Aufgabe:
Gegeben sei folgende Klasse **ratio** zur Darstellung rationaler Zahlen. Man implementiere die Methoden dieser Klasse. Falls der Nenner bei **ratio** oder **letnenn()** mit **0** übergeben wird, dann soll dieser auf **1** gesetzt und eine diesbezügliche Nachricht über **cout** ausgegeben werden.

Der **Konstruktor**, **Destruktor** und **Kopierkonstruktor** sollen über **cout** eine Nachricht ausgeben, die die Methode eindeutig identifiziert ("Konstuktor ratio", "Destruktor ratio", "Kopierkonstruktor ratio").

```
// rationale Zahl, dargestellt durch Zaehler und Nenner
class ratio
{  int    zaehl, nenn;                  // Zaehler, Nenner
   int    ggt(int a, int b);            // Euklidischer Algorithmus (-> Euklid.pdf)
   ratio  &kuerze();                    // Kuerzen, unter Nutzung von ggt
  public:
   ratio(int z=0, int n=1);             // Konstruktor mit default-Werten
   ratio(const ratio &);                // Kopierkonstruktor
   ~ratio();                            // Destruktor

// Methoden, die Referenz auf ratio zurueckgeben:    // Bedeutung der Methoden:
   ratio  &mult(const ratio &x);                     // (*this) = (*this) * x
   ratio  &div(const ratio &x);                      // (*this) = (*this) / x
   ratio  &mult(int i);                              // (*this) = (*this)*i
   ratio  &div(int i);                               // (*this) = (*this)/i
   ratio  &add(const ratio &x);                      // (*this) = (*this) + x
   ratio  &sub(const ratio &x);                      // (*this) = (*this) - x
   ratio  &add(int i);                               // (*this) = (*this) + i
   ratio  &sub(int i);                               // (*this) = (*this) - i
   ratio  &minus();                                  // (*this) = -(*this);

// Methoden, die Kopie von ratio zurueckgeben:              // Bedeutung der Methoden:
   static ratio  mult(const ratio &x1, const ratio &x2);  // return x1 * x2
   static ratio  div(const ratio &x1, const ratio &x2);   // return x1 / x2
   static ratio  mult(const ratio &x1, int i);            // return x1 * i
   static ratio  div(const ratio &x1, int i);             // return x1 / i
   static ratio  add(const ratio &x1, const ratio &x2);   // return x1 + x2
   static ratio  sub(const ratio &x1, const ratio &x2);   // return x1 - x2
   static ratio  add(const ratio &x1, int i);             // return x1 + i
   static ratio  sub(const ratio &x1, int i);             // return x1 - i
   static ratio  minus(const ratio &x1);                  // return -x1

   ratio  &display();                  // Ausgabe zaehl / nenn, z.B. 3/5
   int    getzaehl() const;            // Lesen zaehl
   int    getnenn() const;             // Lesen nenn
   void   letzaehl(int z=0);           // Schreiben zaehl
   void   letnenn(int n=1);            // Schreiben nenn
};
```

In der **main()** - Funktion sollen **r1**, **r2**, **r3**, **r4** dynamisch als Zeiger auf **ratio** angelegt werden und die rationalen Zahlen **1/2**, **3/5**, **2/5**, **1/8** repräsentieren.

In nur einer Anweisung ist der Wert des Ausdruckes

-(((((1/2)*(3/5)*2+2/5)/(1/8)-5)+3)/2)

zu berechnen und auszugeben. Dabei sind nur die Methoden zu nutzen, die eine ***Referenz*** auf ein **ratio**-Objekt zurückgeben, u.a. auch **display()**.

Danach ist ***r1** wieder auf **1/2** zu setzen.

Man wiederhole die Berechnung und Ausgabe obigen Ausdruckes mit nur einer Anweisung, jetzt jedoch **nur** unter Nutzung der Methoden, die eine ***Kopie*** eines **ratio**-Objektes zurückgeben, zusätzlich darf **display()** genutzt werden.