计算四则表达式的值（支持带小括号）

```
#include<strstream>
namespace fy_Exp{
namespace {template <class _T>
inline _T GetExpValue(_T t[], char& csym){
    char c=csym; csym=0;
    switch(c){
    case '+':return t[0] += t[1];
    case '-':return t[0] -= t[1];
    case '*':return t[0] *= t[1];
    default: return t[0] /= t[1];//case '/':
    }
}}
template <class _T, class _Tstream>
// istrin: inputstream, nReturn: get return value
// Return nonzero if get value successfully
int GetExpValue(_Tstream& istrin, _T& nReturn){
    _T t[3] = {0};
    char csym[3] = "++";
    int nLevel = 1, nERR = 0, nSym = 1;
    if(istrin>>t[1])nSym=0;else istrin.clear();
    for(;;){
        if(istrin>>csym[2]){
            switch(csym[2]){
            case '(':
                if(!csym[1]){nLevel=0x100; nERR=1;}else
                if(!GetExpValue(istrin, t[2]))nLevel|=0x10;
                else{nLevel=0x100; nERR=1;}
                break;
            case ')':
                if(nSym==1)nERR=-1;
                nLevel=0x100;break;
            case '+':case '-':case '*':case '/':
                if(nSym==1){nLevel=0x100; nERR=1;}
                else{nSym=1; csym[nLevel++]=csym[2];}
                break;
            case ' ':case '\r':case '\n':case '\t':continue;
            default:
                {nLevel=0x100; nERR=1;}
            }
            if(nLevel==0x100)break;
            if(nLevel&0x10 || istrin>>t[2]){
                nLevel &= 0xF;nSym=0;
                if(nLevel==1){t[1]=t[2];csym[1]=0;continue;}
                if(csym[1]=='*')GetExpValue(t+1, csym[1]);
                else if(csym[1]=='/')
                {
                    if(t[2]==0){nERR = 1; break;}
                    GetExpValue(t+1, csym[1]);
                }
                else{
                    GetExpValue(t, csym[0]);
                    t[1]=t[2];csym[0]=csym[1];csym[1]=0;
                }
                nLevel = 1;
            }
            else istrin.clear();
        }
        else{nERR = -1; break;}
    }
    if(csym[1])t[2]=0,nReturn=GetExpValue(t+1, csym[1]);
    else nReturn=GetExpValue(t, csym[0]);
    return nERR==-1?1:0;
}
template <class _T>
// strin: input string, nReturn: get return value
// Return nonzero if get value successfully
int GetExpStrValue(const char* strin, _T& nReturn){
    std::istrstream isin(strin);
    if(GetExpValue(isin, nReturn))return 1;
    return 0;
}
}

#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
    char str[1000];
    while (std::cin>>str)
    {
        double d;
        if(fy_Exp::GetExpStrValue( str, d))
        {
            std::cout<<d<<endl;
        }
        else
        {
            std::cout<<"ERROR\n";
        }
    }

    return 0;
}
```