

```

1  #include <iostream>
2
3  using namespace std;
4
5  /**
6   * Tort osztály.
7   * Racionális tört viselkedést megvalósító osztály.
8   */
9  class Tort {
10     int szamlalo;
11     int nevező;
12     void egysz();
13 public:
14     Tort( int sz=0, int n=1) { szamlalo=sz; nevező=n; egysz(); }
15     void Set( int sz, int n=1 ) { szamlalo=sz; nevező=n; egysz(); }
16     int GetSz() const { return szamlalo; }
17     int GetNev() const { return nevező; }
18     double GetValue() const { return (double)szamlalo/(double)nevező; }
19
20     Tort operator+( const Tort& a )
21     {
22         Tort t;
23         t.Set(szamlalo*a.nevező+a.szamlalo*nevező,nevező*a.nevező);
24         return t;
25     }
26     Tort operator-( const Tort& a )
27     {
28         Tort t;
29         t.Set(szamlalo*a.nevező-a.szamlalo*nevező,nevező*a.nevező);
30         return t;
31     }
32     Tort operator*( const Tort& a )
33     {
34         Tort t;
35         t.Set(szamlalo*a.szamlalo,nevező*a.nevező);
36         return t;
37     }
38     Tort operator/( const Tort& a )
39     {
40         Tort t;
41         t.Set(szamlalo*a.nevező,nevező*a.szamlalo);
42         return t;
43     }
44 };
45
46 void Tort::egysz()
47 {
48     int a=szamlalo;
49     int b=nevező;
50     int r;
51     while (r=a%b)
52     {
53         a=b;
54         b=r;
55     }
56     szamlalo/=b;
57     nevező/=b;
58 }
59

```

```

60
61 istream& operator>>(istream& s, Tort& t)
62 {
63     int n,m;
64     s >> n, s.ignore(1), s >> m;
65     t.Set(n,m);
66     return(s);
67 }
68
69 ostream& operator<<(ostream& s, const Tort& t)
70 {
71     s << t.GetSz() << '/' << t.GetNev();
72     return(s);
73 }
74
75 Tort operator*( const Tort& a , const Tort& b)
76 {
77     Tort t;
78     t.Set(a.GetSz()*b.GetSz(),a.GetNev()*b.GetNev());
79     return t;
80 }
81
82 /**
83  * Főprogram.
84  */
85 int main()
86 {
87     int i;
88     Tort a1( 3 , 4);
89     Tort a2;
90     Tort z;
91
92     cout << "a1=" << a1 << " erteke=" << a1.GetValue() << endl;
93     cout << "Adj meg egy tortet (pld. 3/4): ";
94     cin >> a2 ;
95     cout << "a2=" << a2 << " erteke=" << a2.GetValue() << endl;
96     z = a1 + a2;
97     cout << "a1+a2=" << z << endl;
98     z = a1 - a2;
99     cout << "a1-a2=" << z << endl;
100    z = a1 * a2;
101    cout << "a1*a2=" << z << endl;
102    z = a1 / a2;
103    cout << "a1/a2=" << z << endl;
104    z = a1 * 8;
105    cout << "a1*8=" << z << endl;
106    z = 8 * a1;
107    cout << "8*a1=" << z << endl;
108
109    return 0;
110 }

```