Darstellung natürlicher Zahlen in einem Stellensystem zur Basis *s*

Sei $s \in \mathbb{N}$, $s > 1$.
Zu jeder Zahl $a \in \mathbb{N}$ gibt es Zahlen $n \in \mathbb{N}_0$ [1]und $a_0, \ldots, a_n \in \mathbb{N}_0$ mit $a_i < s$ für $0 \leqslant i \leqslant n$, so daß

$$a = \sum_{i=0}^{n} a_i s^i$$

Die Zahl *a* ist also durch das Tupel $(a_n, \ldots, a_0)$ bestimmt; diese Darstellung ist bei vorgegebener „Basis" *s* eindeutig, wenn man keine führenden Nullen zuläßt, wenn also $a_n > 0$. Führende Nullen bei einer Zahldarstellung sind aber ansonsten kein Problem. Daß wir a_0 rechts und a_n links schreiben und nicht umgekehrt ist reine Willkür.

Offenbar besitzt die Zahlenmenge $\{0, \ldots, s-1\}$ genau *s* Elemente. Wenn man für jedes dieser Elemente ein eigenes Symbol (Ziffer) zur Verfügung hat, so können wir das Tupel $(a_n, \ldots, a_0)$ mit dem „Ziffernstring" $a_n \cdots a_0$ und letztlich diesen String mit der Zahl *a* identifizieren.

Im Dezimalsystem benutzen wir die Ziffernsymbole 0,1,2,3,4,5,6,7,8,9; im Hexadezimalsystem zusätzlich die Symbole A,B,C,D,E,F . Falls betont werden muß, daß der String $a_n \cdots a_0$ im Dezimalsystem interpretiert werden soll, stellen wir ein „D" voran, schreiben also $\mathrm{D}\, a_n \cdots a_0$; entsprechend ein „H" bei Hexadezimalstrings oder ein „B" bei Binärstrings.

Mit diesen Verabredungen läßt sich z.B. schreiben: D507=H1FB=B111111011, d.h. jeder der drei Ausdrücke stellt dieselbe natürliche Zahl dar.

Die arithmetischen Operationen wie Addition und Multiplikation müssen jetzt in Operationen mit den zugehörigen Ziffernstrings übersetzt werden; dabei übersetzen sich die Algorithmen, die wir in der Schule für das Rechnen im Dezimalsystem gelernt haben, direkt in die entsprechenden Operationen bei anderen Basen.

Wir können jede durch einen String $a_n \cdots a_0$ zur Basis *s* gegebene Zahl leicht ins Dezimalsystem umrechnen, indem wir in letzterem die Summe $\sum_{i=0}^{n} a_i s^i$ auswerten. Hier eine Konversion Hex-Dec:
$\mathrm{H\,1FB} = 1 \cdot 16^2 + 15 \cdot 16 + 11 = 256 + 240 + 11 = 507$. Eleganter ist es, wenn wir schreiben:
H1FB= $(1 \cdot 16 + 15) \cdot 16 + 11 = 31 \cdot 16 + 11 = 496 + 11 = 507$.

Ist durch den String $a_n \cdots a_0$ die Zahl $a = \sum_{i=0}^{n} a_i s^i$ gegeben, so gilt offenbar $a = \left(\sum_{i=1}^{n} a_i s^{i-1} \right) s + a_0$.

Dividiert man jetzt *a* durch *s*, so bleibt gerade der Rest a_0 . Der Quotient ist $\sum_{i=1}^{n} a_i s^{i-1} = \sum_{i=0}^{n-1} a_{i+1} s^i$, was dem verkürzten String $a_n \cdots a_1$ entspricht. Somit kann man sukzessive von rechts alle „Ziffern" aus *a* extrahieren. Formuliert man dies in der Programmiersprache Pari und definiert die Funktion `sys(n,s)=while(n,print(n%s);n=n\s)`, so erhält man mit dem Aufruf `(sys(507,16)` den Output `11 15 1,` woraus man unschwer in umgekehrter Reihenfolge die Ziffernfolge der Hex-Zahl 1FB erkennt.

Diese Rechnung kann man auch „zu Fuß" durch wiederholte Division mit Rest nachvollziehen, der Bequemlichkeit halber schreiben wir die Zahlen der Nebenrechnung im Dezimalsystem:
$507 = 31 \cdot 16 + 11$; $31 = 1 \cdot 16 + 15$; mit 11, 15, 1 hat man die „Hex-Ziffern" 1,F,B beisammen!

---

1 $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$

Die Konversionen Hex-Binär und Binär-Hex sind besonders einfach und erfordern praktisch keine Rechnung:

Man schreibe jede Hex-Ziffer als 4-stellige Binärzahl, also

```
H0=B0000,H1=B0001,H2=B0010,H3=B0011,H4=B0100,H5=B0101,H6=B0110,H7=B0111
H8=B1000,H9=B1001,HA=B1010,HB=B1011,HC=B1100,HD=B1101,HE=B1110,HF=B1111
```

Ist nun eine Zahl als Hexziffernfolge gegeben, so ersetze man jede Hexziffer nach obiger Liste und streiche am Schluß ggf. führende Nullen, z.B.

$$1FB \rightarrow 0001'1111'1011 \rightarrow 111111011$$

Geht man umgekehrt von einem Binärstring aus, so streiche man von rechts Vierergruppen ab und fülle ggf. am linken Rand Nullen auf, bis auch hier eine Vierergruppe voll ist; anschließend ersetze man jede Vierergruppe durch die entsprechende Hexziffer:

$$111111011 \rightarrow 1'1111'1011 \rightarrow 0001'1111'1011 \rightarrow 1FB$$