

328

G

22

4

2022

2

22

21

面向对象程序设计第一次上机

宋胜利

2003051

陈禹译

20009200485

西安电子科技大学计算机科学与技术学院

1. **实验目的**

1、写一个atoi函数，能够把char数组形式的十进制、八进制、十六进制的数转化为int类型量。

2、写一个itoa函数，把int类型的数字转化成char数组形式。

3、实现desk calculator。

1. **实验环境**

IDE：clion cmake mingw

os：windows

硬件：Dell g15 5511笔记本

1. **实验内容**

1、通过%c方式输入十进制、八进制、十六进制的量，通过条件判断和计算得到最终结果值，返回int值，途中要解决错误输入和溢出产生的error。

2、通过%d方式输入一个数字，通过计算得到各位的量，存储在数组中，并输出数组值。

3、实现教材6.1中的desk calculator，输入一个加减乘除表达式，输出表达式计算结果；可以计算多行表达式，并分别输出结果；要处理非法字符、除以0、括号不匹配等error，但不需要中断程序或给出过于复杂的处理；另外，添加error指示错误行数的功能。

1. **实验步骤**

1、

1. 首先处理负号。如果读到第一个负号是负号，则额外用flag标记其为负，然后让整个数组左移一位，使得数字处理统一。
2. 根据数字表达的逻辑，可以如下判断进制base：默认10进制，第一位0则是8进制，第二位再是x则是16进制（注：此时要注意输入字符串的长度）。
3. 然后只要重复ans=ans\*base+a[++i]即可得到结果ans。
4. 处理非法输入error：依旧分逻辑阶层处理。在getDigit(char c, baseEnum base)函数内，先读0~7，若c不是0~7且base=8，则可知有非法输入；同样的，若读到9还是不符合且base=10，则是非法输入；若读到F还是不符合，则是非法输入。
5. 计算过程中的溢出判断：加法：(a>0.0 && b > I\_MAX - a) || (a < 0.0 && b < -I\_MAX - a)。乘法：abs(a)>1.0 && b > I\_MAX / abs(a)。注：这时之后一次作业要求实现的功能，这里基本照搬，且各题的溢出判断都能用。

2、

这道题并没有什么难点。

1. 不断对10取余、除以10，把取余数按顺序存入一个int数组。并保存操作次数。负号用flag\_negative另记占位。
2. 然后，翻转int数组存入char数组b[]中并输出。负号使用flag\_negative参与下表计算来占位。b[flag\_negative + p - 1]= '0' + temp[p2++]。
3. 输入-2147483648时，由于负数处理自动取相反数，出现了2147483648，溢出，产生错误。因此仅对-2147483648特判，立刻输出结果结束程序即可。

3、

1. 我将书上的代码切割、模块化，分成三个namespace：error相关的Error，与get\_token相关（包括token\_value枚举类）的Parser\_BasicReader，与分析输入、四则运算相关的Parser\_Arithmometer。
2. 输入算法基本照搬书上。功能里面有一个存储变量值并且运算的功能，让我学会了使用map进行检索来实现变量名保存。
3. expr、term、prim三个函数，我的理解：表达式按照优先级分析，则入口为expr，将表达式以最低优先级的加减法切割。然后所有求值调用term，使得加减法的那些部分使用乘除算的值后再加减。但term读每个具体值时依旧需要统统调用prim，因为数字、变量符号、取负、括号都是最高的优先级的（同时按数字>变量负号>取负>括号排列具体优先级）。

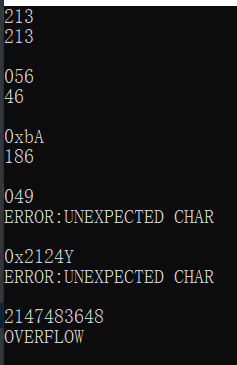
expr和term函数的具体逻辑都是：

取得左值-->寻找对应的下个符号（expr就是找加减）-->计算右值并且和左值进行对应运算。

(4)实现“显示error所在行数”功能：使用line\_indicator记录当前读取的行数（每当读到“；”或“\n”时自增），那么报error的时候就能输出行数了。但这样出现一个bug：8/0报错行数会比8/0+1多一行。因为根据读取相关算法逻辑，8/0会在读到“\n”使得line\_indicator自增后才抛出error，而“8\0+1”则是直接抛出error。解决这个问题，只需要在输出行数时，若当前所读值curr\_tok为PRINT换行符时，输出的行数设为line\_indicator-1，否则就是line\_indicator。

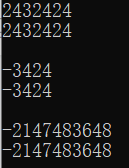
1. **实验结果**

1：

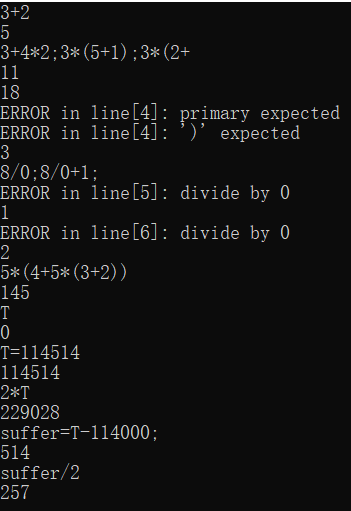


正确处理了非法输入和溢出，没问题。

2：



处理了刁钻的异常，并且正确处理了负数，没有多余项，没问题。

3：

计算没问题。error行数计算没问题。error抛出没问题。实现了变量名、变量值的存储与调用、计算。

1. **实验总结**

我认为这次实验的核心主题就是语句分析（书本上desk calculator部分也强调了认识编译器对语句分析的基本方法的重要性）。atoi里面有十进制<八进制<十六进制的优先级，desk calculator里面有上述提到过的各种符号的分析优先级，都需要对应的逻辑代码来实现。通过实践代码，我更加深切地明白了句法分析的方式（我感到很震惊，因为计算器的句法分析对以前的我来说是极度复杂以至于几乎不可能的事情，但书上的代码用三个简单的函数搞定了内核逻辑），对编程语言的理解也进一步加深了。

此外，一些error处理也是极有意义的，因为error不仅仅是用户的错，它代表了代码逻辑的内在的不完美。它不仅督促我们完善这个逻辑，更是反向地帮助我们加深对这逻辑的理解。