CS205 C/C++ Programming - Project 2

Name: 钟元吉(Zhong Yuanji)

SID: 12012613

Part 1 - Analysis

> This project to implement a much better calculator than that in the last project, which can support addition, substaction, prodction, division and other math function

Part 2 - Code

由于代码较长，这里仅放置输入字符串的分析部分，其他函数简介请参考源文件`Main.cpp`或函数头文件`FunHead.hpp`，结构体简介请参考函数头文件`StructHead.hpp`

```
// Analyse function for input string
void analyse(char *str)
{
    // Variables about input string
    int len = strlen(str), Id, lastId = 0;
    int dot = -1, e = -1;
    bool isNum = false, numEnd = false;
    // Variables about calculate stack
    numPt = -1, opPt = -1;
    numSet = new Number[len];
    opSet = new char[len];
    priorSet = new char[len];
    // Analyse every char of input
    for (Id = 0; Id < len && !error[0]; ++Id)
    {
```

```
        char now = str[Id];
        // It is not a number char
        if (now < '0' || now > '9')
        {
            bool cmp = false;
            // Compare to variables name
            for (int j = 0; j <= varId; ++j)
                if (strncasecmp(&(str[Id]), varName[j], strlen(varName[j])) == 0)
                {
                    addNumBVar;
                    numSet[++numPt] = varVal[j].copy();
                    Id += strlen(varName[j]) - 1;
                    goto NEXT_CHAR;
                }
            // Compare to functions name
            for (int j = 0; j < 12; ++j)
                if (strncasecmp(&(str[Id]), FUN_STR[j], strlen(FUN_STR[j])) == 0)
                {
                    addNumBVar;
                    isNum = false;
                    addOp(j, 5);
                    Id += strlen(FUN_STR[j]) - 1;
                    goto NEXT_CHAR;
                }
            // Compare to operators name
            for (int j = 0; j < 8 && !cmp; ++j)
                if (now == OP_SG[j])
                {
                    // '(' is special
                    if (now == '(')
                    {
                        addNumBVar;
                        isNum = false;
                    }
                    else
                    {
                        addNumBOp;
                        if (now == ')')
                            isNum = true, numEnd = true, str[Id] = ')';
                    }
                    addOp(now, OP_PRIOR[j]);
                    goto NEXT_CHAR;
                }
            // Compare to pi
            if ((now == 'p' || now == 'P') && (str[Id + 1] == 'i' || str[Id + 1]
== 'I'))
            {
                addNumBVar;
                ++Id;
                numSet[++numPt] = constPi;
                goto NEXT_CHAR;
            }
            // Compare to 'e' / 'E'
            if (now == 'e' || now == 'E')
            {
```

```
                if (Id == 0 || str[Id - 1] == '\0' || str[Id - 1] == ')' ||
(str[Id + 1] != '-' && (str[Id + 1] < '0' || str[Id + 1] > '9')))
                {
                    addNumBVar;
                    numSet[++numPt] = constE;
                    str[Id] = '0';
                }
                else if (e >= 0)
                {
                    error[0] = 11, error[1] = Id;
                    return;
                }
                else
                    e = Id - lastId;
                goto NEXT_CHAR;
            } // Compare to '-'
            if (now == '-')
            {
                // As a minus signal
                if (Id != 0 && str[Id - 1] != '\0' && str[Id - 1] != 'e')
                {
                    addNumBOp;
                    addOp('-', 1);
                }
                // As an oppsite signal
                else if (!isNum)
                {
                    isNum = true, numEnd = false, lastId = Id;
                    dot = -1, e = -1;
                }
                goto NEXT_CHAR;
            } // Compare to '.'
            if (now == '.')
            {
                // Check if more than one dots or float exponent
                if (dot >= 0 || e >= 0)
                {
                    error[0] = 9 + (e >= 0), error[1] = Id;
                    return;
                }
                if (!isNum)
                    isNum = true, lastId = Id, e = -1;
                dot = Id - lastId;
                goto NEXT_CHAR;
            }
            // There is invalid char in the input
            error[0] = 4, error[1] = Id;
            return;
        NEXT_CHAR:
            continue;
        }
        // It is a number char
        else
        {
            // Last char is not number
```

```
            if (!isNum)
                lastId = Id, dot = -1, e = -1;
            else if (numEnd)
            {
                error[0] = 7, error[1] = Id;
                return;
            }
            isNum = true;
            numEnd = false;
        }
    }
    // Pick up the Number and finish the rest calculate in the end
    if (isNum && !numEnd)
        addstrNum(&(str[lastId]), dot, e);
    while (opPt >= 0 && priorSet[opPt] >= 0 && !error[0])
        calculate();
    // Save answer
    if (numPt == 0)
    {
        varVal[0].del();
        varVal[0] = numSet[numPt--];
    }
    else
        error[0] = 6;
    return;
}
```

Part 3 - Result & Verification

Test case #1: 基本要求的实现

注：程序中统一采用科学计数法进行输出

```
Input: cmake . & make
Input: ./Project2.out
Input: 2+3
Output: 5
```

```
Input: 5+2*3
Output: 1.1e1
```

```
Input: (5+2)*3
Output: 2.1e1
```

```
Input: x=3
Output: 3
Input: y=6
Output: 6
Input: x+2*y
Output: 1.5e1
```

```
Input: sqrt(3.0)
Output: 1.73205080756887729366
```

注：第一个数包含30个9和10个2

```
Input: 99999999999999999999999999999.2222222222 + 1.0
Output: 1.00000000000000000000000000000002222222222e30
```

```
Consolidate compiler generated dependencies of target Project2.out
-- Generating done
-- Build files have been written to: /mnt/d/VScodeProjects/CppClass
[100%] Built target Project2.out
[3]-  Done                    cmake .
root@LAPTOP-JKBUNEI2:/mnt/d/VScodeProjects/CppClass# ./Project2.out
Please input the expression in the next line: (quit:q)
2+3
5

Please input the expression in the next line: (quit:q)
5+2*3
1.1e1

Please input the expression in the next line: (quit:q)
(5+2)*3
2.1e1

Please input the expression in the next line: (quit:q)
x=3
3

Please input the expression in the next line: (quit:q)
y=6
6

Please input the expression in the next line: (quit:q)
x+2*y
1.5e1

Please input the expression in the next line: (quit:q)
sqrt(3.0)
1.73205080756887729366

Please input the expression in the next line: (quit:q)
99999999999999999999999999999.2222222222 + 1.0
1.00000000000000000000000000000002222222222e30
```

Test case #2: 对错误输入的判断

对于**错误的具体位置**，程序会在告知错误前标明。

格式错误：

```
Input:
Output: There is no input. Please try again.
```

```
Input: =2+3
Output: The input cannot start with equal sign. Please try again.
```

```
Input: 3a=5-2
Output: The variable name on the left of equal sign is invalid. Please try again.
```

```
Input: 3+4_3
Output: There is invalid char in the input. Please try again.
```

```
Input: (3+4))
Output: The number of '(' and ')' are different. Please try again.
```

```
Input: pi2
Output: The numbers cannot follow a variable. Please try again.
```

```
Input: 2//2
Output: The operator cannot follow a operator. Please try again.
```

```
Input: *5
Output: The input starts with a operator. Please try again.
```

```
Please input the expression in the next line: (quit:q)


^~~~~~  ("" is invalid)
There is no input. Please try again.
Please input the expression in the next line: (quit:q)
=2+3
=2+3
^~~~~~  ("=" is invalid)
The input cannot start with equal sign. Please try again.
Please input the expression in the next line: (quit:q)
3a=5-2
3a=5-2
^~~~~~  ("3" is invalid)
The variable name on the left of equal sign is invalid. Please try again.

Please input the expression in the next line: (quit:q)
3+4_3
3+4_3
   ^~~~~~  ("_" is invalid)
There is invalid char in the input. Please try again.

Please input the expression in the next line: (quit:q)
(3+4))
(3+4))
^~~~~~  ("(" is invalid)
The number of '(' and ')' are different. Please try again.

Please input the expression in the next line: (quit:q)
pi2
pi2
  ^~~~~~  ("2" is invalid)
The numbers cannot follow a variable. Please try again.

Please input the expression in the next line: (quit:q)
2//2
2//2
  ^~~~~~  ("/" is invalid)
The operator cannot follow a operator. Please try again.

Please input the expression in the next line: (quit:q)
*5
*5
^~~~~~  ("*" is invalid)
The input starts with a operator. Please try again.

Please input the expression in the next line: (quit:q)
```

Test case #3: 对错误输入的判断

数字输入错误:

```
Input: 1..2
Output: There are more than one dots in a number. Please try again.
```

```
Input: 1e3.2
Output: The exponene cannot be a float. Please try again.
```

计算错误：

```
Input: 1/(pi-pi)
Output: The divider cannot be zero. Please try again.
```

先使用`ans = 2`初始化，然后重复进行以下运算：

```
Input: ans = 2 ^ ans
...
Input: ans = 2 ^ ans
Output: There might be inf or nan in the expression. Please try again.
```

```
Please input the expression in the next line: (quit:q)
1..2
1..2
There are more than one dots in a number. Please try again.

Please input the expression in the next line: (quit:q)
1e3.2
1e3.2
   ^~~~~~  ("." is invalid)
The exponene cannot be a float. Please try again.

Please input the expression in the next line: (quit:q)
1/(pi-pi)
1/(pi-pi)
The divider cannot be zero. Please try again.

Please input the expression in the next line: (quit:q)
ans = 2
2

Please input the expression in the next line: (quit:q)
ans = 2 ^ ans
4

Please input the expression in the next line: (quit:q)
ans = 2 ^ ans
1.6e1

Please input the expression in the next line: (quit:q)
ans = 2 ^ ans
6.5536e4

Please input the expression in the next line: (quit:q)
ans = 2 ^ ans
2.0035299304068464649790723515602557504478254755697514192650169737108940595563114530895061308809333481010382343429072631818229493
8211881266886950636476154702916504187191635158796634721944293092798208430910485599057015931895963952486337236720300291696959215610
8764948889254090805911457037675208500206671563702366126359747144807111774815880914135742720967190151836282560618091458852699826141
4142503012339110827360384376787644904320596037912449090570756031403507616256247603186379312648470374378295497561377098160461441330
8692118102485959152380195331030292162800160568670105651646750568038741529463842244845292537361442533614373729088303794601274724958
5841486491593064725201515569392262818069165079638106413227530726714399815850881129262890113423778270556742108007006528396332215500
7783121428855167555407334510721311242739956298271976915005488390522380435704584819795639315785351001899200002414196370681355984040
6403947219401606951769015611972698233789001764151719005113346630689814021938348143542638730653955296969138802415816185956110064030
6211979610185953480278716720012260464249238511139340046435162386756707874525946467090388654774348321789701276445552940909202195950
8575162297333357615955239488529757995402847194352991354376370598692891375715374000198639433246489005254310662966916524341917469130
8963247656028941519977547770313806478134230959619096065459130089018888758808473362595606544488850144733570605881709016210849971450
2956834406197969056546981363116205357936979140323632849623304642106613620022017564e19728

Please input the expression in the next line: (quit:q)
ans = 2 ^ ans
ans=2^ans
    ^~~~~~  ("2" is invalid)
There might be inf or nan in the expression. Please try again.

Please input the expression in the next line: (quit:q)
```

Test case #4: 程序特色功能

具体实现功能参考 README.md

1. 函数中**可以不使用括号**，常数`pi`与`e`有100位小数的精度

    ```
    Input: cos pi
    Output: -1
    ```

```
Input: log e
Output: 1
```

```
Input: floor 2.5
Output: 2
```

```
Input: asin(sin1)
Output: 1
```

2. 数字与(参数、函数)、(参数、函数)与(参数、函数)之间的乘法**可以不写乘号**（但**数字与数字之间的乘号不可省略**）

```
Input: pipi
Output:
9.869604401089358618834490999876151135313699407240790626413349376220044822419
205243001773403718552231307874263580850209166609835428373261595226022618170033
881496242667944434304721741925322332314849321041
```

```
Input: 1e10e (指1*10^10*e)
Output:
2.718281828459045235360287471352662497757247093699959574966967627724076630353
547594571382178525166427 4e10
```

3. 答案储存

```
Input: ans/10 (与之前的结果有关)
Output:
2.718281828459045235360287471352662497757247093699959574966967627724076630353
547594571382178525166427 4e9
```

```
Please input the expression in the next line: (quit:q)
cos pi
-1

Please input the expression in the next line: (quit:q)
log e
1

Please input the expression in the next line: (quit:q)
floor 2.5
2

Please input the expression in the next line: (quit:q)
asin(sin1)
1

Please input the expression in the next line: (quit:q)
pipi
9.869604401089358618834490999876151135313699407240790626413349376220044822419205243001773403718552231307874263580850209166609835428373261595226022618170338814962426679444343047217419253223323148493210 41

Please input the expression in the next line: (quit:q)
1e10e
2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427 4e10

Please input the expression in the next line: (quit:q)
ans/10
2.718281828459045235360287471352662497757247093699959574966967627724076630353547594571382178525166427 4e9
```

Test case #6: 快速幂与大数高精度运算

```
Input: 1.01^365
Output:
3.778343433288715887761660479649760546027113549159100200330393389369444295219859381193563943688913875294723025746665296695026293779874517233301507922233862428614682541680615253144396919455694277651724794006295820217560495780683332054961828376032992078447444074823282352284877477666337709851763425891809224927535504775170910970056315161670685632917067996914303111984143610198730361066503225373596290071532034477267109474634224398074728853774804481080543151365628472837715086072554406951570418030966946107155062721625508320095968055876732999773925642529901823096810818379078283445112234139169902672871880967067586849494180180014804369532225491871467707211395504215731052494540132169947984320082714253087130288973011802510504401943365501e1
```

```
Input: floor ans
Output: 3.7e1
```

```
Input: (1/3)^900000
Output:
7.425954840255219139971991307294138739458749943515449116605464810222732637481773307530219761268777094562702637459574775924 4...e-429410 (输出中共1500位有效数字)
```

```
Input: (1e10000-1)(1e10000-1)
Output: 9.9999999999...9999998000000...0001e19999 (输出中共19999位有效数字)
```

```
Input: ans-1+2e10000
Output: 1e20000
```

```
Please input the expression in the next line: (quit:q)
1.01^365
3.7783434332887158877616604796497605460271135491591002003303933893694442952198593811935639436889138752947230257466652966950262937
798745172333015079222338624286146825416806152531443969194556942776517247940062958202175604957806833320549618283760329920784474440
748232823522848774776663377098517634258918092249275355047751709109700563151616706856329170679969143031119841436101987303610665032
253735962900715320344772671094746342243980747288537748044810805431513656284728377150860725544069515704180309669461071550627216255
083200959680558767329997739256425299018230968108183790782834451122341391699026728718809670675868494941801800148043695322254918714
67707211395504215731052494540132169947984320082714253087130288973011802510504401943365 01e1

Please input the expression in the next line: (quit:q)
floor ans
3.7e1

Please input the expression in the next line: (quit:q)
(1/3)^900000
7.4259548402552191399719913072941387394587499435154491166054648102227326374817733075302197612687770945627026374595747759244229053
664313122922430097459433323978559874441798150161463363150199475456203974565591899872384561686961678962988021682679221655552159504
194168545517721049216397883551289365524750107351504932218823015736369746493985933529844184218645685571817711011454328869488018987
470071591232609020901973884257490334119045415882713056024943401112139712427695996517848038082297798154771050398059557681654976583
707345494311820698999397009192763227747385336962946214970438750205505754131518335259754782471315169273501071815228775144976661674
532126449830614884211503301327770701718837110347655833314351440682224511979572668306161494354400806331134157110752813724807436847
501578877611358637037499082671641851451011989145654025065613105579253334427064256189090720674720738411217237469112057082017089468
684615235019903923532117505241647638947122195337431561372119477011970027606137968430211790413624032467525105695127793307197134764
526574306522964441916782752951961105560066808289275331379481968734559294660318610708627163746977771029208413764294587011137406044
110827812788993745386646657062145607890489330316574802233571316306922517311336688478118551639849502499056011697322257866366807437
162532055816207804450025137854368965443856558193979228652736106275463667325362900438432136852804292985043578493320222335920594457
2212072363782751602738540400169078896405378149253060264401912169836530883343132036e-429410

Please input the expression in the next line: (quit:q)
(1e10000-1)(1e10000-1)
9.9999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999998000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000001e19999

Please input the expression in the next line: (quit:q)
ans-1+2e10000
1e20000

Please input the expression in the next line: (quit:q)
```

注：中间部分的部分9和0被长截图拼合时吞了

Part 4 - Difficulties & Solutions

问题1：

Part 5 - Summary

以上是本次Report的所有内容，感谢您的阅读！