



链滴

UVA 442

作者: [cms42](#)

原文链接: <https://ld246.com/article/1563960538275>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

提出一种新的解法：递归（虽然我明知道这是栈处理表达式的裸题）

```
#include<iostream>
#include<algorithm>
#include<string>
using namespace std;

typedef pair<int,int> Mat;
Mat a[30]; //储存字母对应的矩阵
int n,x,y;
char c;
string s;
int ps,ls; //ls:字符串s的长度 ps:字符串s的指针

int solve(Mat& t){ //Mat为乘法运算后的矩阵，返回值为当前运算次数
    char c;
    Mat x(0,0),y(0,0);
    int x1=0,y1=0,ans=0;
    if(ps==ls) return 0;
    c=s[++ps];
    if(c=='('){ //左括号，递归处理
        x1=solve(x);
        if(x1==-1) return -1;
        ++ps; //右括号
    }
    else if(c==')'){
        --ps; //必须由其上一层函数处理右括号，想一想为什么
        return 0;
    }
    else{
        x=a[c-'A'];
    }
    if(ps==ls){
        t=x;
        return x1;
    }
    c=s[++ps];
    if(c=='('){ //左括号，递归处理
        y1=solve(y);
        if(y1==-1) return -1;
        ++ps; //右括号
    }
    else if(c==')'){
        --ps; //必须由其上一层函数处理右括号，想一想为什么
        t=x;
        return x1; //注意这里不能返回0，因为第一个参数可能是一个子运算，可能已有运算次数
    }
    else{
        y=a[c-'A'];
    }
    if(x.second!=y.first) return -1; //error
    ans=x1+y1+x.first*x.second*y.second;
    t.first=x.first;
    t.second=y.second;
}
```

```
    return ans;
}

int main(){
    cin >> n;
    for(int i=1;i<=n;i++){
        cin >> c >> x >> y;
        a[c-'A'].first=x;
        a[c-'A'].second=y;
    }
    while(cin >> s){
        ps=-1; ls=s.length()-1;
        Mat t(0,0);
        int res=solve(t);
        if(res==-1) cout << "error\n";
        else cout << res << "\n";
    }
    return 0;
}
```