西安电子科技大学
XIDIAN UNIVERSITY

面向对象程序设计

Object Oriented Programming

实验报告

Experimental Report

学号	2000920048	姓名	陈禹译
班级	2003051	任课教师	宋胜利
实验名称	面向对象程序设计第一次上机		
实验学期	20 21 - 20 22 学年第 2 学期		
实验日期	2022 年 4 月 22 日	实验地点	G 楼 328
报告成绩		评分教师	

西安电子科技大学计算机科学与技术学院

一、 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。

二、 实验环境

IDE：clion cmake mingw

os：windows

硬件：Dell g15 5511 笔记本

三、 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。

四、 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。

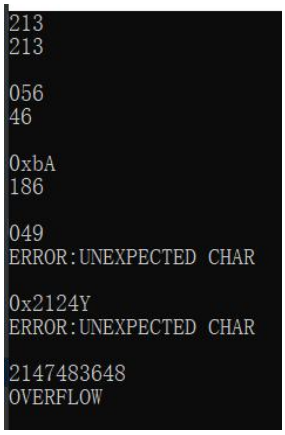
(3) expr、term、prim 三个函数，我的理解：表达式按照优先级分析，则入口为 expr，将表达式以最低优先级的加减法切割。然后所有求值调用 term，使得加减法的那些部分使用乘除算的值后再加减。但 term 读每个具体值时依旧需要统统调用 prim，因为数字、变量符号、取负、括号都是最高的优先级的(同时按数字>变量负号>取负>括号排列具体优先级)。expr 和 term 函数的具体逻辑都是：

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。

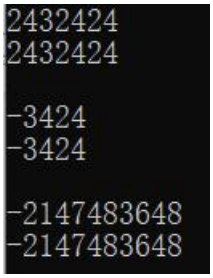
五、 实验结果

1：



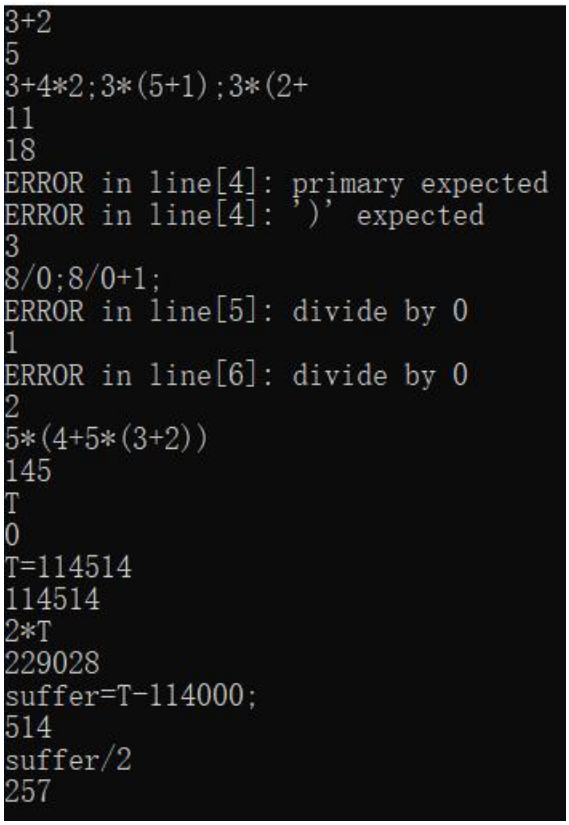
正确处理了非法输入和溢出，没问题。

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。

六、 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。