

```

// Einfacher Datentyp Bruch
// Nutzung von Operatoren

#include <iostream>
#include <cstdlib>
using namespace std;

class Bruch {
private:
    int zaehler;
    int nenner;
public:
    Bruch ();
    Bruch (int);
    Bruch (int, int);

    void print ();

    Bruch operator * (Bruch);
    Bruch operator *= (Bruch);
    bool operator < (Bruch);
}; // Bruch

Bruch::Bruch () {
    zaehler = 0;
    nenner = 1;
}

Bruch::Bruch (int z) {
    zaehler = z;
    nenner = 1;
}

```

```

Bruch::Bruch (int z, int n) {
    if (n == 0) {
        cerr << "Fehler: Nenner ist 0!" << endl;
        exit (1);
    }

    // Erste Normalisierung, Nenner positiv
    if (n < 0) {
        zaehler = -z;
        nenner = -n;
    } else {
        zaehler = z;
        nenner = n;
    }
}

void Bruch::print () {
    cout << zaehler << '/' << nenner << endl;
} // print

Bruch Bruch::operator * (Bruch b) {
    // Ohne kürzen
    return Bruch (zaehler * b.zaehler, nenner * b.nenner);
} // *

Bruch Bruch::operator *= (Bruch b) {
    this->zaehler *= b.zaehler;
    this->nenner *= b.nenner;
    return *this;
} // *=

```

```
bool Bruch::operator < (Bruch b) {
    // bekannt: Nenner > 0
    return zaehler * b.nenner < b.zaehler * nenner;
} //<
```

```
int main() {
    Bruch x;
    Bruch a (7, 3);
    cout << "Bruch a = "; a.print();
    x = a * a;
    cout << "Bruch x = "; x.print();
    while (x < Bruch (10000)) {
        x *= a;
        cout << "Bruch x = "; x.print();
    }
    return 0;
} //main
```

/ Beispiellauf*

```
Bruch a = 7 / 3
Bruch x = 49 / 9
Bruch x = 343 / 27
Bruch x = 2401 / 81
Bruch x = 16807 / 243
Bruch x = 117649 / 729
Bruch x = 823543 / 2187
Bruch x = 5764801 / 6561
Bruch x = 40353607 / 19683
Bruch x = 282475249 / 59049
Bruch x = 1977326743 / 177147
```

**/*