

g-adische Darstellung

```

g-adische Zahldarstellung Haftendorn 2011
Binärsystem <--> Dezimalsystem
44 ▶ Base2 • 0b101100 0b101100 ▶ Base10 • 44 0b111011 ▶ Base10 • 59
Hexadezimalsystem <--> Dezimalsystem
3600 ▶ Base16 • 0hE10 0hE10 ▶ Base10 • 3600
Hexadezimalsystem <--> Binärsystem
0hE10 ▶ Base2 • 0b111000010000 0b111000010000 ▶ Base16 • 0hE10
0hE10 ▶ Base10 • 3600 0hE10 ▶ Base10 • 3600
Das alles geht auch mit den unten definierten Funktionen
gadic(44,2) • {1,0,1,1,0,0} dezgadic({1,0,1,1,0,0},2) • 44
gadic(3600,16) • {14,1,0} E=14 dezgadic({14,1,0},16) • 3600
Direkte Übersetzungen geht mit den selbstgemachten Befehlen nicht.
gadic(dezgadic({14,1,0},16),2) • {1,1,1,0,0,0,0,1,0,0,0,0}
gadic(dezgadic({1,1,1,0,0,0,0,1,0,0,0,0},2),16) • {14,1,0}
    
```

1.1

```

Allgemeines Vorgehen
Man kann auf dieselbe Art mit jeder natürlichen Zahl g ein Zahlssystem machen.
Man nennt es das g-adische System. Die Stellen bedeuten dann
{...g4,g3,g2,g1,g0,g-1,...}
Dazu sind hier definiert gadic(z,g) und dezgadic(gli,g) gli ist dann die Liste im
g-System.
gadic(12345,60) • {3,25,45} und dezgadic({3,25,45},60) • 12345
Hat man Bruchteile im g-System, trennt man die ab, nimmt sie als Ganze und teilt
-----
zum Schluss passend. 2;30 = 2 +  $\frac{30}{60} + \frac{5}{2}$ 
1;13:56 dezgadic({13,56},60) • 836  $1 + \frac{836}{60^2} + \frac{1109}{900}$ 
    
```

1.2

```

Spielwiese
gadic(2011,5) • {3,1,0,2,1}
dezgadic({3,12,15},60) • 11535
gadic(31285,60) • {8,41,25}
dezgadic({4,37,46,40},60) • 1000000
gadic(10000000,60) • {7,42,57,46,40}
[]
    
```

1.3

```

Babylonisches Rechnen Aufgabenblatt 1
dezgadic({4,37,46,40},60) • 1000000
dezgadic({21,26,0,29,37,46,40},60) • 1000000000000 1006 • 1000000000000
gadic(1001,60) • {1,40}
gadic(1002,60) • {2,46,40}
gadic(1003,60) • {4,37,46,40}
gadic(1004,60) • {7,42,57,46,40}
gadic(1005,60) • {12,51,36,17,46,40}
gadic(1006,60) • {21,26,0,29,37,46,40}
fa := dezgadic({21,26,29,37,46,40},60) • 16672960000
gadic(1007,60) • {35,43,20,49,22,57,46,40}
gadic(fa:100,60) • {35,44,9,22,57,46,40}
    
```

1.4

```

gadic 10/11
Define LibPub gadic(z,g)=
Func
©(z,g)->wandelt Ganze Zahl
©ins g-System, gibt
Ziffernliste aus
Local x,y,li
x:=z
li:={[]}
While x>0
y:=mod(x,g)
li:=augment({y},li)
x:= $\frac{x-y}{g}$ 
EndWhile
Return li
EndFunc
x1:=44 • 44 li1:={[]} • {[]}
y1:=mod(44,5) • 4
li2:=augment({y1},li1) • {4}
x2:= $\frac{x1-y1}{5}$  • 8
y2:=mod(x2,5) • 3
li3:=augment({y2},li2) • {3,4}
x3:= $\frac{x2-y2}{5}$  • 1
y3:=mod(x3,5) • 1
li4:=augment({y3},li3) • {1,3,4}
x4:= $\frac{x3-y3}{5}$  • 0
li4 • {1,3,4} das macht return
x3>0 • true x4>0 • false Prüfungen ok
    
```

1.5

```

dezgadic 6/13
Define LibPub dezgadic(gli,g)=
Func
Local x,y,li
x:=0
li:=gli
While dim(li)>0
y:=li[1]
If dim(li)-1 Then
li:={[]}
Else
li:=mid(li,2)
EndIf
x:=x*g+y
EndWhile
Return x
EndFunc
li({1,2,3},1) • {1}
mid({1,2,3,4},2) • {2,3,4}
dezgadic({2,3,2},5) • 67
gadic(67,5) • {2,3,2}
    
```

1.6