Section 1: Pop Quiz

(10 minutes)

A series of terms will be presented to you, and for each term you will be given 60 seconds to write a brief response explaining the meaning or significance of the term.

For example you might be prompted HTML, and your description might be: markup language for rendering a web page

Do not worry if you come across a term you are unfamiliar with. For unfamiliar terms, feel free to use word association. For example, if you are prompted MySQL, you can write some data storage

[Done!](https://www.oneinterview.io/interviews/0279182915/question)

Section 2: Coding Challenge

(50 minutes)

The challenge can be taken in your local environment or our online editor.

* [Use Local Environment](https://www.oneinterview.io/interviews/0279182915#offline-instruction-tab)
* [Use Online Editor](https://www.oneinterview.io/interviews/0279182915#online-editor-tab)

**Calculator**

**Problem**

Given a valid string of math expressions, such as 3.1 + 4 \* 5, implement a calculator. For the sake of simplicity, you can assume all operations will be passed valid parameters. You do not need to concern yourself with error handling for occurrences like non-mathematical symbol inputs.

First Step

Please implement Calc to support double and mathematical operators such as +, -, \*, and /. Mathematical operators will be delimited by a space. Please note:

* Input numbers are of the double type and can be negative.
* Operators \* and / has higher precedence than + and -.

For example,

[1](https://www.oneinterview.io/interviews/0279182915" \l "n1)// should return 23

[2](https://www.oneinterview.io/interviews/0279182915" \l "n2)calculator.Calc("3 + 4 \* 5")

[3](https://www.oneinterview.io/interviews/0279182915" \l "n3)// should return 5.857142

[4](https://www.oneinterview.io/interviews/0279182915" \l "n4)calculator.Calc("3 + 4 \* 5 / 7")

Second Step

Please update Calc to support the use of parenthesis (), where contents within the parenthesis are prioritized during computation.

[1](https://www.oneinterview.io/interviews/0279182915#n1)// should return a 5.0

[2](https://www.oneinterview.io/interviews/0279182915#n2)calculator.Calc("( 3 + 4 ) \* 5 / 7")

Third Step

Please implement CalcWithVars. Enable the instance method to accept a list of strings representing assignment statements. The left-hand side should be the variable and the right-hand side should be the mathematical expression that may contain previously defined variables. Variables are in the form [a-z]+, and there will be spaces around the equal sign =.

[1](https://www.oneinterview.io/interviews/0279182915#n1)// should return [3, 243]. 3 for "pi" and 243 for "9 \* 9 \* 3".

[2](https://www.oneinterview.io/interviews/0279182915#n2)calculator.CalcWithVars([

[3](https://www.oneinterview.io/interviews/0279182915#n3) "pi = 3",

[4](https://www.oneinterview.io/interviews/0279182915#n4) "pizza = 9 \* 9 \* pi"])

#include <iostream>

#include <map>

#include <sstream>

#include <string>

#include <vector>

#include <stdlib.h>

using namespace std;

bool is\_number(const string& s) {

string::const\_iterator it = s.begin();

if (\*it == '-') {

++it;

}

if (it == s.end()) {

return false;

}

while (it != s.end() && (isdigit(\*it) || (\*it == '.'))) {

++it;

}

return !s.empty() && it == s.end();

}

double to\_number(const string& s) {

return atof(s.c\_str());

}

vector<string> split(const string &s, char delim) {

vector<string> elems;

stringstream ss(s);

string item;

while (getline(ss, item, delim)) {

elems.push\_back(item);

}

return elems;

}

class Calculator {

public:

double Calc(const string& input) const {

// IMPLEMENT ME

vector<string> times;

times = split(input,'\*');

double a = 1;

for(int i = 0; i < times.size();i++){

if(is\_number(times[i])){

a = a \* to\_number(times[i]);

}

else{

vector<string> divi;

}

}

for(int i = 0;i < input.size();i++){

if(input[i] =='+'){

double res = to\_number(input.substr(0,i)) + Calc(input.substr(i+1));

}

else if(input[i] =='-'){

double res = to\_number(input.substr(0,i)) - Calc(input.substr(i+1));

}

else if(input[i] =='\*'){

double res = to\_number(input.substr(0,i)) + Calc(input.substr(i+1));

}

}

return 0.0;

}

vector<double> CalcWithVars(const vector<string>& inputs) const {

// IMPLEMENT ME

vector<double> output;

return output;

}

};

#ifndef \_\_main\_\_

#define \_\_main\_\_

int main(int argc, char\* argv[]) {

Calculator calculator;

cout << "First Step" << endl;

// should print 5.85714

cout << calculator.Calc("3 + 4 \* 5 / 7") << endl;

cout << "\nSecond Step" << endl;

// should print a 5

cout << calculator.Calc("( 3 + 4 ) \* 5 / 7") << endl;

cout << "\nThird Step" << endl;

// should print 3 and 243. 3 for "pi" and 243 for "9 \* 9 \* 3".

vector<double> output = calculator.CalcWithVars(

{ "pi = 3", "pizza = 9 \* 9 \* pi" });

for (size\_t i = 0; i < output.size(); ++i) {

cout << output[i] << endl;

}

}