报告内容及对应分数（16分）：**

* **实验内容（1）**，发现错误所使用的测试源程序，分析错误原因，并说明对程序的修改。为工程写一个完整的说明文件。 *（3分）*
* **实验内容（2）**，分别说明在词法分析器和语法分析器中是如何修改的。是否发现了语义分析部分的错误，给出发现错误所使用的测试源程序，以及修改方法。给出最后的演示源程序例子。 *（4分）*
* **实验内容（3）**，你是否修改了文法？如有，给出文法的修改部分，并表述相应的语义动作。给出最后的演示源程序例子。 *（5分）*
* **实验内容（4）**，分别说明词法、语法、语义分析是如何实现的。给出最后的演示源程序例子。 *（4分）*

**2. 实验报告质量要求（4分）：**

* 实验报告应清晰地说明问题、方法，不能简单地粘贴代码。不要简单粘贴课本、课件内容，也无需重复课堂上讲过的原理。（1分）
* 演示源程序例子应简洁，并能准确、全面地反映你修改后解释器的能力。（1分）
* 目录、章节自行设计，它应该能够让人从目录就清晰地看出你具体做了什么。（1分）
* 排版应清晰、简洁、美观。（1分）

**允许参考，但不得抄袭！一经发现，双方均计0分。**