**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.

图片包含 笔记本, 灯光, 电脑, 钟表

描述已自动生成

1. Use parentheses to enforce the priorities

图片包含 游戏机

描述已自动生成

1. Variables can be defined as follows



1. Some math functions can be supported

图片包含 游戏机

描述已自动生成

1. It can support arbitrary precision.

手机屏幕的截图

描述已自动生成

1. 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>
2. 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.)

1. 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. You 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.
2. 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.

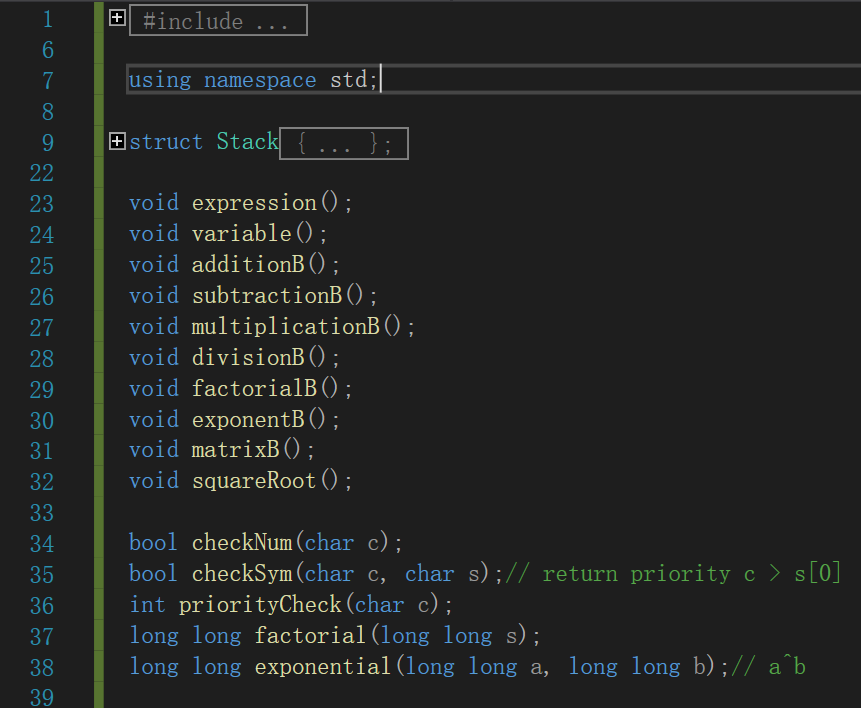
**Report Template:**

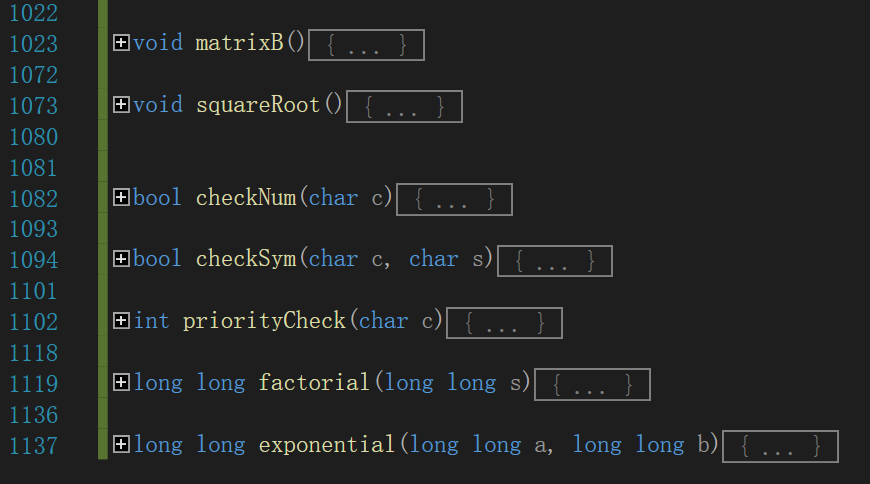
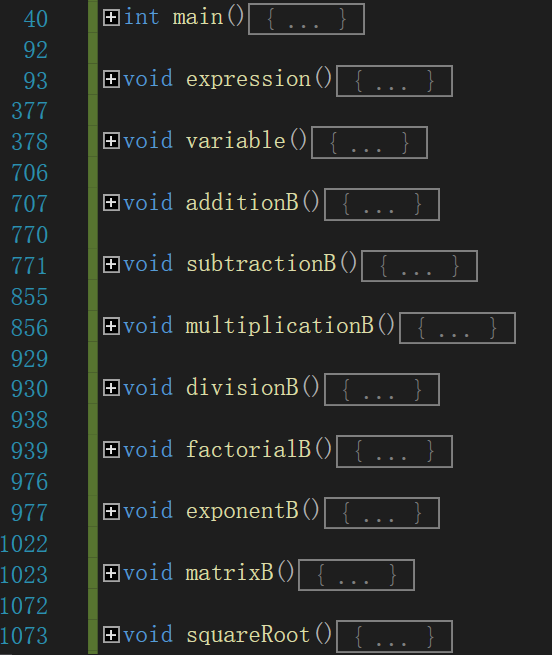
**CS205 C/ C++ Program Design**

**Assignment 2**

**Name**: 巫晓, **SID**: 11912803

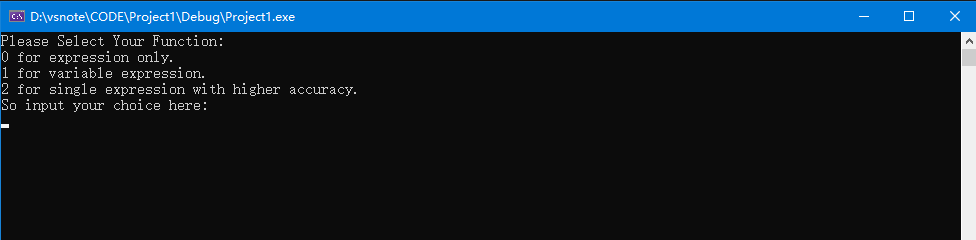
## Part 1. Life is short, show me the code.





**详细部分见附件cpp.**

因为要实现更多的功能，所以我做成了一个文字版的计算器，并分成了许多部分来进行实现，并让使用者自己选择想要的功能。



## Part 2. Result & Verification

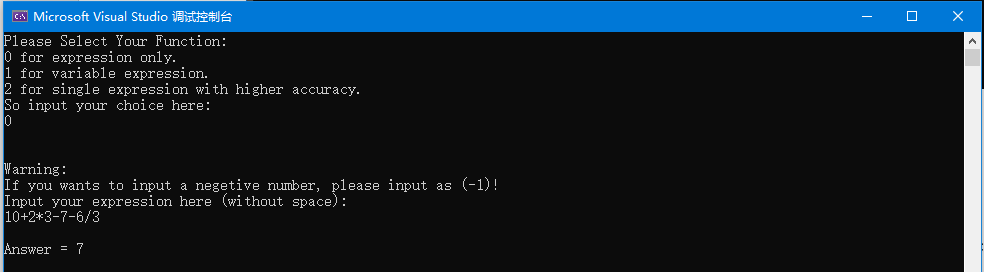
In this part, you should present the result of your program by listing the output of test cases and optionally add a screen-shot of the result.

Test case #1:（加减乘除四则运算）

10+2\*3-7-6/3

结果应为 7

Screen-short for case #1:

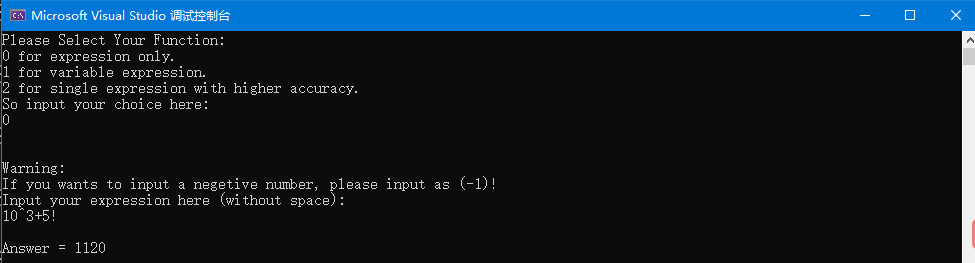


Test case #2:（乘方、阶乘运算）

10^3+5!

结果应为 1120

Screen-short for case #2:

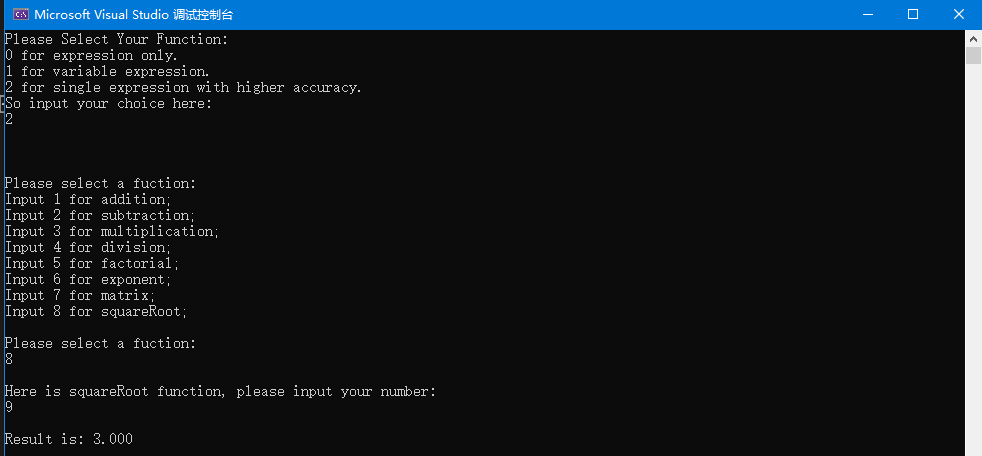


Test case #3:（求平方根）

9

结果应为 3.000

Screen-short for case #3:

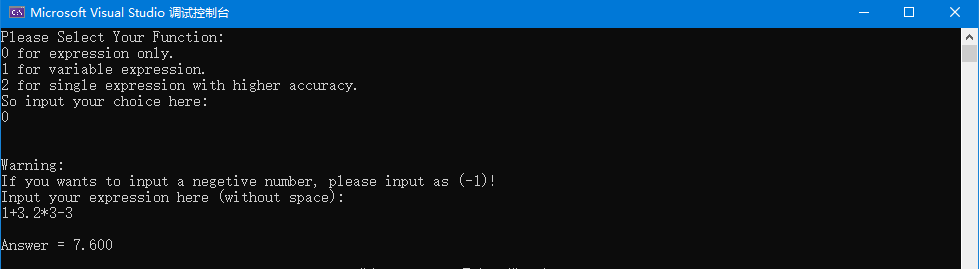


Test case #4:（整数和小数混杂的表达式）

1+3.2\*3-3

结果应为 7.600

Screen-short for case #4:

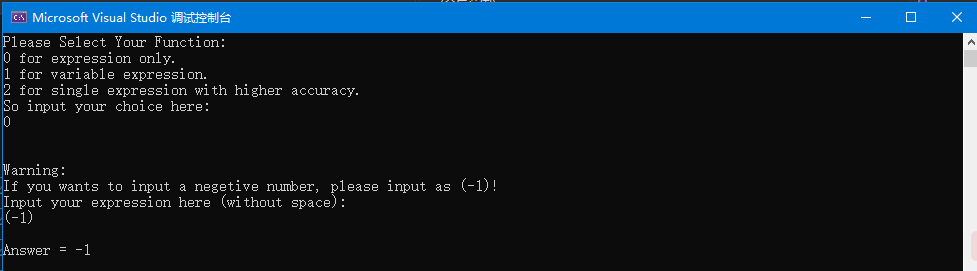


Test case #5:（有负数的情况）

（-1）

结果应为 -1

Screen-short for case #5:

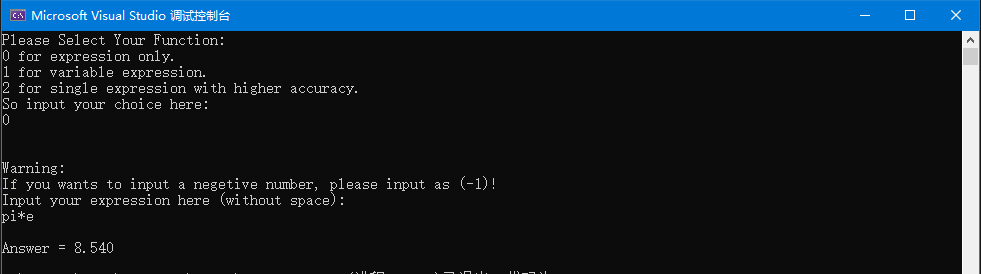


Test case #6:（pi和e的情况）

Pi \* e

结果应为 8.540

Screen-short for case #6:

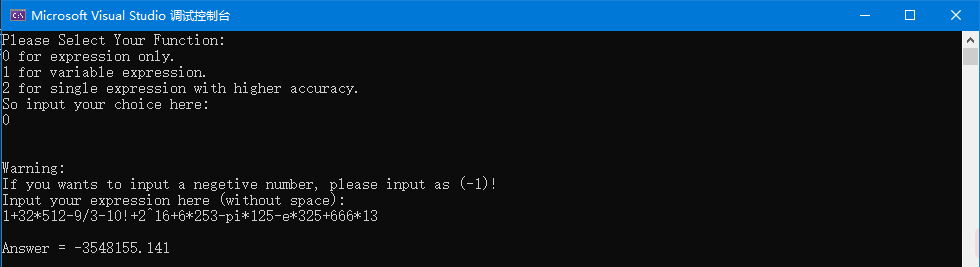


Test case #7:（很长的表达式）

1+32\*512-9/3-10!+2^16+6\*253-pi\*125-e\*325+666\*13

结果应为 -3548155.141

Screen-short for case #7:

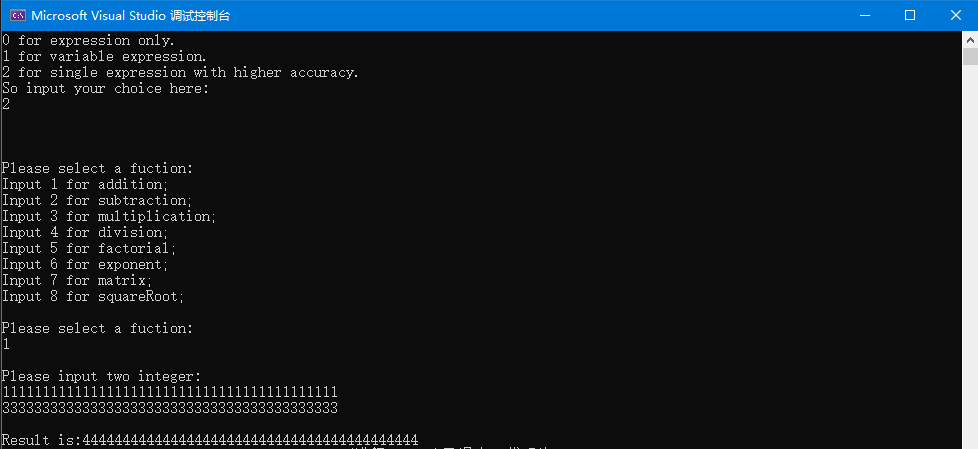


Test case #8:（高精度加法）

111111111111111111111111111111111111111111 333333333333333333333333333333333333333333

结果应为 44444444444444444444444444444444444444

Screen-short for case #8:



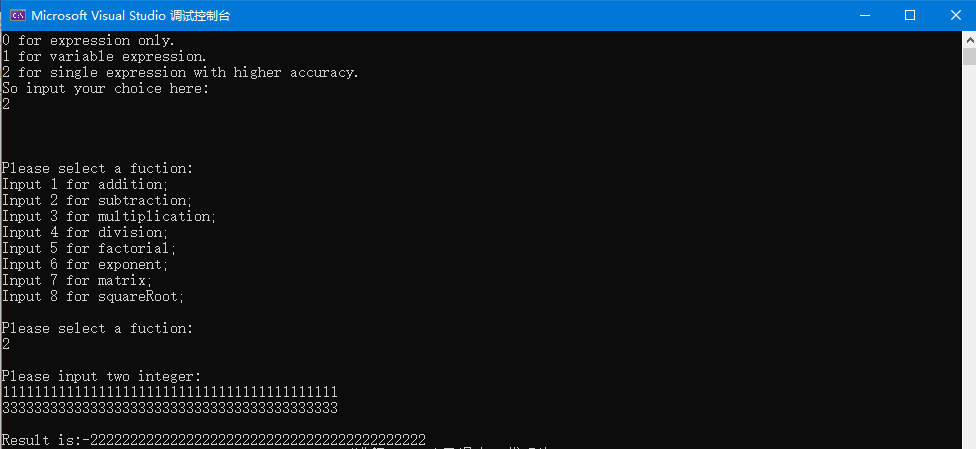
Test case #9: （高精度减法）

111111111111111111111111111111111111111111

333333333333333333333333333333333333333333

结果应为 -2222222222222222222222222222222222222222

Screen-short for case #9:

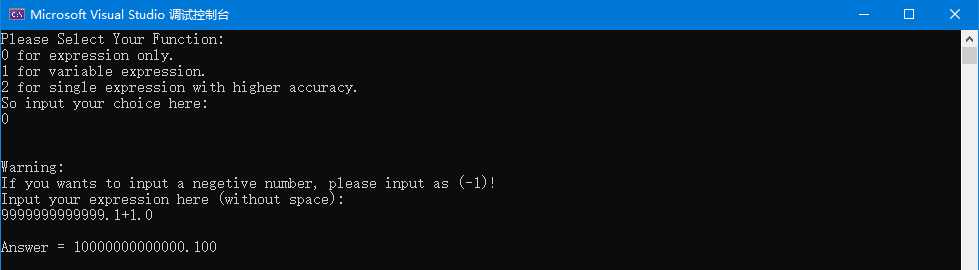


Test case #10:（小数高精）

9999999999999.1+1.0

结果应为10000000000000.100

Screen-short for case #10:



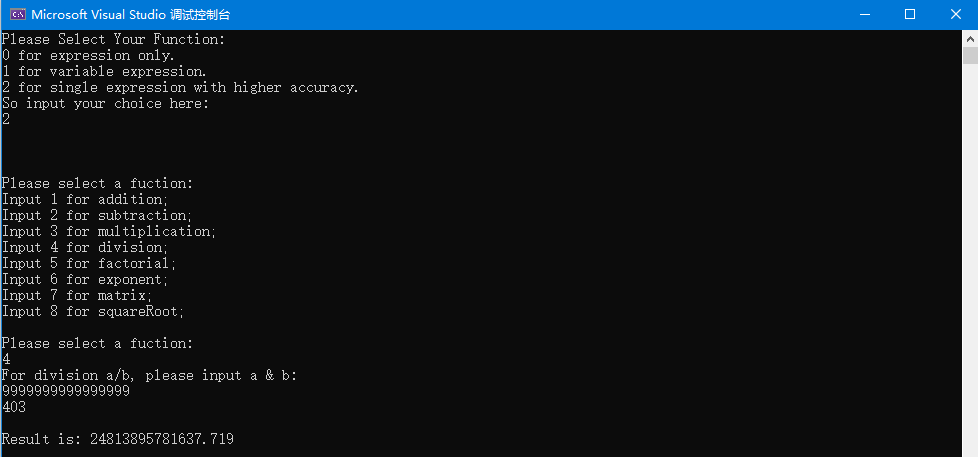
Test case #11:（高精度除法）

9999999999999999

403

结果应为 24813895781637.719

Screen-short for case #11:



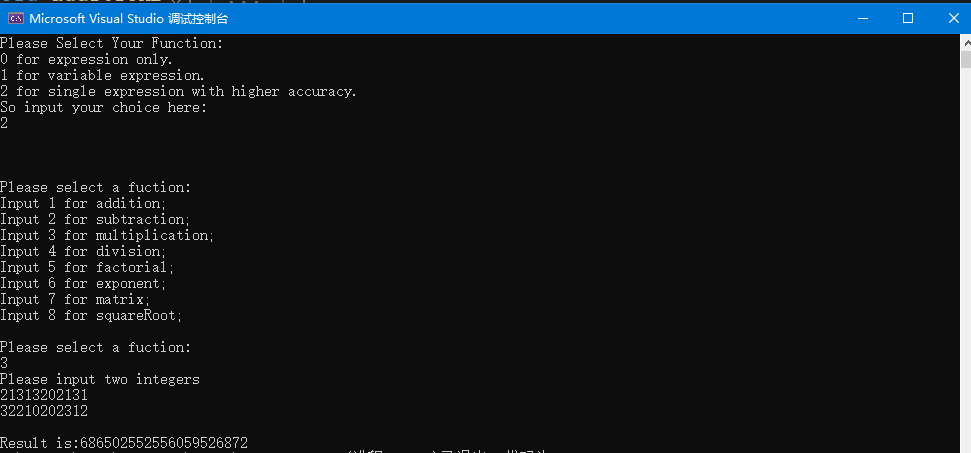
Test case #12:（高精度乘法）

21313202131

32210202312

结果应为 686502552556059526872

Screen-short for case #12:



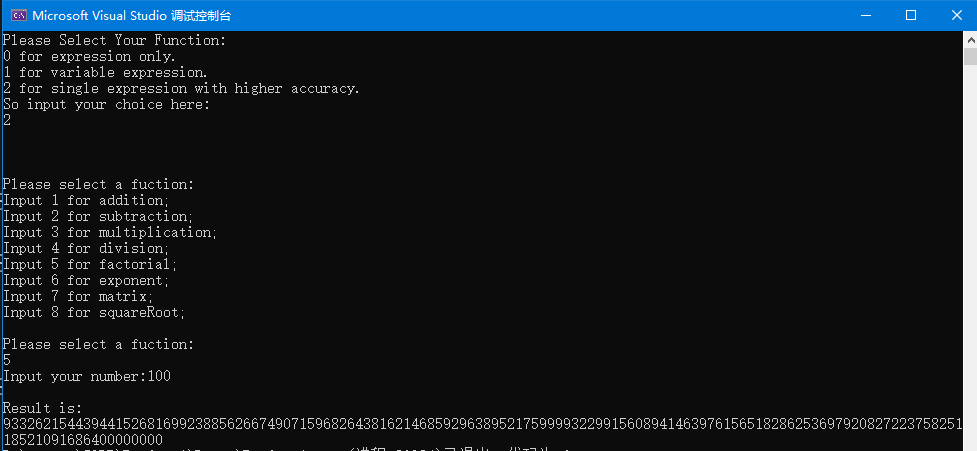
Test case #13:（高精度阶乘）

100

结果应为

93326215443944152681699238856266749071596826438162146859296389521759999322991560894146397615651828625369792082722375825118521091686400000000

Screen-short for case #13:



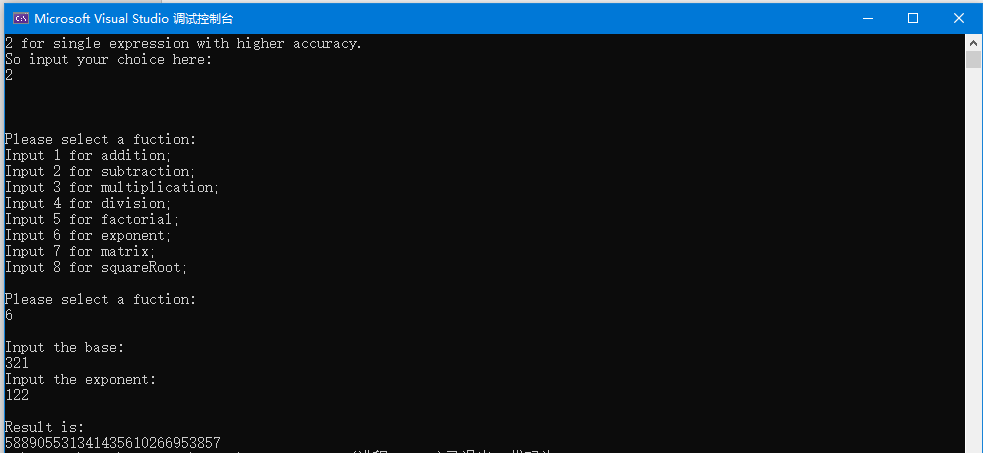
Test case #14:（高精度指数）

321

122

结果应为 588905531341435610266953857

Screen-short for case #14:



Test case #15:（矩阵乘法）

3 2 2 3

1 2 7 8 9

3 4 8 7 6

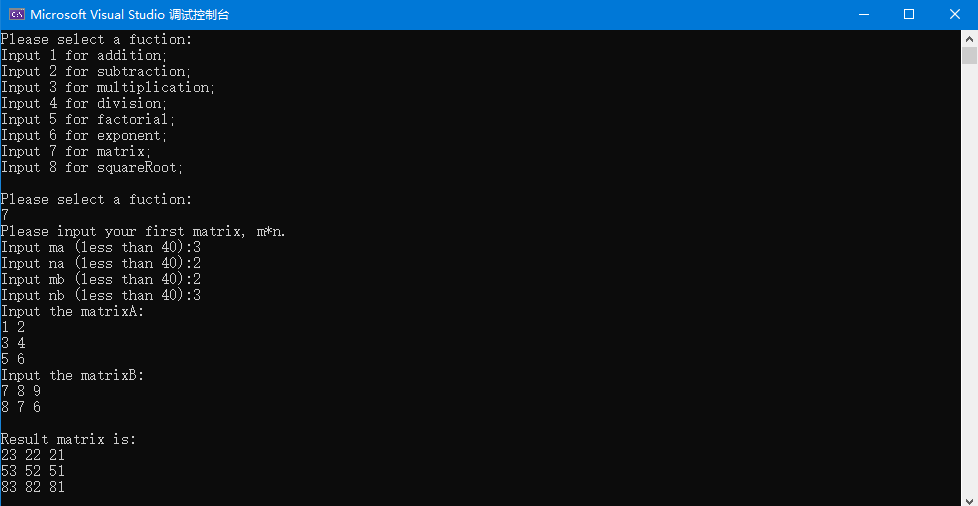
5 6

结果应为 23 22 21

53 52 51

83 82 81

Screen-short for case #15:

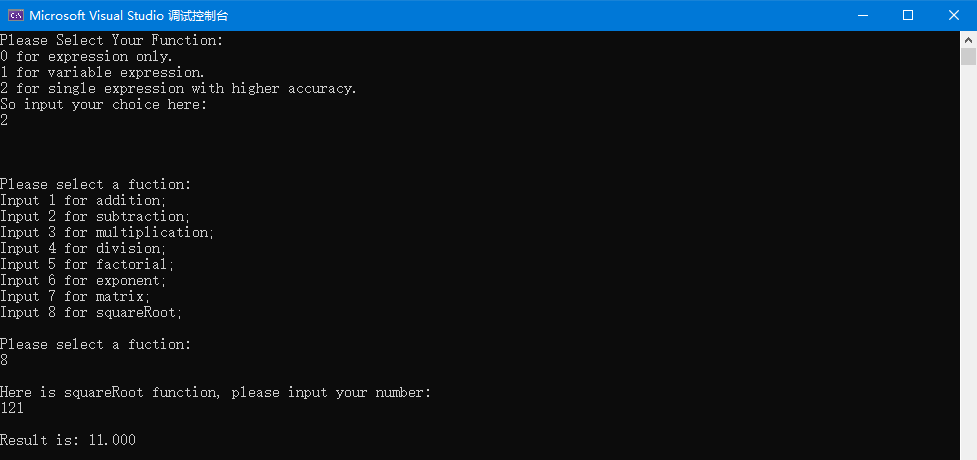


Test case #16:（求平方根）

121

结果应为 11.000

Screen-short for case #16:



Test case #17:（变量表达式）

x = 1

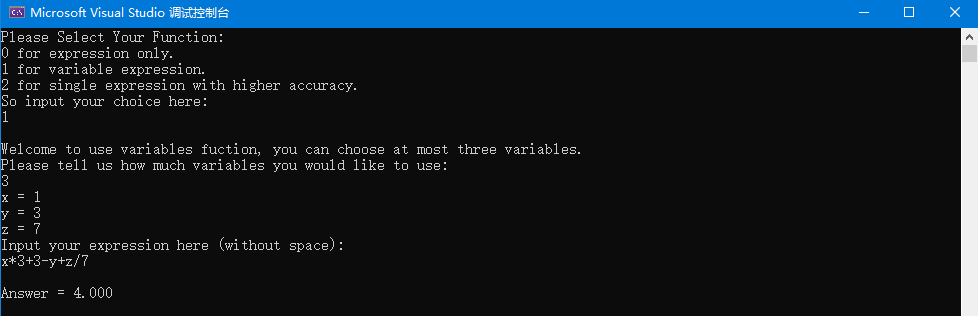
y = 3

z = 7

x\*3+3-y+z/7

结果应为 4.000

Screen-short for case #17:



## Part 3. Difficulties & Solutions, or others

1. 难点：对于输入的中缀表达式的处理

解决：中缀转后缀方便计算

1. 难点：表达式中存在小数整数混杂的情况

解决：用结构体储存，分别判断并储存

1. 难点：表达式中出现负数

解决：规定输入例如（-1）

1. 难点：符号的判定以及pi、e等的识别，还有符号优先级的验证

解决：单独写一个函数进行判断

1. 难点：高精度方法中，函数不能开过大的数组

解决：用压位方法

1. 难点：幂函数运算慢

解决：使用快速幂方法

1. 难点：表达式中夹杂未知变量

解决：定义未知量并替换