

Es sollen Zahlen von einem Basissystem in ein anderes umgewandelt werden.
Die Quellbasis sei g , die Zielbasis gz .

Die umzuwandelnde Zahl sei $zV_k = a_n g^n + a_{n-1} g^{n-1} + \dots + a_1 g^1 + a_0$

Gesucht ist die Darstellung $zV_k = b_m g^m + b_{m-1} g^{m-1} + \dots + b_1 g^1 + b_0$ Beachte: $m \neq n$

Es sind Basen von $g=2$ bis $g=36$ möglich. Für jede Basis g sind Ziffern von 0 bis $g-1$ erlaubt !

Bei Basen größer als 10 verwendet man für Ziffern oberhalb von 9 die Buchstaben A(=10), B(=11), ..., Z(=35).

Z ist nur erlaubt für die Basis $g = 36$.

Algorithmus für die Umwandlung von der Basis g auf die Basis 10:

```
// Eingaben und Berechnung von zVk
Lies g // Zahl Bereich 2,3, ... ,9,11, ... ,36
Für i von n ab bis 0 wiederhole Lies a[i] // Ziffern im Bereich 2,3, ... , 9,A,B, ... ,Z
zVk = a[n]
Für i von n-1 ab bis 0 wiederhole zVk = g·zVk + a[i]
Schreib zVk
```

Algorithmus für die Umwandlung von der Basis 10 auf die Basis g :

```
// Eingaben und Berechnung der Koeff. c[i] von zVk
Lies g // Zahl im Bereich 2,3, ... ,9,11, ... ,36
Lies zVk // Ziffern im Bereich 0, ... , 9
j = 0
wiederhole
  c[j] = zVk mod g
  zVk = zVk div g
  j = j+1
bis zVk = 0
Für i von j-1 ab bis 0 wiederhole Schreib c[i]
```

Variablen für das TI-Basic (zVk sei die unzuwandelnde Zahl):

```
Str0 i-tes Zeichen der Zahl zVk
Str1 umzuwandelnde Zahl zVk
Str2 Zeichenkette für „0123 ... Z“
Q Quellbasis
Z Zielbasis
T Maximale Länge der Zahl zVk
L Tatsächliche Länge der Zahl zVk
S Position des i-ten Zeichens von zVk
im String Str2
X Zwischenspeicher für zVk
L1 Liste für a[j], c[j]
I, J Schleifenzähler
M Zeilenindex(Bildschirm)
K Spaltenindex(Bildschirm)
V Index für Str2
```

TI84-Basic-Programm: BASUMWAC

```
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"→Str2
ClrHome
Disp "QUELLBASIS (≤36) "
37→Q
Repeat Q≤36
  Input "Q=",Q
  If Q>36
    Disp „ZU LANG“
  End
Disp "ZIELBASIS (≤36) "
37→Z
Repeat (Z≤36) and (Z≠Q)
  Input "Z=",Z
  If Z>36
    Disp „ZU LANG“
  If Z=Q
    Disp „Z=Q“
  End
Disp "UMZUWANDELNDE "
Disp "ZAHL(LAENGE≤ ) "
round(10/log(Q),0)→T
Output(6,13,T)
T+1→L
Repeat L≤T
  Input "X=",Str1
  length(Str1)→L
  If L>T
    Disp „ZAHL ZU LANG“
  End
For(I,1,L)
  sub(Str1,I,1)→Str0
  inString(Str2,Str0)→S
  If S>Q
    Then
      Disp „FALSCHE ZIFFERN“
      Stop
    End
  S-1→L1(I)
End
```

```
L1(1)→X
For(I,2,L)
  QX+L1(I)→X
End
0→J
Repeat X=0
  J+1→J
  remainder(X,Z)→L1(J)
  ; TI83: round(fPart(X/Z)Z,0)
  iPart(X/Z)→X
End

ClrHome
Output(2,1,"BASIS ")
Output(2,7,Q)
Output(3,1,Str1)
Output(5,1,"BASIS ")
Output(5,7,Z)
For(I,1,J)
  L1(1+J-I)→V
  If I>16
    Then
      I-16→K
      7→M
    Else
      I→K
      6→M
    End
  Output(M,K,sub(Str2,V+1,1))
End
Pause
```

TI84-Screenshots

```

QUELLBASIS(≤36)
Q=16
ZIELBASIS(≤36)
Z=2
UMZUWANDELNDE
ZAHL(LAENGE≤8 )
X=FFA28E
    
```

```

BASIS 16
FFA28E

BASIS 2
1111111110100010
10001110
    
```

```

QUELLBASIS(≤36)
Q=10
ZIELBASIS(≤36)
Z=16
UMZUWANDELNDE
ZAHL(LAENGE≤10)
X=16777215
    
```

```

BASIS 10
16777215

BASIS 16
FFFFFF
    
```

```

QUELLBASIS(≤36)
Q=3
ZIELBASIS(≤36)
Z=8
UMZUWANDELNDE
ZAHL(LAENGE≤21)
X=10021221011
    
```

```

BASIS 3
10021221011

BASIS 8
176477
    
```

```

QUELLBASIS(≤36)
Q=16
ZIELBASIS(≤36)
Z=36
UMZUWANDELNDE
ZAHL(LAENGE≤8 )
X=AFFE
    
```

```

BASIS 16
AFFE

BASIS 36
VRI
    
```

```

QUELLBASIS(≤36)
Q=36
ZIELBASIS(≤36)
Z=2
UMZUWANDELNDE
ZAHL(LAENGE≤6 )
X=KARLO1
    
```

```

BASIS 36
KARLO1

BASIS 2
1001001001010001
100000001110001
    
```

```

QUELLBASIS(≤36)
Q=10
ZIELBASIS(≤36)
Z=20
UMZUWANDELNDE
ZAHL(LAENGE≤10)
X=2538610770
    
```

```

BASIS 10
2538610770

BASIS 20
1JD666IA
    
```