

1356: 计算(calc)

题目描述

小明在你的帮助下，破密了Ferrari设的密码门，正要往前走，突然又出现了一个密码门，门上有一个算式，其中只有“(”，“)”，“0-9”，“+”，“-”，“*”，“/”，“^”，求出的值就是密码。小明数学学得不好，还需你帮他的忙。（“/”用整数除法）

输入格式

共1行，为一个算式。

输出格式

共1行，就是密码。

输入样例

$1+(3+2)*(7^2+6*9)/(2)$

输出样例

258

解析

本题的难度进一步提高，需要编写一个简易计算器。在这个计算器中，我们需要考虑括号、还有运算的优先级。

首先，我们需要两个栈结构，一个用来存储数据，一个用来存储运算符号。

当我们读入数字时，将其存储到数字栈，当我们读取到运算符时，则存储符号栈。如果此刻栈中的符号不为空，我们就需要判断一下栈中的符号是否为更高，如果是的话，则需要立刻计算，否则会出现错误，例如 $3 * 2 + 5$ 。

当我们读入第二个符号“+”时，如果不计算前面的 $3 * 2$ ，那么当我们最后处理计算的时候就会先执行 $2 + 5$ ，这是因为“+”号在栈顶。

另外，当读入括号时，我们需要立刻将括号内的式子计算完毕。

最后，我们再来计算括号外的式子。

编码

```
#include <bits/stdc++.h>
using namespace std;
stack<int> digit;
stack<char> symbol;
//获取符号的优先级, 数字越大优先级越高
int level(char c) {
    if (c == '+' || c == '-') {
        return 1;
    }
    if (c == '*' || c == '/') {
        return 2;
    }
    if (c == '^') {
        return 3;
    }
    return 0;
}

void calculation() {
    //取出左运算符
    int a = digit.top();
    digit.pop();
    //取出右运算符
    int b = digit.top();
    digit.pop();
    //取出运算符号
    char ch = symbol.top();
    symbol.pop();
    if (ch == '+') {
        digit.push(a + b);
    } else if (ch == '-') {
        digit.push(b - a);
    } else if (ch == '*') {
        digit.push(a * b);
    } else if (ch == '/') {
        digit.push(b / a);
    } else if (ch == '^') {
        digit.push(pow(b, a));
    }
}

int main() {
    string str;
    cin >> str;
    int len = str.length();
    int x = 0;
    bool tag = false;
    for (int i = 0; i < len; i++) {
        if (str[i] >= '0' && str[i] <= '9') {
```

```

        x = x * 10 + str[i] - '0';
        tag = true;
    } else {
        if (tag) {
            digit.push(x);
            tag = false;
            x = 0;
        }
        if (str[i] == '(') {
            symbol.push(str[i]);
            continue;
        }
        if (str[i] == ')') {
            //持续的，将括号内的运算全部完成
            while (symbol.top() != '(') {
                calculation();
            }
            //把另一侧的括号给删除
            symbol.pop();
            continue;
        }
        //处理读入中途的符号
        while (!symbol.empty() &&
            level(symbol.top()) >= level(str[i])) {
            calculation();
        }
        symbol.push(str[i]);
    }
}
if (tag) {
    digit.push(x);
}
//处理不带括号的运算
while (!symbol.empty()) {
    calculation();
}
//输出最终的结果
cout << digit.top() << endl;
return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

