

Umwandlung: Beliebig nach Dezimal (Horner)

Schreiben Sie eine Funktion `anyToDez(a, basis)`, die Ziffern `a` als String, und eine Basis `basis` als Integer entgegennimmt und dann die entsprechende Zahl in der Dezimaldarstellung zurückgibt. Nutzen Sie zur Auswertung des Polynoms das Horner-Schema. Es genügt hierbei, wenn Zahlensysteme bis einschließlich der Basis 16 erlaubt sind.

Beispiel

```
a = 'f', basis = 16
```

```
>>> anyToDez(a, basis) = 15
```

Solution:

```
1  #!/usr/bin/env python
2  # coding: utf-8
3  # <h1>Table of Contents<span class="tocSkip"></span></h1>
4  # <div class="toc"><ul class="toc-item"><li><span><a href="#Umwandlung-von-einem-Zahlensystem-
   in-Dezimalsystem" data-toc-modified-id="Umwandlung-von-einem-Zahlensystem-in-Dezimalsystem
   -1"><span class="toc-item-num">1&nbsp;&nbsp;&nbsp;</span>Umwandlung von einem Zahlensystem in
   Dezimalsystem</a></span></li></ul></div>
5  # # Umwandlung von einem Zahlensystem in Dezimalsystem
6  def horner(z, a):
7      n = len(a)
8      x = a[n-1]
9      for k in range(1,n):
10         x = z*x + a[n-1-k]
11     return x
12 def anyToDez(a, basis):
13     """
14     Converts a representation a in a given basis to its representation as decimal number.
15
16     :param a: Representation as string, ex: a = 'fff'
17     :param basis: Basis, ex: basis = 16.
18
19     :return: Dezimal number, int
20     """
21     digitList = '0123456789abcdefghijklmnopqrstuvwxyz'
22     if basis > len(digitList):
23         raise ValueError("Sorry, a basis larger than 36 is not supported.")
24     aList = [digitList[:basis].find(digit) for digit in str(a)[::-1]] # this is list
   comprehension
25     for a in aList:
26         if a < 0:
27             raise ValueError("Some coefficients exceed the basis.")
28     return horner(basis, aList)
29 a = anyToDez("f", 16)
30 a
```