

CS205 C/ C++ Program Design - Assignment 2

Please implement a much better calculator than that in Assignment 1.

Requirements:

1. When you run your program and input an express in a line as follows, it can output the correct results. The operator precedence (order of operations) should be correct.

```
2+3
5
5+2*3
11
```

2. Use parentheses to enforce the priorities

$(5+2)*3$
21

3. Variables can be defined as follows

```
x=3
y=6
x+2*y
15
```

4. Some math functions can be supported

```
sqrt(3.0)
1.7
```

5. It can support arbitrary precision.

[illegible]

6. More features which can be found from the calculator BC in Unix-like systems. You can visit this page for more information.

https://www.gnu.org/software/bc/manual/html_mono/bc.html

- To host your source code at GitHub.com is an option. You are strongly recommended to create a repository at GitHub.com to hold your source code for this assignment. If so, you can just put a link in the report of the assignment, and do not need to upload the source code to Blackboard. If you do not host your source code at GitHub.com, please upload your source with your report to Blackboard.

Rules:

1. Please submit your assignment report before its deadline. After the deadline (even 1 second), 0 score!

(For students who register this course before Sep. 6, they should submit their assignment reports before 23:59 on Sep. 27. For the rest students they should submit their

assignment reports in two weeks after they register.)

2. If you only implement requirement 1 and 2, the upper boundary of your score is 80. For a better score, you should implement the rest requirements. Your score will also depend on the quality of your source code and your report. Your report should be easy to understand and describe your work well, especially the highlights of your work.
3. Please pay more attention to your code style. After all this is not ACM-ICPC contest. You have enough time to write code with both correct result and good code style. You will get deduction if your code style is terrible. You can read Google C++ Style Guide (<http://google.github.io/styleguide/cppguide.html>) or some other guide for code style.

CS205 C/ C++ Program Design

Assignment 2

Name: 王奕童

SID: 11910104

【说明】本次 Report 共 25 页，包含大量说明与测试截图，阅读需要一定时间

Part 1. Description

关于本次作业的源码描述：

一、本次作业共上交 5 个文件：

1. main.cpp
2. Variable.cpp
3. Variable.h
4. CWTNumber.cpp
5. CWTNumber.h

【声明】本次作业中的 CWTNumber.cpp 和 CWTNumber.h 为网络上查找到的较好的源码库。

文章原链接：<https://www.cnblogs.com/acbingo/p/4116206.html>

文章作者：ACBingo

本次作业对于该源码库进行了改进和提升，以能够适应本次作业的需求。调用和改进该链接的源码库应用于本次作业已经得到文章作者的同意授权。

二、关于 CWTNumber 源码头文件的解释说明：

1. 支持由 const char* 指针和无类型来创建对象
2. 支持对象销毁的时候的内存管理
3. 支持对象转化为人类可读的小数或者整数进行运算
4. 支持多余 0 的清除操作
5. 支持任意长度的整数和小数的加减乘除运算
6. 支持 c++ 的 double 类型的运算符重载
7. 支持小数点的左移右移
8. 支持两个 CWTNumber 对象的比较大小

三、关于 Variable 头文件的解释说明

1. 该类用于保存变量名和变量代表的值（代表的值用 CWTNumber 保存），将 std::string 对象和 CWTNumber 设置为 private 类型封装起来
2. 该类提供读取变量名和变量值的 public 对象函数
3. 该类提供修改变量值的 public 对象函数
4. 该类主要目的为应用于实现 Requirement3 的自变量保存
5. 该类后由于内存超限被弃用

四、关于 main 函数实现的基本功能：

0. 事先定义了两个基本常量：E=2.718281828459，PI=3.1415926535898，并且禁止用户在自变量命名的时候对于基本常量进行修改操作

1. 支持基本数学加减乘除（Requirement1）
2. 支持带括号的基本数学加减乘除（Requirement2）
3. 支持对于左右括号是否匹配的检查操作
4. 支持单个字符的自变量命名（Requirement3）
5. 支持对于自变量命名的命名规则检查，检查规则与 java，C/C++一致
6. 支持对于不存在的字符自变量的检查并显示错误信息
7. 支持部分非复合数学函数的运算（Requirement4）

（支持：sqrt(a), pow(a,b), max(a,b), abs(a), exp(a), min(a,b), random(), sin(x), cos(x), tan(x), ceil(x), floor(x), log(x)）

【注意：以上函数是基于 math.h 头文件的相应函数进行设计，因此结果有可能受到计算精确定度的限制】

8. 支持高精度的数字运算（Requirement5）
9. 支持用户写入注释，支持形式为单行多行均可，语法规则与 java、C/C++一致
10. 支持显示欢迎语句和帮助文档，帮助用户熟悉本计算器具有的功能
11. 支持中英双语，支持用户进行两种语言的相互转换
12. 支持用户正常退出计算器

五、TO DO：

1. 支持实现更多的数学函数
2. 支持数学函数的复合运算
3. 支持自变量以字符串形式命名
4. 支持字符串自变量自增和自减操作
5. 支持程序内写入 while 循环语句

Part 2. Result & Verification

In this part, you should present the result of your program by listing the output of test cases and optionally add a screen-shot of the result.

【Test case #0】欢迎语句

以下所有测试用例都包含欢迎语句，因此将欢迎语句截图至此，在此报告中避免多次冗余显示。

Input: nothing

Output: Several sentences for welcome.

Screen-short for case #0:

```
-----  
Welcome using this simple calculator!  
This calculator is mainly for the Assignment2 in CS205(C/C++) course.  
This calculator is much better than that in Assignment1!  
Course Instructor: Shiqi Yu  
Author: YeeTone Wang  
Version: 1.06  
If you need help, please print help to get some information.  
If you need welcome, please print welcome to get this information again.  
If you need exit, please print exit to get exit information.  
如果需要显示中文, 请输入Chinese  
-----
```

Please enjoy your calculation!

【Test case #1】不带括号的基本加减乘除

Input:

2+3

5+2*3

Output:

5

11

Screen-short for case #1:

```
-----  
Please enjoy your calculation!  
2+3  
5  
5+2*3  
11
```

【Test case #2】带括号的基本加减乘除

Input:

(5+2)*3

((5+2*0)+6)*2

Output:

21

22

Screen-short for case #2:

```
-----  
Please enjoy your calculation!  
(5+2)*3  
21  
((5+2*0)+6)*2  
22
```

【Test case #3】自变量命名并计算

Input:

```
x=3
y=6
x+2*y
x*x+y*(x+x)
```

Output:

```
15
45
```

Screen-short for case #3:

```
-----
Please enjoy your calculation!
x=3
y=6
x+2*y
15
x*x+y*(x+x)
45
```

【Test case #4】 数学函数运算

Input:

```
sqrt(3.0)
exp(3)
abs(-300)
max(-1,3)
random()
sin(PI/6)
```

Output:

```
1.732051
20.085537
300
3
41
0.500000
```

Screen-short for case #4:

0.500000

9.99

【写在前面】本次 Assignment 很像 CS102B 课程的最后一次作业，但是难度是升级版

Home / Problems / Problem

Problem 38: Calculate String

🕒 10s 📄 2048 MiB 🟢 Easy

Problem

Answer

Problem & Answer

Calculate String

Description

Give you a simple expression string, please calculate it and print the answer.
The expression string may contain (,), +, -, *, /, and non-negative integers.
We guarantee that the given expression is always valid and we won't get decimals numbers after division.

Input

The first line is an integer n , which means there are n different expression strings.
Next n lines are expression strings.

Output

Print the answer of each expression string.

Sample input

4
1+1
7-4/2
2*(5+5*2)/3+(8/2+3)
(2+6*3+5-(3*14/7+9)*5)-3

Sample output

2
5
17
-53

Problem1: 基本数学逻辑运算

在中小学的学习过程中，我们对于四则基本运算的运算方法已经滚瓜烂熟，下面给出一个简单的解决基本数学运算的计算方法。

Sample Question:

$2+3*3+5$

Sample Answer:

- ①在运算中 $3*3$ 的乘号优先级最高，因此先计算 $3*3$ 得到 9;
- ②后续两个加法的优先级一致，因此从左到右计算：得到 $2+9+5=16$

在上面的求法中，可以通过递归/循环实现，不过考虑到该方法不容易实现后续的要求，故将思路更换为下面问题点中介绍的后缀表达式运算方法

在 java 语言中，有 JexlEngine 库可用于计算表达式的值。而 c/c++ 语言中，我暂时没有找到类似简单的语法规则计算器，因此只能自己用心一步一步来写出来。

可能存在的问题点：

1. 四种运算规则的优先次序
2. 数字的读入与存储
3. 正确结果的计算
4. 负数情形和小数情形的处理
5. 特殊情形：除数为 0 的情形

问题解决：

1. 四种运算规则的优先次序

构建 prior 函数：传入类型为 char 类型，返回类型为 int 类型，在传入*或者/的时候返回 2，在传入+或者-的时候返回 1，其余返回 0 即可。

具体方法：

```
static inline int prior(char op);
```

2. 数字的读入与存储

3. 正确结果的计算

分为几个步骤：

①将字符串预处理，处理掉一些特殊的情形如-1+3，3*(-1) 等等

构建 format 函数：在特定位置插入'0'字符

【特定位置：首个运算数字为负数，或者括号第一个数字为负数】

具体方法：

```
static inline string format(string expression);  
static inline string pretreat(string expression);
```

②③均在

```
static inline string solve(string expression);
```

函数里面实现

②将中缀表达式的 string 类型转化为后缀表达式的 string 类型

参考学习链接：<https://blog.csdn.net/sgbfblog/article/details/8001651>

应用自带库：c++的<stack>库，以及 c++的<vector>库

构建 middleToBack 函数，传入中缀表达式的 string 对象，返回后缀表达式的 vector 对象

具体方法：

```
static inline vector<string> middleToBack(string expression);
```

③将后缀表达式计算为最终的结果

参考学习链接：<http://www.nowamagic.net/librarys/veda/detail/2306>

应用自带库：c++的<stack>库

应用外源库：CWTNumber.h 头文件

构建 result 函数，传入在②中的后缀表达式 vector 对象，计算正确的结果

具体方法：

```
static inline string result(vector<string> bh);
```

④将结果打印出来

```
cout << solve(str) << endl;
```

4. 负数情形和小数情形的处理

负数在 format 函数中已经初步处理了异常的情形，然后接下来在 middleToBack 函数里进行识别和判断。

实现方法：逐个字符扫描，若是运算符 (+-*/) 则进行判断，需要判断运算符与栈顶的优先级；若是运算符低于栈顶优先级则取出栈顶计算，否则直接入栈；若不是运算符则用一个 while 循环扫描数字和小数点，直到扫描到非数字或小数点为止。

具体方法：

```
static inline string format(string expression);  
static inline vector<string> middleToBack(string expression);
```

5. 除数为 0 的情形

CWTNumber.h 头文件已经针对除数为 0 的情形进行了处理，因此直接调用 CWTNumber.h 的除法进行运算即可。

【Test case #6】除数为 0 情形

Input:

4/0

Output:

#INFINITE

Screen-short for case #6:

```
-----  
Please enjoy your calculation!  
4/0  
#INFINITE
```

Problem2: 括号的基本处理

在确保左右括号数目匹配的前提下，在基本的运算法则中，括号括起来的表达式的优先级将被提升，将会优先计算。

参考学习链接 1:

https://blog.csdn.net/weixin_40500026/article/details/80767071

参考学习链接 2:

https://blog.csdn.net/qq_39445165/article/details/87641261?utm_medium=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.edu_weight&depth_1-utm_source=distribute.pc_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.edu_weight

基本思路：先检测左右括号数目是否匹配，然后 Problem1 的 stack 上进行改进，增加左括号和右括号的优先级判断。

升级后栈的运算法则：

1. 遇到数字（可识别小数）时直接压入栈内
2. 遇到运算符则判断和栈顶的运算等级，如果栈顶等级高则取出栈顶元素计算，并将结果压入栈内，直到栈为空或者栈顶等级不高于运算符
3. 遇到左括号直接压栈，此后对于除了右括号的任意运算和数字均直接压入栈内；
4. 遇到右括号则持续出栈运算，直到遇到左括号；
5. 最终取出栈顶的元素即可

具体代码的实现：

```
static inline bool isBracketMatching(string expression);
```

```
static inline vector<string> middleToBack(string expression);
```

【Test Case#7】左右括号数目不匹配

Input:

`((2+3)*6`

Output:

The bracket is not matching! (error message)

Screen-short for case #7:

```
-----  
Please enjoy your calculation!  
((2+3)*6  
The bracket is not matching!
```

Problem3: 自定义变量

在之前中学学过的方程问题中，我们对于问题的求解是基于自变量与对应的数值进行替换，从而得到结果。

Sample Question:

已知 $x=3$ ，请问 $2*x+4=?$

Sample Answer:

- ①将所要计算的式子中的 x 用 3 替换掉，得到 $2*3+4$
- ②计算 $2*3+4=10$ ，得到计算结果为 10

基于 Sample Q&A 的思路，实现自变量命名可以使用字符串替代为真正数值的方法来解决该问题。

可能存在的问题点：

1. 如何判断语句是不是在赋值？以及变量名是否符合命名规则？
2. 如何保存语句中的变量名和对应的数值？
3. 如何对后续语句的字符变量进行替代从而计算结果？
4. 如何重复修改先前定义过的变量？以及如何阻止用户对于常量 π 、 e 进行修改操作？
5. 如何检测未定义的变量？

1. 语句是否在赋值的判断？命名规则是否合法？

语句赋值至少有三个要素：

- ①左侧是一个合法变量名
- ②中间有一个等号
- ③右侧是一个可计算的表达式

实现思路：

- ①自建一个简易版 ArrayList 类，最大上限容量为 1000，内部支持 java 的 ArrayList 类的一些简单操作，如添加元素 (add)，读取元素 (get)，设置元素 (set)，读取当前已用容量 (getSize)，根据元素寻找下标 (indexOf) 等等。

实现代码：

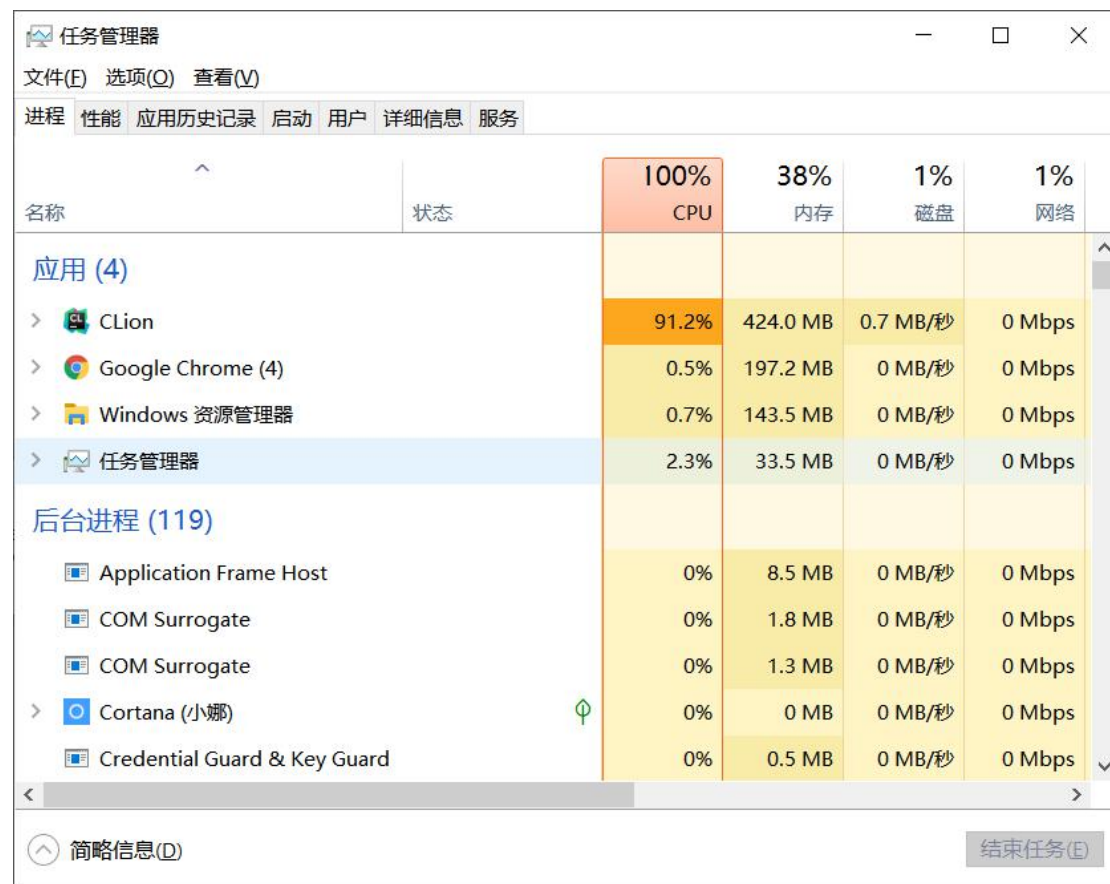
```
template <typename T> class ArrayList;
```

②设置全局静态 ArrayList 变量用于保存所有自变量信息和自变量值。因后续可能会涉及到高精度数值的运算，所以放弃了使用 double 类型而改用 string 类型。

```
static ArrayList<string> varNameList;
```

```
static ArrayList<string> valueList;
```

【小插曲】先前提到的 Variable.h 由于不明原因在将 2 个 String 的 ArrayList 合并成 1 个 Variable 的 ArrayList 后输入任何信息会直接爆内存（电脑 CPU 占用高达 100%，程序直接异常退出，错误码是 exit code 3），故弃用该类。



③利用 string 类的 find 函数，定位等号出现的位置，然后将字符串以等号为界，一分为三，则前一部分为变量名（合法性还未判定），中间部分仅有一个等号，后一部分为赋值语句。其中，如果变量名不合法则认为该语句不是在赋值。

【合法命名规则介绍】首个字符必须是下划线或者字母，并且后续组成必须由下划线、数字和字母组成。

判定是否在赋值实现具体函数：

```
static inline bool isAssigning(string expression);
```

【Test Case#8】自变量命名不合法

Input:

1_123=1

Output:

The variable name is not valid! (error message)

Screen-short for case #8:

```
-----  
Please enjoy your calculation!  
1_123=1  
The variable name is not valid!
```

2. 如何保存语句中的变量名和对应的数值？

④判定是一个合法的命名后，即可开始添加元素进入 ArrayList 中储存。

有 3 种可能：

1. 该自变量名是常量名（PI 或者 E），与常量发生了冲突，则提示用户错误信息
2. 变量 List 中含有该自变量名，则替换原本的自变量对应的数值
3. 变量 List 中不含有该自变量名，则新建添加自变量名

实现方法：

先将右侧的数学函数，自变量等等元素化到最简（利用自己写的 solve 函数），然后 for-loop 遍历自变量 List，看有没有相应的元素，如果是常量则为情形 1，如果是自变量则为情形 2，其他则为情形 3。然后在不同的情形下对于变量 List 进行 set 或者 add 操作即可。

具体实现：

```
static inline void addVariable(string expression);
```

3. 如何对后续语句的字符变量进行替代从而计算结果？

按照 Sample Q&A 的思路，我们可以将原本的表达式一分为三，分别为 before(要替换的字符串前面所有的字符)，after(要替换的字符串后面所有的字符)，以及要自变量字符串本身。

因此还是使用 string 类的 find 函数定位所在位置，然后字符串切割处理将字符串一分为三。由于变量名和变量值是同步添加的，因此下标是一致的。所以用 indexOf 找到位置后，将替换好的值给重新拼接即可。

具体实现：

```
static inline string replaceVar(string expression);
```

【TODO】该字符串切割方法存在一个问题，因此当前最保险的方式是使用单个字母进行自变量命名。具体问题请见下面这个测试用例：

【Wrong Answer】【已局部修复】

【Test Case#9】WA 的测试用例，表达式中出现复合字符串时会计算错误

Input:

x=2

y=3

xy+1

Output:

[Excepted1]

The variable does not exist! (error message)

[Excepted2]

7

[Actually]

24

Screen-short for case #9:

```
-----  
Please enjoy your calculation!  
x=2  
y=3  
xy+1  
24
```

修复该 bug 的思路:

- ①告诉用户只能用单个字符命名
- ②后续增加后以支持字符串自变量的命名

暂时修复成果:

Input:

x=2

y=3

z=5

xyz+xy+yz

Output:

51

Screen-short for case #9:

```
-----  
Please enjoy your calculation!  
x=2  
y=3  
z=5  
xyz+xy+yz  
51
```

4. 如何实现重复修改先前定义过的变量的要求? 以及如何阻止用户对于常量 PI、E 进行修改操作?

- ①for-loop 遍历变量名 List 查看有无先前定义过该变量
- ②存在的话则找对应的下标, 然后修改对应变量值 List 的数值即可

具体实现:

```
static inline void addVariable(string expression);
```

【Test Case#10】 用户修改先前定义的自变量

Input:

x=2

y=3

x=4

x+y*2

Output:

10

Screen-short for case #10:

```
-----  
Please enjoy your calculation!  
x=2  
y=3  
x=4  
x+y*2  
10
```

③如果修改的是常数对应的下标，则 cerr 出错误信息，阻止用户进行修改

【Test Case#11】 用户尝试修改定义的常量 PI 和 E

Input:

PI=31.4

E=27.18

Output:

This is constant value. Cannot be changed! (error message)

This is constant value. Cannot be changed! (error message)

Screen-short for case #11:

```
-----  
Please enjoy your calculation!  
PI=31.4  
This is constant value. Cannot be changed!  
E=27.18  
This is constant value. Cannot be changed!
```

5. 如何检测未定义的变量？

遍历 List 检查，如果含有则返回 true，否则返回 false。

具体方法实现：

```
static inline bool checkUnknownVar(string expression);
```

【Test Case#12】 用户输入未定义的变量名

Input:

x=1

y+2

Output:

The variable does not exist! (error message)

Screen-short for case #12:

```
-----  
Please enjoy your calculation!  
x=1  
y+2  
The variable does not exist!
```

Problem4: 数学函数的支持实现

在 java、C/C++ 语言中，我们已经知道了有很多的数学函数可供我们使用。现在结合 C/C++ 的 math.h 头文件的支持下，实现输入数学表达式的求值。

Sample Question:

4+sqrt(3.0)*3+3

Sampler Answer:

- ①先取出 sqrt(3.0)计算得到 1.732051
- ②代回去得到 4+1.732051*3
- ③计算得到 8.732051

基于此思路，我们对于该问题的求解思路为：

1. 剪切掉表达式里面所有的空格
2. 寻找定位数学函数的位置，并将原表达式一分为三
3. 调用 math.h 头文件计算中间仅含有数学表达式的数值
4. 再将前面+计算结果+后面拼接到一起

1. 剪切空格

新建目标空字符串，遍历表达式，如果遇到非空格则加入目标字符串，然后返回目标字符串。
方法实现：

```
static inline string clearEmptySpace(string expression);
```

2. 定位数学函数的位置，并一分为三

先基于 find 函数检测有无指定的数学函数，具体方法实现：

```
static inline bool containsCertainMathFunctions  
(string expression, string funName);
```

基于 string 字符串的 find 函数定位位置，然后使用 substr 函数将原字符串一分为三。

【提示】关于 substr 的坑：c++ 的 substr 函数与 java 中的 subString 方法有所不同：

c++ 的 substr 第一个参数是起始位置，第二个参数是读取的字符个数；而 java 的 subString 第一个参数是起始位置，第二个参数是结束位置。

用 substr 方法取出中间的数字，如 sqrt(3.0) 中的 3.0，然后使用 stod 方法转为 double 类型，然后调用相应的数学函数计算，最后使用 to_string 函数转回 string 类型进行拼接回去即可。

```
static inline string replaceAllMathFunctions(string expression);
```

sqrt(a), pow(a,b), max(a,b), abs(a), exp(a), min(a,b), random(), sin(x), cos(x), tan(x), ceil(x), floor(x), log(x)

当前运算不能完全处理复合数学函数处理的情形,只能处理单一数学函数的运算和部分次序的复合数学函数的运算

Screen-short for case #13:

9.000004

1. 支持由 `const char*` 指针和无类型来创建对象
2. 支持对象销毁的时候的内存管理
3. 支持对象转化为人类可读的小数或者整数进行运算
4. 支持多余 0 的清除操作
5. 支持任意长度的整数和小数的加减乘除运算
6. 支持整形小数绝大多数的运算符重载
7. 支持小数点的左移右移
8. 支持两个 `CWTNumber` 对象的比较大小

Output:

Screen-short for case #14:

[illegible]

用户可写入注释，main 程序自动忽视注释内的内容

具体方法实现：

【Test Case#15】单行注释测试

 $x+y$

5

5

具体方法实现:

Input:

```
/*x=3
```

```
y=4
```

```
x+y
```

```
x=6*/
```

```
x+y
```

Output:

The variable does not exist! (error message)

Screen-short for case #16:

```
-----  
Please enjoy your calculation!  
/*x=3  
y=4  
x+y  
x=6*/  
x+y  
The variable does not exist!
```

Problem7: 对用户的友好界面提示

主要支持 3 条来帮助用户熟悉计算器的使用说明

1. welcome
2. help
3. exit

1. welcome

当用户初次打开计算器或者输入 welcome 的时候，会显示对用户的欢迎信息：

具体方法实现：

```
static inline void printWelcome();
```

【Test case #17】 欢迎语句

Input:

welcome

Output:

Several sentences for welcome.

Screen-short for case #17:

```

-----
Welcome using this simple calculator!
This calculator is mainly for the Assignment2 in CS205(C/C++) course.
This calculator is much better than that in Assignment1!
Course Instructor: Shiqi Yu
Author: YeeTone Wang
Version: 1.06
If you need help, please print help to get some information.
If you need welcome, please print welcome to get this information again.
If you need exit, please print exit to get exit information.
如果需要显示中文, 请输入Chinese
-----
Please enjoy your calculation!

```

2. help

当用户初次打开计算器或者输入 help 的时候, 会显示对用户的帮助信息:

【Test case #18】帮助文档语句

Input:

help

Output:

Several sentences for help.

Screen-short for case #18:

```

-----
Please enjoy your calculation!
help
-----
This is the instruction to the calculator user.
Author: YeeTone Wang
Version: 1.06
The author firstly shows his gratitude to the blog of ACBingo.
This calculation is based on the library of "CWTNumber.h"
The original link: https://www.cnblogs.com/acbingo/p/4116206.html

0. The constant value in this calculator:
E=2.718281828459
PI=3.1415926535898

Supported functions:
1. Basic mathematical calculation: + - * /
example-1.1:
3*4
12
example-1.2:
3.45*4.32
14.904

2. calculation with brackets: ( )
example-2.1:
(2+5)*9.9
69.3

```


exit

Y

Output:

Are you sure to exit? Print[Y] to exit, otherwise will return
(main function exits properly)

Screen-short for case #19:

```
-----  
Please enjoy your calculation!  
exit  
Are you sure to exit? Print[Y] to exit, otherwise will return  
Y  
  
Process finished with exit code 0
```

【Test case #20】退出语句 2（取消退出）

Input:

exit

QWER

x=3

x+3

Output:

Are you sure to exit? Print[Y] to exit, otherwise will return

```
-----  
Please enjoy your calculation!  
6
```

Screen-short for case #20:

```
-----  
Please enjoy your calculation!  
exit  
Are you sure to exit? Print[Y] to exit, otherwise will return  
QWER  
-----  
Please enjoy your calculation!  
x=3  
x+3  
6
```

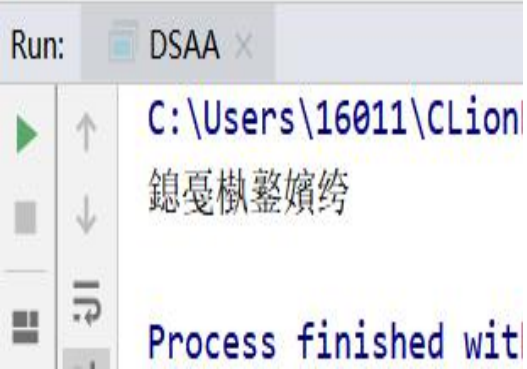
Problem8: 支持中英双语两种语言供用户选择

C++语言由于编码规则不同问题，在控制台上直接打印中文会显示奇奇怪怪的乱码，如下所示：

```
#include <iostream>

using namespace std;

int main() {
    cout<<"我是王奕童"<<endl;
    return 0;
}
```



而这对于提升用户的体验感不利;而本计算器适用的人群应当包括英语水平没有那么好的用户, 因此可以通过修改控制台编码来实现中文、英文两种语言的切换。

参考链接: <https://www.cnblogs.com/roadwide/p/10533594.html>

实现思路:

①先修改控制台编码:

SetConsoleOutputCP (65001);

②使用一个全局 bool 变量来记录当前的语言状态, 根据不同的语言状态调用不同的方法来输出信息。

【Test case #21】中文显示信息

Input:

Chinese
welcome
help
exit
R
x+3
1_x=3
English
PI=7

Output:

(英文版本的欢迎界面)

中文设置成功!

(中文版本的欢迎界面)

(中文版本的帮助说明)

(中文版本的退出语句)

开始运算吧!

(中文提示变量不存在)

(中文提示变量不合法)

English is set successfully!

(英文提示不可修改常量)

Screen-shorts for case #21:

C:\Users\16011\CLionProjects\cppLearn\cmake-build-debug\cppLearn.exe

Welcome using this simple calculator!
This calculator is mainly for the Assignment2 in CS205(C/C++) course.
This calculator is much better than that in Assignment1!
Course Instructor: Shiqi Yu
Author: YeeTone Wang
Version: 1.06
If you need help, please print help to get some information.
If you need welcome, please print welcome to get this information again.
If you need exit, please print exit to get exit information.
如果需要显示中文, 请输入Chinese

Please enjoy your calculation!

Chinese

中文设置成功!

welcome

欢迎使用本计算器!
本计算器主要是用于南方科技大学CS205(C/C++程序设计)课程的第二次作业。
本计算器实现的功能比第一次作业要强大很多!
课程教师: 于仕琪
作者: 王奕童
版本号: 1.06
如果需要帮助, 则请在屏幕上输入help
如果需要欢迎, 则请在屏幕上输入welcome
如果需要退出, 则请在屏幕上输入exit
If you need to show in English, please print English

开始运算吧！

help

这是对于计算器用户的使用说明

作者：王奕童

版本：1.06

作者首先很感谢ACBingo的一篇博客

这个计算器是基于"CWTNumber.h"头文件实现的

原链接：<https://www.cnblogs.com/acbingo/p/4116206.html>

0. 计算器中的一些常数：

E=2.718281828459

PI=3.1415926535898

计算器中支持的功能

1. 基本数学运算：+ - * /

示例-1.1：

3*4

12

示例-1.2：

3.45*4.32

14.904

2. 带括号的运算：()

示例-2.1：

(2+5)*9.9

该计算器仍然处于改进当中。由于时间的限制，现暂时只支持这么多功能。

exit

确认退出吗？在屏幕中输入[Y]则退出，否则将返回

R

开始运算吧！

x+3

变量不存在！

1_x=3

变量名不合法！

English

English is set successfully!

PI=7

This is constant value. Cannot be changed!

【总结】本次 Assignment2 看起来难度较高，实际上考察的是多方面的综合能力：

1. c++的 string 类的处理操作；
2. 类和对象的设计与使用；
3. 静态和内联成员的声明与使用；
4. c++中 stack 和 vector 数据类型的使用；
5. 异常情形的识别与处理；
6. 修改控制台的编码格式；
7. 编码能力，改 bug 的基本能力，以及细心处理问题的能力；