

1358: 中缀表达式值(expr)

题目描述

输入一个中缀表达式（由0-9组成的运算数、加+减-乘*除/四种运算符、左右小括号组成。注意“-”也可作为负数的标志，表达式以“@”作为结束符），判断表达式是否合法，如果不合法，请输出“NO”；否则请把表达式转换成后缀形式，再求出后缀表达式的值并输出。

注意：必须用栈操作，不能直接输出表达式的值。

输入格式

一行为一个以@结束的字符串。

输出格式

如果表达式不合法，请输出“NO”，要求大写。

如果表达式合法，请输出计算结果。

输入样例

1+2*8-9@

输出样例

8

解析

本题与1356高度相仿，唯一的差别在于，我们需要校验等式是否合法。需要校验的地方包括括号是否匹配，数位是否正确对应等等。

编码

```
#include <bits/stdc++.h>

using namespace std;
stack<int> digit;
stack<char> symbol;
string str;

//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() {
    if (digit.empty() || symbol.empty() || digit.size() < 2) {
        printf("NO");
        exit(0);
    }
    //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 == '/') {
        if (a != 0) {
            digit.push(b / a);
        } else {
            printf("NO");
            exit(0);
        }
    } else if (ch == '^') {
        digit.push(pow(b, a));
    }
}

void rule() {
    int r = 0;
    for (int i = 0; str[i] != '@'; i++) {

```

```

        if (str[i] == '(') {
            r++;
        }
        //一上来就是一个右括号肯定错误
    else if (str[i] == ')') {
        if (r <= 0) {
            printf("NO");
            exit(0);
        } else {
            r--;
        }
    }
}
//括号数量不匹配
if (r != 0) {
    printf("NO");
    exit(0);
}
}

int main() {
    getline(cin, str);
    //检测括号匹配是否正确
    rule();
    int x = 0;
    bool tag = false;
    for (int i = 0; str[i] != '@'; 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。

