COL226 Assignment 1

Student	Kavya Chopra
Entry number	2021CS10081

Terminology

1. List representation: The list representation of a string representing a base 10 number consists of all its digits represented in the form of an SML list. For instance "3456" will be represented as [3,4,5,6]. The algorithm of generating this from a list is trivial.

2. Processed list: Let the base 10 representation of a number be $a_0 a_1 .....a_k$. Then, if k is even, the processed list representation looks like [[a_0], [$a_1 a_2$].....[$a_{k-1} ak$]]. If k is odd, then we have [[$a_0 a_1$], [$a_2 a_3$].....[$a_{k-1} a_k$]].

Pseudocode

Algorithm 1 Find maximum digit

```
function findDigit(number, digit, dividend)
    x ← (number*10 + digit)*digit
    if x ≤ dividend then
        return (digit, x)
    else
        return findDigit(number, digit - 1, dividend)
    end if
```

Algorithm 2 Performing long division

```
function findRoot(immediate_dividend, square_root, further_dividend)
    x ← findDigit(2*square_root, 9, immediate_dividend)
    y ← immediate_dividend - x.second
    if further_dividend.empty() then
        return (square_root*10 + x.first, y)
    else
        return findRoot(100 * y+further_dividend[0], square_root*10 + x.first, further_dividend[1:])
    end if
```

Algorithm 3 Initiating long division

```
function isqrtld(string)
    x ← processedList(listRepresentation(string))
    return findRoot(x[0], 0, x[1:])
```

Proof

Base Case: A number has 0 digits, in which case the algorithm returns (0,0), or the number has 1 digit n, in which the algorithm returns (x,r) where x is the maximum digit such that $x^2 < n$, and $r = n - x^2$. We can see that x is indeed the integer square root of n, as by the maximality of x we have $x^2 < n < (x+1)^2$
Induction Hypothesis: Let us assume that for all $k < n$, this algorithm returns the correct value of (sqrt, remainder).
Inductive Step: We proceed by strong induction. We shall now induct on the number of digits of n. Let that be x. Clearly, $x >= 2$, else this reduces to a base case. Let $n = a_1 a_2 ....a_x$. Now consider the number $m = a_1 a_2 ....a_{x-2}$.

Evidently $m < n$, and in fact $n = 100m + a_{x-1}a_x$. Let isqrtld(m) = (p,q).
Claim 1: There exists numbers r and h such that $r = 10p + h$, where $0 <= h <= 9$ and isqrtld(n) = (r, $n - r^2$)
Proof: By the definition of isqrtld, we have $m = p^2 + q$. Clearly, we have $p^2 <= m < (p+1)^2$, so $p^2 <= m <= (p+1)^2 - 1$. Also, since $n = 100m + a_{x-1}a_x$, so $100p^2 <= 100m <= n < 100m + 100 <= 100(p+1)^2$, so $(10p)^2 <= n < (10p+10)^2$. So, $10p <= r < 10p + 10$, and $10p <= r <= 10p + 9$, proving our claim.
Claim 2: Finding the maximum digit d such that $(20p+d)d <= 100q + a_{x-1}a_x$ gives us the square root $r = 10p + d$
Proof: The maximality of d ensures that $(20p+d)d <= n < (20p+d+1)(d+1)$. For the sake of contradiction, let us assume $r = 10p + h$, where $h \neq d$. Then, we have $(10p+h)^2 <= n < (10p+h+1)^2$, and so $(20p+h)h <= n - 100p^2 < (20p+h+1)(h+1)$ and since $p^2 = m - q$, we have $(20p+h)h <= (n - 100m) + 100q = a_{x-1}a_x + 100q < (20p+h+1)(h+1)$. But $100q + a_{x-1}a_x$ is the new dividend for our computation, and h is the maximum such digit such that $(20p+d)d <= 100q + a_{x-1}a_x$. Consequently, $h = d$, and by contradiction, we're done.
Main Claim: The output of the long division is $(r, n - r^2)$
Proof: findDigit, by definition, computes the maximum digit h such that (20square_root$+h)h <=$ immediate_dividend. Consider the computation after the algorithm has finished computing the answer for the first $x - 2$ digits of n. It then calls findRoot($100 * q + a_{x-1}a_x$, p, []) Then, the immediate dividend for the last step is $100 * q + a_{x-1}a_x$, and square_root$= p$. So, findDigit computes the required d, and we get the square root $r = 10p + d$, and the remainder, denoted by t, is $100q + a_{x-1}a_x - (20p+d)d$. But note that $t + r^2 = 100q + a_{x-1}a_x + 100p^2 = 100m + a_{x-1}a_x = n$. Thus, $t = n - r^2$. Moreover, since there's no further dividend (evidenced by the empty list), the computation is completed, and we're done.
This completes our proof of correctness of the algorithm.