

Programmieren C: **while**-Schleife, Restrechnung; Größter gemeinsamer Teiler nach Euklid, Bruch kürzen

Klaus Kusche

Der größte gemeinsame Teiler (ggT, englisch “greatest common divisor”, gcd) zweier Zahlen ist die größte Zahl, durch die sich beide Zahlen glatt teilen lassen.

Der ggT von 15 und 6 ist beispielsweise 3. Man braucht den ggT zum Kürzen von Brüchen.

Man berechnet den ggT immer von den Absolutwerten der beiden gegebenen Zahlen.

In C liefert dir die Funktion **abs** (aus **stdlib.h**) als Ergebnis den Absolutwert eines **int**.

In der Schule berechnet man den ggT durch Primfaktorenzerlegung beider Zahlen und Zusammenmultiplizieren der gemeinsamen Faktoren, aber seit den alten Griechen ist ein viel besseres Verfahren (zumindest für Computer) bekannt, um den ggT von a und b zu ermitteln:

- Wiederhole die folgenden beiden Schritte, solange dein b nicht 0 ist:
 - Ermittle den Rest r der Division a / b .
 - Verwende das bisherige b als dein neues a und das r als dein neues b .
- Wenn dein b 0 ist, so ist a der gesuchte ggT.

Jetzt haben wir also keine Schleife, die eine fixe Anzahl von Durchläufen macht und dabei mitzählt, sondern eine, die etwas immer wieder macht, solange eine Bedingung gilt (ohne vorher zu wissen, wie oft, und ohne mitzuzählen).

In C sieht das wie folgt aus:

while (*bedingung*) {

Befehle in der Schleife;

}

Außerdem brauchst du noch die Prüfung auf “ungleich” (!= statt ==) und die Restbildung: $a \% b$ liefert (für b ungleich 0) den Rest der ganzzahligen Division a / b .

Zusatzaufgabe “Bruch kürzen”:

- Speichere das ursprüngliche a und b in zwei zusätzlichen Variablen, damit du am Ende den gekürzten Bruch ausgeben kannst!
- Erkenne $a / 0$ (==> Fehlermeldung) und behandle negative Zähler und Nenner richtig (das – gehört in der Ausgabe immer in den Zähler, zwei – heben sich auf!).
- Und wenn die Ausgabe ganz schön sein soll: Unterscheide Bruch und ganze Zahl! (d.h. wenn der gekürzte Nenner 1 ist, wird nur der Zähler ausgegeben)