

## 1198：逆波兰表达式

### 题目描述

逆波兰表达式是一种把运算符前置的算术表达式，例如普通的表达式 $2 + 3$ 的逆波兰表示法为 $+ 2 3$ 。逆波兰表达式的优点是运算符之间不必有优先级关系，也不必用括号改变运算次序，例如 $(2 + 3) * 4$ 的逆波兰表示法为 $* + 2 3 4$ 。本题求解逆波兰表达式的值，其中运算符包括 $+ - * /$ 四个。

### 输入格式

输入为一行，其中运算符和运算数之间都用空格分隔，运算数是浮点数。。

### 输出格式

输出为一行，表达式的值。

可直接用`printf("%f\n", v)`输出表达式的值 $v$ 。

### 输入样例

```
* + 11.0 12.0 + 24.0 35.0
```

### 输出样例

```
1357.000000
```

### 解析

前缀表达式：符号后紧跟着两个运算数字的表达式。

在输入处理中，不能使用`char`类型，会导致只能读取一个字符，超过一位数的数字就只能读第一位，例如读入36，只会记录个3；

在字符串到浮点型转化时，主要用的是`atof()`，`atof()`能够将字符数组转换为浮点数；

### 编码

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

using namespace std;

//每次读入的长度
```

```
char str[101];
//最终结果
double res;

double exp() {
    //读入字符串数组
    scanf("%s", str);
    //判断第一位的符号，进行各类计算
    switch (str[0]) {
        case '+':
            res = exp() + exp();
            break;
        case '-':
            res = exp() - exp();
            break;
        case '*':
            res = exp() * exp();
            break;
        case '/':
            res = exp() / exp();
            break;
        default:
            //将字符串转为浮点数
            res = atof(str);
    }
    return res;
}

int main() {
    printf("%f\n", exp());
}
```

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。

