Algorithmen für Kettenbrüche

Eine reelle Zahl x der Form $z_0 + \cfrac{1}{z_1 + \cfrac{1}{z_2 + \ldots + \cfrac{1}{z_n}}}$ heißt **endlicher regulärer Kettenbruch** (KB) .

Kurzschreibweise: **$x = [\, z_0 ; z_1 , z_2 , \ldots , z_n \,]$**

z_0 ist eine ganze Zahl und heißt Anfangsglied.
Die $z_1 , z_2 , \ldots , z_n$ sind natürliche Zahlen .
Bedingung: $z_n \neq 1$, denn 1/1 am Schluss wäre kein echter Teilbruch.

Anmerkung: Sind nicht alle Zähler = 1, so spricht man von einem nicht regulären Kettenbruch !

Die folgenden Algorithmen gelten für positive Kettenbrüche !

Für negative Zahlen x rechne man mit |x| = abs(x) und setze vor das Ergebnis ein Minuszeichen .

Wie wandelt man einen Bruch in einen Kettenbruch um ?

Verwendung des Euklidschen Algorithmus:
Beispiel: 67 / 29 = 2+9/29 = 2+1/(29/9) = 2+1/(3+2/9) = 2+1/(3+1/(9/2)) = **2** + 1 / (**3** + 1 / (**4** + 1 / **2**))
Kurzschreibweise: 67 / 29 = [2,3,4,2]

```
Eingabe (zähler, nenner)
Solange  nenner > 0 wiederhole
   Ausgabe (zähler div nenner)
   Setze zähler = zähler mod nenner
   Vertausche zähler mit nenner
Ende Solange
```

Rechnung für obiges Beispiel (zae=zähler ; nen=nenner) :

Gegeben sind zae=67 nen=29
zae div nen = 67 div 29 = **2**
zae = 67 mod 29 = 9
vertauschen: zae = 29 nen = 9
zae div nen = 29 div 9 = **3**
zae = 29 mod 9 = 2
vertauschen: zae = 9 nen = 2
zae div nen = 9 div 2 = **4**
zae = 9 mod 2 = 1
vertauschen: zae = 2 nen = 1
zae div nen = 2 div 1 = **2**
zae = 2 mod 1 = 0
vertauschen: zae = 2 nen = 0 Abbruch ! Die ermittelten Kettenbruchelemente sind: 2 3 4 2

Wie wandelt man einen Kettenbruch in einen Bruch um ?

Dies geschieht durch Abarbeitung des Kettenbruchterms von rechts nach links.

```
Für i von 0 bis n wiederhole
   Eingabe(zi)     // zi = Kettenbruchelemente
x = 0   // x ist der gesuchte Bruch
Für i von n ab bis 1 wiederhole
   x = 1/(x+zi)
x = x + z0
```

Rechnung für obiges Beispiel (exakte Bruchrechnung) :

Eingabe: z0 = 2 z1 = 3 z2 = 4 z3 = 2
x = 0
x = 1/(0+2) = ½
x = 1/(1/2+4) = 1/(9/2) = 2/9
x = 1/(2/9+3) = 1/(29/9) = 9/29
x = 9/29 + 2 = **67/29**

Wie wandelt man eine Dezimalzahl in einen Kettenbruch um ?

Dies geschieht durch Abspaltung von ganzzahligen Anteilen der Teilbrüche des Kettenbruchs.
Der Kettenbruch wird auf diese Weise von links nach rechts berechnet.
Zu beachten ist, dass die am weitesten rechts stehende Zahl (z_n) nicht 1 sein darf !
Mit dieser Methode lassen sich auch Bruchnäherungen reeller Zahlen bilden .
Achtung: Hier kann auch ein unendlicher Kettenbruch entstehen !
Deshalb sollte man eine Grenze nmax für die Anzahl der Stellen angeben !

```
Eingabe(x, nmax)   // x ist die gegebene Dezimalzahl
n=1
Solange (x <> int(x)) und (n<=nmax) wiederhole
   Ausgabe(int(x))
   fracx = x - int(x)
   Falls fracx > 0
      dann x = 1 / fracx
   n=n+1
Ende Solange
Falls letzte Zahl des KBs = 1, dann addiere sie zur vorletzten Zahl
```

Achtung: Durch die fehlerbehaftete Gleitkommaarithmetik kann es zu groben Rundungsfehlern kommen, insbesondere bei der Division 1 / fracx . Daher wird bereits nach wenigen Schleifendurchläufen mit Sicherheit ein falsches Kettenbruchelement ermittelt werden ! (vgl. Beispiele unten)
Wegen der Gleikommaarithmetik ist auch ein Test auf Gleichheit bzw. Ungleichheit wie bei fracx > 0 sinnlos.
Man muss hier auf **fracx < eps** (mit z.B. eps = 1e-15) testen .

(Exakte) Rechnung für x = 3,125:
int(3,125) = **3**
fracx = 3,125-3 = 0,125 <> 0 daher x = 1/0,125 = 8
int(8) = **8**
fracx = 8-8 = 0
Der Kettenbruch lautet daher [3,8]. Der zugehörige Bruch ist 1/8+3 = 25/8

(Exakte) Rechnung für x = 9,7:
int(9,7) = **9**
fracx = 9,7-9 = 0,7 <> 0 daher x = 1/0,7 = 10/7 = 1,42857...
int(10/7) = **1**
fracx = 10/7-1 = 3/7 = 0,428571428571... (Periode) <> 0 daher x = 1/(3/7) = 7/3 = 2,333...
int(7/3) = **2**
fracx = 7/3-2 = 1/3 = 0,3333... (Periode) <> 0 daher x = 1/(1/3) = 3
int(3) = **3**
fracx = 3-3 = 0 Abbruch !
Der Kettenbruch lautet daher [9;1,2,3]. Als Bruch 9+1/(1+1/(2+1/3)) = 97/10

(Exakte) Rechnung für x = 0,18:
int(0,18) = **0**
fracx = 0,18-0 = 0,18 <> 0 daher x = 1/0,18 = 5,55555555.. (eine Periode !)
int(5,555..) = **5**
fracx = 5,555...-5 = 0,555... daher x = 1/0,555... = 1,8
int (1,8) = **1**
fracx = 1,8-1 = 0,8 daher x = 1/0,8 = 1,25
int (1,25) = **1**
fracx = 1,25-1 = 0,25 daher x = 1/0,25 = 4
int (4) = **4**
fracx = 4-4 = 0 Abbruch !
Kettenbruch x = [0;5,1,1,4]. Als Bruch 1/(5+1/(1+1/(1+1/4))) = 9/50

Wegen der oben erläuterten unvermeidlichen Rundungsfehler ist evtl. eine andere Methode besser geeignet:

Die Dezimalzahl in einen Bruch umwandeln und diesen dann (s.o.) in einen Kettenbruch überführen!

```
Eingabe(x)    // x ist die gegebene Dezimalzahl
x in Bruch z umwandeln mit Zähler zae und Nenner nen  ( siehe unten)
Solange  nen > 0 wiederhole
   Ausgabe(zae div nen)
   Setze zae = zae mod nen
    Vertausche zae mit nen
Ende Solange
```

x in Bruch z umwandeln:

- **Anzahl n der Nachkommastellen von x ermitteln**
- **zae = x ohne Komma**
- **nen = 10^n**

Java7 - Methode dazu:

```
void xZuBruch(double x) {
   String s = String.valueOf(x);
   int pos = s.indexOf('.');
   String sNk = s.substring(pos + 1);
   int nkStellen = sNk.length();
   long zae = Long.parseLong(s.substring(0, pos) + sNk);
   long nen = 1;
   for (int i = 1; i <= nkStellen; i++) nen = 10 * nen;
}
```

Anmerkung: In der Programmiersprache Java sind bei Verwendung des Datentyps Long (max 9223372036854775807L) für Zähler und Nenner jeweils bis zu 19-stellige Zahlen (10^{18}) möglich.

Beispiel: x = 3,141592653589793 (Näherung für Pi ; 15 Nachkommastellen)

Die Umwandlung in einen Bruch liefert :

zae = 3141592653589793
nen = 10^{15} = 1000000000000000

Umwandlung des Bruches in einen Kettenbruch :

zae div nen = 3141592653589793 div 1000000000000000 = **3**
zae = 3141592653589793 mod 1000000000000000 = 141592653589793
zae div nen = 1000000000000000 div 141592653589793 = **7**
zae = 1000000000000000 mod 141592653589793 = 8851424871449
zae div nen = 141592653589793 div 8851424871449 = **15**
zae = 141592653589793 mod 8851424871449 = 8821280518058
zae div nen = 8851424871449 div 8821280518058 = **1**

usw.

Es entsteht schließlich der für die Pi-Approximation 3,141592653589793 exakte Kettenbruch
[3;7,15,1,292,1,1,1,2,1,3,1,14,**4**,2,3,1,12,5,1,5,20,1,11,1,1,1,2] .

Die Zahl 4 ist die erste falsche Zahl des Kettenbruchs; demnach sind auch die restlichen Zahlen falsch !

Zum Vergleich (exakte Zahlen für den Kettenbruch von Pi):
[3;7,15,1,292,1,1,1,2,1,3,1,14,**2**,1,1,2,2,2,2,1,84,2,1,1,15,3]

Die Bedeutung der Kettenbrüche

(1) Konstruktion von Zahnrädern:

1682 baute **Christaan Huygens** (berühmter niederländischer Astronom / Physiker ; 1629-1695) ein Planetarium, in dem das Verhältnis der Umlaufzeiten von Saturn und Erde um die Sonne dargestellt werden sollte.

Huygens fand für die Umlaufzeiten von Saturn und Erde das Verhältnis 77708431 : 2640858 ≈ 29,42544847 . Für eine passende Astronomische Uhr mussten die Zahnräder mit wesentlich weniger Zähnen als die hier genannten Zahlen auskommen, weil kleine Zahnräder mit einer hohen Zähnezahl technisch schwer zu realisieren sind. Deswegen suchte Huygens für dieses Verhältnis optimale Näherungen mithilfe von Kettenbrüchen.

Gesucht waren also möglichst kleine natürliche Zahlen a und b mit a / b ≈ 29,43 .

Mithilfe einer Kettenbruchentwicklung für 77708431 / 2640858 gelang Huygens die folgende Näherung:

77708431 / 2640858 = 29 + 1123549 / 2640858 = 29 + 1 / (2640858 / 1123549) = 29 + 1 / (2 + 393760 / 1123549)

Vernachlässigt man hier den Bruch 393760 / 1123549, so erhält man als Näherung 29 + 1 / 2 = 59 / 2.
Der Fehler bezüglich 77708431 / 2640858 beträgt 0,25% , ist also noch zu groß.

Weitere Zerlegung:
29 + 1 / (2 + 393760 / 1123549) = 29 + 1 / (2 + 1 / (1123549 / 393760)) = 29 + 1 / (2 + 1 / (2 + 336029 / 393760))

Die Näherung ist dann 29 + 1 / (2 + 1 / 2) = 29 + 1 / (5/2) = 29 + 2/5 = 147 / 5 .
Der Fehler bezüglich 77708431 / 2640858 beträgt 0,086% . Dies war Huygens immer noch zu viel.

Daher ersetzte er den Restbruch 336029 / 393760 durch 1 und erhielt als Näherung:
29 + 1 / (2 + 1 / (2 + 1)) = 29 + 1 / (2 + 1 / 3) = 29 + 1 / (7 / 3) = 29 + 3 / 7 = **206 / 7** .

Mit dem **Fehler von 0,01%** bezüglich 77708431 / 2640858 war Huygens dann zufrieden.

Er verwendete das Zahnradverhältnis 206 : 7 !

(2) Näherungsbrüche für reelle Zahlen finden :

Wenn man beim Taschenrechner TI84 45*78/99 ausrechnet und dann auf das Ergebnis 35.4545... MATH Frac anwendet, so erhält man den Bruch 390/11 . Dies ist der exakte Bruch für die durchgeführte Rechnung.
Ebenso kann man 35.4545454545 eintippen und nach ENTER wieder MATH Frac aufrufen.
Tippt man jedoch 35.45454546 ein, so erhält man nach ENTER MATH Frac wieder 390/11.
Dies zeigt, dass der Rechner versucht, für eingegebene Dezimalzahlen einen bestmöglichen Näherungsbruch zu finden. Dies gelingt nicht immer, wie das Beispiel 35.45454547 zeigt . Es wird kein Bruch erzeugt !

Einen Näherungsbruch zu einer gegebenen Dezimalzahl x kann man mittels Kettenbruchentwicklung von x finden.
Soll dieser Näherungsbruch bezüglich der Rechnersoftware optimal sein, so muss man nur prüfen, ob der Wert des jeweiligen Kettenbruchs sich um höchstens 10^{-15} (Datentyp double) von x unterscheidet.

Beispiel : x = 99999999 / 777777777 = 0,1285714274142857

Entwicklung von x in einen Kettenbruch (Rechnung mit 16 Nachkommastellen):

x = 0.1285714274142857 = **0**+1/(1/ 0.1285714274142857) = **0**+1/7,7777778477777793 =
0+1/(**7**+0,7777778477777793) = **0**+1/(**7**+1/(1/0,7777778477777793)) =
0+1/(**7**+1/(1,2857141700000079)) = **0**+1/(**7**+1/(**1**+0,2857141700000079)) =
0+1/(**7**+1/(**1**+1/(1/0,2857141700000079))) = **0**+1/(**7**+1/(**1**+1/(3,5000014175004773))) =
0+1/(**7**+1/(**1**+1/(**3**+0,5000014175004773))) = **0**+1/(**7**+1/(**1**+1/(**3**+1/(1/0,5000014175004773)))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(1,9999943300141652)))) = **0**+1/(**7**+1/(**1**+1/(**3**+1/(**1**+0,9999943300141652)))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(1/0,9999943300141652))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(1,0000056700179837))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+0,0000056700179837))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(1/0,0000056700179837)))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(176366,2836475599236996)))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+0,2836475599236996)))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(1/0,2836475599236996))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(3,5255018596634401))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+0,5255018596634401))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(1/0,5255018596634401)))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(1,9029428376151785)))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(**1**+0,9029428376151785)))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(**1**+1/(1/0,9029428376151785))))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(**1**+1/(1,1074898192239561))))))))) =
0+1/(**7**+1/(**1**+1/(**3**+1/(**1**+1/(**1**+1/(**176366**+1/(**3**+1/(**1**+1/(**1**+0,1074898192239561)))))))))

Hier kann abgebrochen werden, denn [0;7,1,3,1,1,176366,3,1,1] = 0.1285714274142857 .
Damit gibt es in der letzten Nachkommastelle keine Abweichung mehr.

Wandelt man den Kettenbruch in einen Bruch um, so erhält man 11111111/86419753 .
Dies ist der mit 9 gekürzte Ausgangsbruch 99999999 / 777777777 !

Exakte Werte von Kettenbrüchen

π = [3;7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1,84,2,1,1,15,3,13,1,4,2,6,6,99,1,2,2,6,3,5,
1,1,6,8,1,7,1,2,3,7,1,2,1,1,12,1,1,1,3,1,1,8,1,1,2,1,6,1,1,5,2,2,3,1,2,4,4,16,1,161,45,1,22,1,2,2,1,4,1,2,24,1,2,1,3
,1,2,1,1,10,2,5,4,1,2,2,8,1,5,2,2,26,1,4,1,1,8,2,42,2,1,7,3,3,1,1,7,2,4,9,7,2,3,1,57,1,18,1,9,19,1,2,18,1,3,7,30,1,1
,1,3,3,3,1,2,8,1,1,2,1,15,1,2,13,1,2,1,4,1,12,1,1,3,3,28,1,10,3,2,20,1,1,1,1,4,1,1,1,5,3,2,1,6,1,4,1,120,2,1,1,3,1,2
3,1,15,1,3,7,1,16,1,2,1,21,2,1,1,2,9,1,6,4,127,14,5,1,3,13,7,9,1,1,1,1,1,5,4,1,1,3,1,1,29,3,1,1,2,2,1,3,1,1,1,3,1,1,
10,3,1,3,1,2,1,12,1,4,1,1,1,1,7,1,1,2,1,11,3,1,7,1,4,1,48,16,1,4,5,2,1,1,4,3,1,2,3,1,2,2,1,2,5,20,1,1,5,4,1,436,8,1,
2,2,1,1,1,1,1,5,1,2,1,3,6,11,4,3,1,1,1,2,5,4,6,9,1,5,1,5,15,1,11,24,4,4,5,2,1,4,1,6,1,1,1,4,3,2,2,1,1,2,1,58,5,1,2,1,
2,1,1,2,2,7,1,15,1,4,8,1,1,4,2,1,1,1,3,1,1,1,2,1, ...]

e = [2;1,2,1,1,4,1,1,6,1,1,8,1,1,10,1,1,12,1,1,14,1,1,16,1,1,18,1,1,20,1,1,22,1,1,24,1,1,26,1,
1,28,1,1,30,1,1,32,1,1,34,1,1,36,1,1,38,1,1,40,1,1,42,1,1,44,1,1,46,1,1,48,1,1,50,1,1,52,1,1,54,1,1,56,1,1,58,1,1
,60,1,1,62,1,1,64,1,1,66,1,1,68,1,1,70,1,1,72,1,1,74,1,1,76,1,1,78,1,1,80,1,1,82,1,1,84,1,1,86,1,1,88,1,1,90,1,1,
92,1,1,94,1,1,96,1,1,98,1,1,100,1,1,102,1,1,104,1,1,106,1,1,108,1,1,110,1,1,112,1,1,114,1,1,116,1,1,118,1,1,12
0,1,1,122,1,1,124,1,1,126,1,1,128,1,1,130,1,1,132,1,1,134,1,1,136,1,1,138,1,1,140,1,1,142,1,1,144,1,1,146,1,1,
148,1,1,150,1,1,152,1,1,154,1,1,156,1,1,158,1,1,160,1,1,162,1,1,164,1,1,166,1,1,168,1,1,170,1,1,172,1,1,174,1
,1,176,1,1,178,1,1,180,1,1,182,1,1,184,1,1,186,1,1,188,1,1, ...]
Hier ist ein Bildungsgesetz zu erkennen ! Man könnte also die Folge fortsetzen.

√(2) = [1;2,2,2,2,2,2,2,2,2,2,2,2,2,2, ...]
√(3) = [1;1,2,1,2,1,2,1,2,1,2,1,2, ...]
√(5) = [2;4,4,4,4,4,4,4,4,4,4,4,4,4,4, ...]
√(6) = [2;2,4,2,4,2,4,2,4,2,4,2,4, ...]
√(31) = [5;1,1,3,5,3,1,1,10,1,1,3,5,3,1,1,10,1,1,3,5,3,1,1,10,1,1,3,5,3,1,1,10,1,1,3, ...]

Allgemein gilt für natürliche Zahlen k :

$$\sqrt{k} = [\,a0\,;\ \overline{a1\,,\ \ldots\,,\ an\,,\ 2a0}\,] \quad \text{mit} \quad a0 = \mathrm{int}(\sqrt{k})$$

Beweis für √(3):

√(3) = **1** + √(3)-1 ;   mit 1 = int(√(3)) und √(3)-1 = frac(√(3))
1/(√(3)-1) = (√(3)+1)/((√(3)+1)(√(3)-1)) = (√(3)+1)/(3-1) = (2+√(3)-1)/2 = **1** + (√(3)-1)/2
mit 1 = int   und (√(3)-1)/2 = frac
2/(√(3)-1) = 2(√(3)+1)/((√(3)+1)(√(3)-1)) = 2(√(3)+1)/(3-1) = √(3)+1 = **2** + √(3)-1
mit 2 = int   und √(3)-1 = frac
Als nächstes wäre also 1/(√(3)-1) zu bilden; dies war 2 Schritte vorher aber schon berechnet worden . Wir haben damit eine Periode erhalten, denn 1 und 2 wechseln fortan ab.
Damit ist $\sqrt{3} = [1;\overline{1,2}]$

Beweis für √(5):

√(5) = **2** + √(5)-2 ;   mit 2 = int(√(5)) und √(5)-2 = frac(√(5))
1/(√(5)-2) = (√(5)+2)/((√(5)+2)(√(5)-2)) = (√(5)+2)/(5-4) = **4** + √(5)-2
mit 4 = int   und √(5)-2 = frac
Als nächstes wäre also 1/(√(5)-2) zu bilden; dies war 1 Schritt vorher aber schon berechnet worden . Wir haben damit eine Periode erhalten, denn die 4 wiederholt sich immer.
Damit ist $\sqrt{5} = [2;\overline{4}]$

Beispiele (Ergebnisse wurden ermittelt mit Java7 ; Datentyp double):

Bruch in Kettenbruch umwandeln:

67/29 = [2; 3, 4, 2]
77708431/2640858 = [29;, 2, 2, 1, 5, 1, 4, 1, 1, 2, 1, 6, 1, 10, 2, 2, 3]

Und rückwärts: Kettenbruch in Bruch umwandeln:

[29;2,2,1,5,1,4,1,1,2,1,6,1,10,2,2,3] = 77708431 / 2640858
[29;2,2,1,5,1,4,1,1,2,1,6,1,10,2,2,3] = 77708431 / 2640858

Kettenbruch in Dezimalzahl umwandeln:

e = [2;1,2,1,1,4,1,1,6,1,1,8,1,1,10,1,1,12,1,1,14,1,1,16,1,1,18,1,1,20,1,1,22,1,1,24,1,1,26,1,1,28,1,1,30,1,1] = 2.7182818284590455

(wurzel(5)-1)/2 = [0;1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] = 0.6180339887498969

-77708431/2640858 = -[29;2,2,1,5,1,4,1,1,2,1,6,1,10,2,2,3] = -29.42544847167095

wurzel(2) = [1;2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2] = 1.4142135623730951

pi = [3;7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1,84,2,1,1,15,3,13,1,4,2,6,6,99,1,2,2,6,3,5,1,1,6,8,1,7,1,2,3,7,1,2,1,1,12,1,1,1,3,1,1,8,1,1,2,1,6,1,1,5,2,2,3,1,2,4,4,16,1,161,45,1,22,1,2] = 3.141592653589793

log10(2) = [0;3,3,9,2,2,4,6,2,1,1,3,1,18,1,6,1,2,1,1,5,2,1,12,12,2,11,1,141] = 0.3010299956639812

ln(2) = [0;1,2,3,1,6,3,1,1,2,1,1,1,1,3,10,1,1,1,2,1,1,1,1,3,2,3,1,13,7,4,1,1,1,7,2,4,1,1,2,5,14,1,10,1,4,2,18,3,1,4,1,6,2,7,3,3,1,13,3,1,4,4,1,3,1,1,1,1,2,17,3,1,2,32,1,1,1,1,3] = 0.6931471805599453

99999999/777777777 = [0;7,1,3,1,1,176366,3,1,1]= 0.1285714274142857

Dezimalzahl in Kettenbruch umwandeln:

pi = 3.141592653589793 = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 3, 3, 23, 1, 1, 7, 4, 35, 1, 1, 1, 2, 3, 3, 3, 3, 1, 1, 14, 6, 4, 5, 1, 7, 1, 5, 1, 1, 3, 18, 2, 1, 2, 4, 2, 96, 2, 3]
genau: pi =[3;7,15,1,292,1,1,1,2,1,3,1,14,2,1,1,2,2,2,2,1,84,2,1,1,15,3,13,1,4,2,6,6,99,1,2,2,6,3,5,1,1,6,8,1,7,1,2,3,7,1,2]

e = 2.718281828459045 = [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, 1, 12, 1, 1, 11, 3, 2, 1, 3, 1, 73, 6, 1, 1, 1, 1, 1, 2, 31, 1, 1, 1, 2, 1, 1, 2, 1, 2, 15, 9, 1, 3, 1, 4, 2]
genau: e =[2;1,2,1,1,4,1,1,6,1,1,8,1,1,10,1,1,12,1,1,14,1,1,16,1,1,18,1,1,20,1,1,22,1,1,24,1,1,26,1,1,28,1,1,30,1,1]

wurzel(2) = 1.4142135623730951 = [1; 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 3, 3]

wurzel(3) = 1.7320508075688772 = [1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 3]

wurzel(31) = 5.5677643628300215 = [5; 1, 1, 3, 5, 3, 1, 1, 10, 1, 1, 3, 5, 3, 1, 1, 10, 1, 1, 7, 8, 3, 1, 2, 2]

9,7 = [9;1,2,3,15011998757901,1,1,1,8,3,6,1477558932864,2,1,41,1,4139337892803,1,3,4,59,1,19628659463,1,4]

```
Dezimalzahl in bestmöglichen Näherungsbruch umwandeln:

0.5151515151515151 = 17 / 33

29.42544847167095 = 77708431 / 2640858

8000.005840104263 = 9876543210 / 1234567

0.1285714274142857 = 11111111 / 86419753


Periodische Dezimalbrüche in Brüche umwandeln:

0,67p27 = 37 / 55

0,08p05013 = 17889 / 222220


Bruch in periodischen Dezimalbruch umwandeln:

vorPeriodenLaenge = 3
periodenLaenge = 2
777 / 88 = 8,829p54
```