```
1: #include<iostream>
 2: #include<cstdlib>
 3: #include<cmath>
4: using namespace std;
 5:
 6: class science{
       friend ostream& operator<<(ostream&out, science &sci);</pre>
 8:
        friend istream& operator>>(istream&in, science &sci);
9: private:
10:
       double a;
11:
       int n;
12: public:
13:
        science():a(0), n(0) {}
        science(double a_) : a(a_), n(0) {}
14:
        science(double a , int n ): a(a ), n(n ){
15:
            if(a!=0)
16:
17:
       {
           while(abs(a)>=10)
18:
19:
20:
               a/=10;
21:
               n+=1;
22:
23:
           while(abs(a)<1)</pre>
24:
25:
               a*=10;
26:
               n-=1;
27:
           }
28:
       }
29:
       // Member function declarations
30:
31:
       // ----- Write Your Code Here ----- //
32:
33:
       const science operator+(const science&)const;
34:
       const science operator-(const science&)const;
35:
       const science operator*(const science&)const;
36:
       const science operator/(const science&)const;
37:
       science operator=(const science&);
38:
39: };
41: // Finish the ctor and function definitions
42: // ----- Write Your Code Here ----- //
44: ostream& operator<<(ostream&out, science &sci){
45: if(sci.a!=0)
46:
       {
```

```
47:
            while(abs(sci.a)>=10)
48:
            {
49:
                sci.a/=10;
50:
                sci.n+=1;
51:
52:
            while(abs(sci.a)<1)</pre>
53:
54:
                sci.a*=10;
55:
                sci.n-=1;
56:
            }
57:
        out<<sci.a<<"*10^"<<sci.n;
58:
59:
        return out:
60: }
61: istream& operator>>(istream&in,science &sci){
        in>>sci.a>>sci.n;
63:
        return in:
64: }
65: const science science::operator+(const science& rhs) const{
66:
            int i=n-rhs.n;
            return science(pow(10,i)*a+rhs.a,rhs.n);
67:
68:
69: const science science::operator-(const science& rhs) const{
70:
            int i=n-rhs.n;
71:
            return science(pow(10,i)*a-rhs.a,rhs.n);
72:
73: const science science::operator*(const science& rhs) const{
74:
            return science(a*rhs.a,n+rhs.n);
75:
76: const science science::operator/(const science& rhs) const{
            return science(a/rhs.a,n-rhs.n);
77:
78:
        }
79: science science::operator=(const science& rhs) {
80:
            a=rhs.a;
81:
            n=rhs.n;
82:
            return *this;
83:
84: int main(){
85:
        science v, t, w;
86:
        char op;
87:
        while(1){
            cout << "Please enter an expression:" << endl << endl;</pre>
88:
89:
90:
            cin >> v:
91:
92:
            cin >> op;
```

```
if( op != '+' && op != '-' && op != '*' && op != '/' )break
 93:
 94:
 95:
             cin >> t;
             if(cin.fail()) break;
 96:
 97:
             switch(op){
 98:
 99:
                 case '+': w = v + t; break;
                 case '-': w = v - t; break;
100:
101:
                 case '*': w = v * t; break;
102:
                 case '/': w = v / t; break;
103:
             cout<<v<<' '<<op<<' '<<t<" = "<<w<<endl<<endl;</pre>
104:
105:
         return 0;
106:
107: }
```