Base-k Representation of Rational Numbers

Felix P. Muga II
Mathematics Department
Ateneo de Manila University
Quezon City, Philippines

ABSTRACT

We shall show in this paper that every rational number has an infinite and periodic base-k representation. We shall also implement the representation of a rational number into a base number using the Python programming language.

1. INTRODUCTION

Let N be a rational number. Then there exists integers s and $t \neq 0$ such that

$$|N| = \frac{s}{t}$$

By the Euclidean algorithm, we can find nonnegative integers a and s_0 uniquely such that

$$S = at + s_0$$

where $0 \leq s_0 < t$.

Hence, $|N| = a + \dfrac{s_0}{t}$.

1. Suppose $s_0 = 0$. Then $|N| = a$ is a nonnegative integer.

   (a) If $a = 0$, its base-k representation is $(0)_k$ where $k \geq 2$.

   (b) If $0 < a < k^m$ for some positive integer m, then by the division algorithm (see for example, [3]), a base-k representation on a is given by

   $$a = (a_1 a_2 \ldots a_m)_k$$

   where

   $$a = a_1 \cdot k^{m-1} + a_2 \cdot k^{m-2} + \ldots a_{m-1} \cdot k + a_m \quad (1)$$

   and $a_i \in \{0, 1, \ldots, k-1\}$ for $i = 1, 2, \ldots, m$.

   If $a_1 \neq 0$, the effective length of the base-k representation of a is m.

The base-k representation of a may be extended to $m+r$ by adding r zeros to the left of a_1. Thus,

$$a = [a_1 a_2 \ldots a_m]_k = [\underbrace{00\ldots0}_{r \text{ zeros}} a_1 a_2 \ldots a_m]_k$$

The following is an implementation to compute the base-k representation of a nonnegative integer using the Python Programming Language which was created by Guido van Rossum.

The Base-k Representation of an Integer

Python Program 1

```
def WholePartOf_BaseNumber(a,k):
    """
    The representation of number a base k
    """
     if a == 0:
        baseNumber = [0]
        return baseNumber
    else:
        baseNumber = []
        q = a
        while q > 0:
            q,coeff = divmod(q,k)
            baseNumber.insert(0,coeff)
        return baseNumber
```

2. Suppose $s_0 \neq 0$. Then $|N|$ consists of the whole part, the nonnegative integer, a and the fractional part $\dfrac{s_0}{t}$ where $0 < \dfrac{s_0}{t} < 1$.

   $\dfrac{s_0}{t}$ may not be in simplest form, *that is*, s_0 and t may not be relatively prime. It will be reduced by canceling their greatest common divisor.

   The Euclidean algorithm finds the greatest common divisor of two positive numbers. The algorithm in [2] is used in this paper using the following Python program.

The Euclidean Algorithm

Python Program 2

```
def gcdOf(a,b):
    """
    The greatest common divisor of
    two numbers a and b
    """
    if b == 0:
        return a
    else:
        return gcdOf(b,a%b)
```

Let $\frac{p}{q}$ be the simplest form of $\frac{s_0}{t}$.

Thus, p and q are relatively prime integers and $0 < p < q$.

If $s_0 \neq 0$, we can write $|N|$ as

$$|N| = a + \frac{p}{q} \tag{2}$$

where $0 < p < q$.

Again by the division algorithm, we have nonnegative integers b_1 and p_1 such that

$$\begin{aligned}\frac{p}{q} &= \frac{1}{k}\left(\frac{kp}{q}\right)\\ &= \frac{1}{k}\left(b_1 + \frac{p_1}{q}\right)\end{aligned}$$

where $0 \leq p_1 < q$.

Since $0 < \frac{p}{q} < 1$, we have

$$\begin{aligned}0 < &\frac{p}{q} < 1\\ 0 < &\frac{kp}{q} < k\\ 0 < &b_1 + \frac{p_1}{q} < k\end{aligned}$$

Since b_1 is an integer, it follows that $b_i \in \{0, 1, \ldots, k-1\}$.

By repeating the division algorithm several times, we write $\frac{p}{q}$ as

$$\begin{aligned}\frac{p}{q} &= \frac{1}{k}\left(b_1 + \frac{1}{k}\left(b_2 + \frac{1}{k}\left(\ldots + \frac{1}{k}\left(b_n + \frac{p_n}{q}\right)\right)\right)\right)\\ &= b_1 \cdot k^{-1} + b_2 \cdot k^{-2} + \ldots + b_n \cdot k^{-n} + \frac{1}{k}\left(\frac{p_n}{q}\right)\end{aligned} \tag{3}$$

where $b_i \in \{0, 1, \ldots, k-1\}$ for $i = 1, 2, \ldots, n$.

(a) If $p_n = 0$ and $b_n \neq 0$, then the base-k representation of $\frac{p}{q}$ is given by

$$\frac{p}{q} = [0.b_1 b_2 \ldots b_n]_k$$

The expansion is finite and its effective length is n.

Hence, the base-k representation of $|N|$ is finite and it is given by

$$|N| = [a_1 a_2 \ldots a_m . b_1 b_2 \ldots b_n]_k$$

The expansion of N may be extended by adding 0 to the right of b_n where $b_n \neq 0$, say,

$$N = [a_1 a_2 \ldots a_m . b_1 b_2 \ldots b_n 00 \ldots 0]_k$$

Thus, the expansion may be infinite and periodic of period 1. But, this is a trivial periodic expansion.

(b) If there does not exist a positive integer n such that $p_n = 0$, then the binary expansion is infinite. If it is infinite, then it is a nontrivial periodic expansion.

Theorem 1 *If the base-k representation of $\frac{p}{q}$ is finite with effective length of n, then $q = k^n$.*

Proof. Suppose that $\frac{p}{q} = [b_1 b_2 \ldots b_n]_k$ where $b_n \neq 0$ and $b_i = 0$ for all $i > n$.

Then

$$\begin{aligned}\frac{p}{q} &= [b_1 b_2 \ldots b_n]_k\\ &= b_1 \cdot k^{-1} + b_2 \cdot k^{-2} + \ldots + b_{n-1} \cdot k^{-n+1} + b_n \cdot k^{-n}\\ &= \frac{b_1 \cdot k^{n-1} + b_2 \cdot k + \ldots + b_{n-1} \cdot k + b_n}{k^n}\\ &= \frac{[b_1 b_2 \ldots b_n]_k}{k^n}\\ \frac{p}{q} &= \frac{b}{k^n}\end{aligned}$$

where $b = [b_1 b_2 \ldots b_n]_k$.

Since $0 < b_n < k-1$, the base-k representation $b = b_1 \cdot k^{n-1} + b_2 \cdot k + \ldots + b_{n-1} \cdot k + b_n$ is relatively prime to k.

Hence, b and k^n are relatively prime. This means that the fraction $\frac{b}{k^n}$ is in simplest form.

Therefore, $p = b$ and $q = k^n$. □

2. INFINITE BASE-K REPRESENTATION

In this section, we shall show that a rational number may be expressed as a nonterminating (infinite) and periodic base-k number written in the form

$$|N| = [a_1 a_2 \ldots a_m . c_1 c_2 \ldots c_r (b_1 b_2 \ldots b_n)^\omega]_k$$

$a_1 a_2 \ldots a_m$ is the whole part of the base-k representation of N and $c_1 c_2 \ldots c_r (b_1 b_2 \ldots b_n)^\omega]_k$ is its fractional part.

The subsequence $c_1 c_2 \ldots c_r$ is called the *preperiod* and the subsequence $b_1 b_2 \ldots b_n$ is called the *period* of the fractional part of the base-k representation of $|N|$. The length of the preperiod is r and the length of the period is n.

The subsequence $b_1 b_2 \ldots b_n$ is repeating indefinitely. Hence, the symbol $(\cdots)^{\omega}$.

A periodic expansion is trivial if the bits in the periodic subsequence are all zeros or all ones.

The number zero has a trivial nonterminating and repeating base-k representation. It is given by

$$(0)_2 = (\overline{0})_2$$

Theorem 2 *The base-k representation of 1 may be periodic of length 1. It is given by*

$$(1)_k = (0.\overline{k-1})_k$$

PROOF. Let $x = (0.\overline{k-1})_k$. Then $kx = (k-1.\overline{k-1})_k$.

Hence, $kx - x = (k-1.\overline{k-1})_k - (0.\overline{k-1})_k = (k-1)_k$. Thus, $(k-1)x = (k-1)_k$. Therefore, $x = (1)_k = (\overline{k-1})_k$. ☐

Theorem 3 *The following infinite and periodic base-k representations are trivial.*

1. $k^{-n} = (0.\underbrace{00\ldots0}_{n\ zeros}\overline{1})_k$
2. $(a_n a_{n-1} \ldots a_1 a_0)_k = (b_n b_{n-1} \ldots b_1 b_0.\overline{k-1})_k$ *where* $(b_n b_{n-1} \ldots b_1 b_0)_2 = (a_n a_{n-1} \ldots a_1 a_0)_k - (1)_k$.
3. $(0.b_1 b_2 \ldots b_{n-1} b_n)_k = (0.b_1 b_k \ldots b_{n-1}(b_n - 1)\overline{k-1})_k$

PROOF. The theorem follows immediately from the previous theorem. ☐

Definition 1
Let q be a positive integer and $\mathbb{U}_n$ be the group of all positive integers less than q that are relatively prime to q. If $a \in \mathbb{U}_q$, then n_0 is the order of a under modulo q (or the order of a in $\mathbb{U}_q$) if and only n_0 is the smallest positive integer such that $a^{n_0} \equiv 1 \pmod q$.

The following result follows immediately from the definition.

Lemma 1 *If $gcd(k, q) = 1$, then there exists a positive integer b such that*

$$bq = k^{tn_0} - 1 \tag{4}$$

where n_0 is the order of k under modulo q and $t \in \mathbb{Z}^+$.

Computing the order of k under modulo a number that is relatively prime k is straightforward. Below is a Python program to compute the order of k and the value k^n such that $k^n \equiv 1 \pmod q$ which we call *unity*.

Order and Unity of k under Modulo q

Python Program 3

```
def unityAndOrderOf(k,q):
    if gcdOf(k,q) > 1:
        return 'The numbers are not relatively prime!'
    else:
        unity,  n = k,  1
        while unity%q > 1:
            unity = unity*2
            n = n+1
        ou = [n,unity]
        return ou
```

Prime factorization of a positive number can be implemented using the Python program. If the prime factorization of n is

$$n = p_1^{k_1} p_2^{k_2} \ldots p_n^{k_n}$$

then the output of the program is given by

```
[[p1,k1],[p2,k2],...,[pn,kn]]
```

Prime Factorization of a Positive Number

Python Program 4

```
def primeFactorizationOf(n):
    max_iter = int(n**0.5)+1
    q = n
    factors = {}
    k = 1
    while k <= max_iter:
        k += 1
        if q%k == 0:
            exponent = 1
            q = q/k
            while q%k == 0:
                exponent += 1
                q = q/k
            factors[k]=exponent

    if q > 1:
        factors[q]=1

    return factors
```

Theorem 4 *Let $\frac{p}{q}$ be a rational number in the open interval $(0, 1)$ such that p and p are relatively prime positive integers and let $k = k_1^{d_1} k_2^{d_2} \ldots k_t^{d_t}$ be the prime factorization of k.*

1. *If $gcd(k, q) = 1$, then there exists a positive integer b such that*

$$\frac{p}{q} = \frac{b}{k^n - 1} = [0.(b_1 b_2 \ldots b_n)^{\omega}]_k$$

where n is the integral multiple of the order of k under modulo q and $b = [b_1 b_2 \dots b_n]_k$.

2. *Suppose $gcd(k,q) > 1$.*

   Then $q = k_{r_1}^{e_{r_1}} k_{r_2}^{e_{r_2}} \dots k_{r_s}^{e_{r_s}} q_0$ where $k_{r_1}, k_{r_2}, \dots, k_{r_s}$ are prime factors of k, and $gcd(k, q_0) = 1$.

   (a) *If $q_0 = 1$ then there exists a positive integer b such that*

   $$\frac{p}{q} = \frac{b}{k^r} = [0.b_1 b_2 \dots b_r]_k = [0.b_1 b_2 \dots b_{r-1} 0 (k-1)^\omega]_k$$

   where $r = \max\left\{1, \left\lceil \frac{e_{r_1}}{d_{r_1}} \right\rceil, \left\lceil \frac{e_{r_2}}{d_{r_2}} \right\rceil, \dots, \left\lceil \frac{e_{r_s}}{d_{r_s}} \right\rceil \right\}$ and $b = [b_1 b_2 \dots b_r]_k$.

   (b) *If $q_0 > 1$, then there exist positive integers b and c such that*

   $$\frac{p}{q} = \frac{1}{k^r}\left(c + \frac{b}{k^n - 1}\right) = [0.c_1 c_2 \dots c_r (b_1 b_2 \dots b_n)^\omega]_k$$

   where n is an integral multiple of the order of k under modulo q_0, $c = [c_1 c_2 \dots c_r]_k$, $b = [b_1 b_2 \dots b_n]_k$ and $r = \max\left\{1, \left\lceil \frac{e_{r_1}}{d_{r_1}} \right\rceil, \left\lceil \frac{e_{r_2}}{d_{r_2}} \right\rceil, \dots, \left\lceil \frac{e_{r_s}}{d_{r_s}} \right\rceil \right\}$.

PROOF. **Case 1.** Suppose $gcd(k,q) = 1$.

Then by **Lemma 4** there exists a positive integer b such that

$$q b_0 = k^n - 1 \tag{5}$$

where n is an integral multiple of the order of k under modulo q.

Multiplying **Equation 5** by $\frac{p}{(k^n - 1)\,q}$, we have $\frac{p}{q} = \frac{b}{k^n - 1}$. where $b = p b_0$.

Thus,

$$\begin{aligned}
\frac{p}{q} &= \frac{b}{k^n - 1} \cdot \frac{k^n}{k^n} \\
&= \frac{b}{k^n} \cdot \frac{k^n}{k^n - 1} \\
&= \frac{b}{k^n}\left(1 + \frac{1}{k^n - 1}\right) \\
&= \frac{b}{k^n} + k^{-n}\left(\frac{b}{k^n - 1}\right) \\
&= \frac{b}{k^n} + k^{-n}\left(\frac{b}{k^n} + k^{-n}\left(\frac{b}{k^n - 1}\right)\right) \\
&= \frac{b}{k^n} + \frac{b}{k^{2n}} + k^{-2n}\left(\frac{b}{k^n - 1}\right) \\
\frac{p}{q} &= \frac{b}{k^n} + \frac{b}{k^{2n}} + \frac{b}{k^{3n}} + \dots
\end{aligned}$$

Therefore, if $(k,q) = 1$,

$$\frac{p}{q} = [0.(b_1 b_2 \dots b_n)^\omega]_k$$

where $b = b_0 p = [b_1 b_2 \dots b_n]_k$, $b_0 = \frac{k^n - 1}{q}$, and

n is an integral multiple of the order of k under modulo q.

Since $0 < \frac{p}{q} < 1$, it follows that $0 < \frac{b}{k^n - 1} < 1$. Hence, $b_i \neq 0$ and $b_i \neq k - 1$ for all $i = 1, 2, \dots, n$.

Therefore, the periodic expansion is not trivial.

Case 2. Suppose $gcd(k,q) > 1$.

Then $q = k_{r_1}^{e_{r_1}} k_{r_2}^{e_{r_2}} \dots k_{r_s}^{e_{r_s}} q_0$ where $k_{r_1}, k_{r_2}, \dots, k_{r_s}$ are prime factors of k, and $gcd(k, q_0) = 1$.

Case 2.1. Let $q_0 = 1$. Suppose that

$$r = \max\left\{1, \left\lceil \frac{e_{r_1}}{d_{r_1}} \right\rceil, \left\lceil \frac{e_{r_2}}{d_{r_2}} \right\rceil, \dots, \left\lceil \frac{e_{r_s}}{d_{r_s}} \right\rceil \right\}$$

Since $k = k_1^{d_1} k_2^{d_2} \dots k_t^{d_t}$ and $q = k_{r_1}^{e_{r_1}} k_{r_2}^{e_{r_2}} \dots k_{r_s}^{e_{r_s}}$, it follows that q is a factor k^r.

Let $k^r = k_0 q$.

Then $\frac{p}{q} = \frac{k_0 p}{k^r} = \frac{b}{k^r}$ where $b = k_0 p$. Since $0 < p < q$, it follows that $0 < b < k^r$.

Thus, $b = [0.b_1 b_2 \dots b_{r-1} b_r]_k$ for some $b_1, b_2, \dots b_{r-1}, b_r \in \{0, 1, \dots, k-1\}$.

Hence, $\frac{p}{q} = \frac{b}{k^r} = [0.b_1 b_2 \dots b_{n-1} 0 (k-1)^\omega]_k$.

Case 2.2. Let $q_0 > 1$. Suppose that $q = q_1 q_0$ where $q_1 = k_{r_1}^{e_{r_1}} k_{r_2}^{e_{r_2}} \dots k_{r_s}^{e_{r_s}}$.

Also, we assume that $r = \max\left\{1, \left\lceil \frac{e_{r_1}}{d_{r_1}} \right\rceil, \left\lceil \frac{e_{r_2}}{d_{r_2}} \right\rceil, \dots, \left\lceil \frac{e_{r_s}}{d_{r_s}} \right\rceil \right\}$.

Then $k^r = k_0 q_1$.

Thus, $\frac{p}{q} = \frac{k_0 p}{k_0 q_1 q_0} = \frac{k_0 p}{k^r q_0}$.

By the Euclidean algorithm, there exists nonnegative integers c and p_0 such that

$$k_0 p = c q_0 + p_0$$

where $0 \leq p_0 < q_0$.

We have

$$\frac{p}{q} = \frac{1}{k^r}\left(c + \frac{p_0}{q_0}\right) \tag{6}$$

Since q is not a power of k, then $p_0 \neq 0$. Thus, $1 \leq p_0 < q_0$.

Then by **Lemma 5**, there exists positive integer b_0 such that

$$q_0 b_0 = k^n - 1 \tag{7}$$

where n is an integral multiple of the order of k under modulo q_0.

Multiplying **Equation 7** by $\frac{p_0}{(k^n-1)\,q_0}$, we obtain

$$\frac{p_0}{q_0} = \frac{b}{k^n-1}$$

where $b = b_0 p_0$.

Thus,

$$\frac{p_0}{q_0} = \frac{b}{k^n-1} = \frac{b}{k^n}\left(1+\frac{1}{k^n-1}\right)$$

As in the proof of **Case 1**, since q_0 is relatively prime to k and is greater than 1, we obtain

$$\frac{p_0}{q_0} = \frac{b}{k^n-1} = \frac{b}{k^n} + \frac{b}{k^{2n}} + \frac{b}{k^{3n}} + \ldots = [0.(b_1 b_2 \ldots b_n)^{\omega}]_k$$

where $b = [b_1 b_2 \ldots b_n]_k$.

Hence, **Equation 6** becomes

$$\frac{p}{q} = \frac{1}{k^r}\left(c + \frac{b}{k^n-1}\right) = \frac{1}{k^r}\left(c + [0.(b_1 b_2 \ldots b_n)^{\omega}]_k\right)$$

Since $0 < \frac{p}{q} < 1$, we have $c < k^r$.

Hence, $c = [c_1 c_2 \ldots c_r]_k$ for some $c_i \in \{0, 1, \ldots, k-1\}$, for all $i = 1, 2, \ldots, r$.

It follows that

$$\frac{p}{q} = \frac{1}{2^r}\,[c_1 c_2 \ldots c_r.(b_1 b_2 \ldots b_n)^{\omega}]_k$$

Therefore, if $q = k_{r1}^{e_{r1}} k_{r2}^{e_{r2}} \ldots k_{rs}^{e_{rs}} q_0$ where $gcd(k, q_0) = 1$ and $q_0 > 1$, we have

$$\frac{p}{q} = \frac{1}{k^r}\left(c + \frac{b}{k^n-1}\right) = [0.c_1 c_2 \ldots c_r (b_1 b_2 \ldots b_n)^{\omega}]_k$$

where $k^r = k_0 \cdot k_{r1}^{e_{r1}} k_{r2}^{e_{r2}} \ldots k_{rs}^{e_{rs}}$
$k_0 p = c q_0 + p_0$, $0 < p_0 < q_0$,
$b = b_0 p_0 = [b_1 b_2 \ldots b_n]_k$,
$c = [c_1 c_2 \ldots c_n]_k$, and
n is an integral multiple of the order of k under modulo q.

Since $0 < \frac{p_0}{q_0} < 1$, it follows that $0 < \frac{b}{k^n-1} < 1$.

Hence, $b_i \neq 0$ and $b_i \neq k-1$ for all $i = 1, 2, \ldots, n$.

Therefore, the periodic expansion is not trivial. □

The computation of the base-k representation of a positive rational number less than 1 is implemented using a Python program as shown below.

Fractional Part of a Base Number Given a Specified Length

Python Program 5

```
def FractionalPartOf_BaseNumber(b,n,k):
    baseNumber = WholePartOf_BaseNumber(b,k)
    m = n  - len(baseNumber)

    for i in range(m):
        baseNumber.insert(0,0)
    return baseNumber
```

---

3. CONCLUSION

Any rational number with a finite base-k representation can be expressed as a base-k number with a nonterminating and periodic expansion with period that is equal to 1. See **Theorem 3**.

In particular, an integer a with a finite expansion $[a_1 a_2 \ldots a_m]_k$ can be expressed as

$$a = [b_1 b_2 \ldots b_m.(k-1)^{\omega}]_k$$

where $[b_1 b_2 \ldots b_m]_k = [a_1 a_2 \ldots a_m]_k - [1]_k$.

Also, in **Theorem 4**, we have shown that any rational number $\frac{p}{q}$ where $0 < p < q$, p and q are relatively prime can be expressed as a base-k number with a nonterminating and periodic expansion with

- period that is equal to 1 if q is a product of prime factors of k, and with
- period that is equal to n if q is relatively prime to k or q is a product of prime factors of k and a positive integer greater than 1 that is relatively prime to k. The number n is an integral multiple of the order of k under modulo of the positive integer.

Combining these theorems, we have our final result.

Theorem 5 *Every rational number has an infinite and periodic base-k representation.*

This gives a proof to the claim of the book in [1] that the fractional part of every rational number can be expressed as

$$k^{-r}\left([a_1 a_2 \ldots a_r]_k + \frac{[a_{r+1} a_{r+2} \ldots a_{r+s}]_k}{k^s - 1}\right)$$

for some integers r, s with $r \geq 0$ and $s > 0$.

The following Python program computes the infinite and periodic base-k representation of a rational number. The input is a rational number of the form $\frac{s}{t}$ and the output is of the form

```
(sign)[aaaaaaa.ccccccc(bbbbbbb)^w]_k
```

where (sign) gives '+' if the given rational is positive, otherwise '-', aaaaaaa gives the base-k representation of the integral part of the rational number (bbbbbb) is the period of the fractional part of the base-k which is enclosed in parentheses ccccccc is the preperiod of the fractional part of the base-k number.

If $t = 0$, the output is 'No Expansion.!'.

Base-k Number Representation of s/t

Python Program 6

```
def expandBaseNumberOf(s,t,k):
    zero = '(0)'
    if t == 0:
        return 'No Expansion!'
    elif s == 0:
        return zero
    elif s*t < 0:
        sign = '(-)'
        s,t =abs(s),abs(t)
    else:
        sign = '(+)'

    a, p0 = s/t, s%t
    whole_part = WholePartOf_BaseNumber(a,k)
    trivial_part = WholePartOf_BaseNumber(a-1,k)
    # if p/q is an integer
    if p0 == 0:
        answer = sign+str(whole_part)+'_'+str(k)\
                 +' or '+sign+'('+str(trivial_part)\
                 +'.['+str(k-1)+']^w)_'+str(k)
        return answer

    d = gcdOf(p0,t)
    p, q = p0/d, t/d

    # if q is relatively prime to k and
    # is greater than 3
    if gcdOf(k,q) == 1:
        order,unity = unityAndOrderOf(k,q)
        denom = unity - 1
        b = p*(denom/q)
        period_part = FractionalPartOf_BaseNumber\
                      (b,order,k)
        answer = sign+'('+ str(whole_part)\
                 +'.'+str(period_part)+'^w)_'+str(k)
        return answer

    # if q is product of prime factors of k and
    # a number q0 such that gcd(q0,k)=1

    gcd = gcdOf(k,q)

    primefactors = primeFactorizationOf(gcd)

    q0 = q
    q1 = {}
    for prime in primefactors:
        q1_temp,r_temp = prime,0
        while q0%q1_temp == 0:
           r_temp += 1
           q0 = q0/q1_temp
        q1[q1_temp]=r_temp

    primesOf_k = primeFactorizationOf(k)
    ceiling = [1]
    for prime in q1:
        ceiling.append(int(ceil(q1[prime]\
                *1.0/primesOf_k[prime])))
    r = max(ceiling)

    # if q is a product of prime factors of k
    if q0 == 1:
        k0 = k**r/q
        fract_part = FractionalPartOf_BaseNumber\
                     (k0*p,r,k)
        preperiod = fract_part[0:r-1]
        preperiod.insert(r-1,fract_part[r-1]-1)
        answer = sign+'('+ str(whole_part)+'.'+\
                 str(fract_part)+ ')_'+str(k)+ \
                 ' or '+sign+'('+str(whole_part)\
                 +'.'+str(preperiod)+'['+str(k-1)\
                 +']^w)_'+str(k)
        return answer

    # if q0 > 1
    else:
        q1 = q/q0
        k0 = k**r/q1
        order,unity = unityAndOrderOf(k,q0)
        denom = unity - 1
        c,p0 = divmod(k0*p,q0)
        b = p0*(denom/q0)
        preperiod_part= FractionalPartOf_BaseNumber\
                        (c,r,k)
        period_part = FractionalPartOf_BaseNumber\
                      (b,order,k)
        answer = sign+'('+str(whole_part)+'.'+\
                 str(preperiod_part)+str(period_part)\
                 +'^w)_'+str(k)
        return answer
```

Example 1

Consider the rational number 2/3.

$$\frac{2}{3} = \frac{(1,0)_2}{2^2-1} = ([0].[1,0]^{\omega})_2$$

The output of expandBaseNumberOf(2,3,2) is given by '(+)([0].[1,0]^w)_2'.

Example 2
Consider the rational number $-5/36$.

$$\begin{aligned}
-\frac{5}{36} &= -\frac{1}{4}\left(\frac{5}{9}\right) \\
&= -\frac{1}{4}\left(\frac{5\cdot 7}{9\cdot 7}\right) \\
&= -\frac{1}{4}\left(\frac{35}{63}\right) \\
&= -\frac{1}{4}\left(\frac{(1,0,0,0,1,1)_2}{2^6-1}\right) \\
&= -\frac{1}{4}\left([1,0,0,0,1,1]^\omega\right)_2 \\
&= -([0].[0,0][1,0,0,0,1,1]^\omega)_2
\end{aligned}$$

The output of `expandBaseNumberOf(-5,36,2)` is given by
`'(-)([0].[0, 0][1, 0, 0, 0, 1, 1]^w)_2'`.

Example 3
Consider the rational number $23/36$.

$$\begin{aligned}
\frac{23}{36} &= \frac{1}{4}\left(\frac{23}{9}\right) \\
&= \frac{1}{4}\left(2+\frac{5}{9}\right) \\
&= \frac{1}{4}\left(2+\frac{35}{63}\right) \\
&= \frac{1}{4}\left(2+\frac{(1,0,0,0,1,1)_2}{2^6-1}\right) \\
&= \frac{1}{4}\left([1,0].[1,0,0,0,1,1]^\omega)_2\right) \\
&= ([0].[1,0][1,0,0,0,1,1]^\omega)_2
\end{aligned}$$

The output of `expandBaseNumberOf(23,36,2)` is given by
`'(+)([0].[1, 0][1, 0, 0, 0, 1, 1]^w)_2'`.

Example 4
Consider the rational number $\dfrac{25,275,000}{759,375}$.

$$\begin{aligned}
\frac{25,275,000}{759,375} &= \frac{(2,3,4,3,13,5,0)_{15}}{15^5} \\
&= ([2,3].[4,3,13,5,0])_{15} \\
&= ([2,3].[4,3,13,5])_{15} \\
&= ([2,3].[4,3,13,4][14]^\omega)_{15}
\end{aligned}$$

The output of `expandBaseNumberOf(25275000,759375,15)` is given by
`'(+)([2, 3].[4, 3, 13, 5])_15'` or
`'(+)([2, 3].[4, 3, 13, 4][14]^w)_15'`

Example 5
Consider the rational number $\dfrac{998,870,000}{-4}$.

$$\begin{aligned}
\frac{998,870,000}{-4} &= -\frac{249,717,500}{1} \\
&= -(1,1,8,10,6,0,4,10,10)_{11} \\
&= -([1,1,8,10,6,0,4,10,9].[10])^\omega]_{11}
\end{aligned}$$

The output of `expandBaseNumberOf(998870000,-4,11)` is given by

`'(-)[1, 1, 8, 10, 6, 0, 4, 10, 10]_11'` or
`'(-)([1, 1, 8, 10, 6, 0, 4, 10, 9].[10]^w)_3'`.

Acknowledgements
This research is being supported by a grant from the professorial chair funded by Ateneo de Manila University.

4. REFERENCES

[1] Jean-Paul Allouche and Jeffrey Shallit, *Automatic Sequences: Theory, Applications, Generalizations*, Cambridge University Press, 2003.

[2] Thomas H. Cormen, Charles E. Lesserson and Ronald L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.

[3] Kenneth H. Rosen, *Discrete Mathematics and Its Applications, 5th edition.* McGrawHill, 2004.

[4] John Zelle, *Python Programming: An Introduction to Computer Science.* Franklin, Beadle and Associates, Inc. 2004