

第二次实验：关于我转生到异世界当老师这件事

Ver 1.1

by 穿越了的助教们

实验背景

助教因连续熬夜批改作业和审阅实验代码而死。在他醒来之后，发现自己成了一名异世界的小学数学教师。但是这个世界的运算规则很是诡异：运算符的优先级不同于我们原来所在世界的优先级。

优先级由高到低：

"()" = "abs()" > '-' (负号,单目运算符) > '+' = '-' (减号) > '*' = '/' > '^'

现在急需同学们写一个异世界计算器帮助可怜的助教。各位写的程序会通过oj发往异世界。

实验内容

为了运用堆栈的知识，本次实验以计算数值表达式为题材。

输入：只有一行，包含数字(1234567890)和运算符（包括+ - * / () ^ abs()）。不会出现汉字括号。

输出：只有一行，输出表达式计算出的结果。

要求与说明

-2. 蓝字是 1.1 版本更新的内容，修复了负号右侧对象的问题。

-1. 黄字是 1.0 版本更新的内容。黄字不代表本次实验的重点与难点。1.0 版本修复了负号说明不明确的 bug。

0. 实验背景是认真的。符号的优先级确实要发生改变。

1. abs()是取绝对值运算。+ - * / ^分别代表加、减、乘、除、乘方运算。

2. '-'可能会被用作单目运算符，此时的含义是取负。 '+'不会被用作单目运算符。

3. 计算顺序是从左到右计算。除负号之外，其余符号皆为左结合。

4. 表达式内不含空格和小数点。

5. 输入保证运算符"abs("这个字符串不会出错。abs 三个字母后面保证会紧跟一个左括号。abs()内保证会有完整的表达式。除法直接舍去小数点后的部分。每一步除法都要抛弃小数点。

6. 手残的助教可能会漏写若干个括号。如果左右括号失配，请直接输出 NULL 四个字母。

7. 可爱的小学生可能会给出计算中出现除数为 0 的式子，或 0^0 的式子。如果出现这两种式子，也请直接输出 NULL 四个字母。异世界的小学生不会给出其他种类的错误式子。

8. 栈等数据结构需要自己手写。不准使用 C++ 的 STL。

9. 为各位判断方便，我们规定负号只可能出现在左括号的右边，例如：(-1)、 $2^{(-1)}$ 、 $3^{*(-1)}$ 、 $7+(-2)$ 、 $5-(-10)$ 。为降低难度，我们规定负号右侧一定是数字。

10. 由于我们不会输入除第四条和第五条以外的错误表达式，因此你们无需判断输入是否非法。

输入输出样例

输入	输出
1+2-3*4/5^abs(6)	0
1+3*(5-7	NULL
1+4/1-1	NULL
-1	我们不会输入这种东西。因为负号左侧没有左括号。
(-1)	-1
(-1+5)	4
(-1*-6)	我们不会输入这种东西。因为负号左侧没有左括号。
(-1*6)	-6
(-1	NULL
(-12+12)	0

$32/(-1+1)$	NULL
$22+3)^3$	NULL
$23+(-32)*7/abs(-2^3)$	-7
(-6^4)	1296
$21^{(-3)}$	0
$abs((-2^3)/(-1+3))$	4
2^3^4	4096

除非实验要求更新，我们不会提供其他测试样例。

提示

1. 计算的中间结果和最终结果均在-2147483648~2147483647 之内。
2. 输入的表达式字符个数不会大于 200。
3. 以上是手残的助教目前想到的要求，也许还有稍许细则补充，会在群里通知。

Deadline

2019-10-27 23:59 (GMT+8:00)。这是最终确定的 deadline。

除非后续由老师说明，本次实验无实验报告。

提交方式

在 oj.ustc.edu.cn 上提交。预计在 2019-10-24 23:59 (GMT+8:00)前，我们会调试好 oj 供大家使用。届时我们会放出 oj 的使用指南。各位可以先注册账号，注册完成后请点击头像-profile-USTC ID 以绑定学号。各位还可以先从 oj 的 problemlist 里找几个简单的题（比如 1000）尝试使用 oj。

提示：在 C/C++ 下，oj 的输入输出用的是标准输入输出，即 gets/puts/getchar/putchar/scanf/printf/cin/cout 等屏幕输入。请勿使用 scanf_s。oj 使用的是 gcc/g++ 编译，使用的不是 Visual C 标准，因此不认这个函数。

VS 用户可以在你的 .c/.cpp 文件第一行写如下语句以消除安全警告：

```
#define _CRT_SECURE_NO_WARNINGS
```