**Arithmetic Operations on decimal numbers**

Tanvi Patil (tsp130130)

Pradnya Mohite (pam130430)

**Abstract:**

The project aims to perform arithmetic operations on decimal numbers whose length is not limited. The only arithmetic operation that hasn’t been implemented is division. These decimal numbers are represented in the form of linked list.

**Problem Statement:**

Representation of a number: A number is represented in a LinkedList. Each LinkedList node contains a single digit from the number. The LinkedList stores the number from Least Significant Digit to Most Significant Digit.

The operations that are to be performed on numbers are:

1. strToNum() : This function accepts a number in the form of string and returns a LinkedList representing the number.
   1. Input : “x=325287”
   2. Output : x : 7 -> 8 -> 2 -> 5 -> 2 -> 3
2. numToStr() : This function accepts a linked list as input and returns the number that it represents in the form of string.
   1. Input : x : 3 -> 5 -> 8 -> 0 ->4
   2. Output : “40853”
3. Add() : Accepts two linked lists and returns their sum represented as linked list.
   1. Input : x : 3 -> 5 -> 6 y : 1 -> 2
   2. Output : 4 -> 7 -> 6
4. Sub() : Accepts two linked lists and returns their difference as linked list. If the first linked list in the parameter represents a number smaller than the second, then we return 0.
   1. Input : x : 5 -> 2 -> 3 -> 5 y : 6
   2. Output : 9 -> 1 -> 3 -> 5
5. Input : x : 6 y : 4 -> 7 -> 3
6. Output : 0
7. Multiply() : Accepts two linked lists and returns their product as linked list.
   1. Input : x : 9 -> 9 -> 9 y : 2
   2. Output : 8 -> 9 -> 9 -> 1
8. Power() : Accepts two linked lists and returns the exponential result as linked list.
   1. Input : x : 9 y : 2
   2. Output : 8 -> 1

**Pseudocode :**

1. strToNum():
   1. Initialize a linked list (Eg. Res)
   2. Traverse from end of string to start (Iterator i)
      1. Add the element at i to res
   3. Ret res.
2. numToStr():
   1. Initialize an empty string
   2. Traverse from end of linked list to its start (Iterator i)
      1. Add the linked list at position I to string
   3. Return the string
3. Add() :
   1. Initialize an empty linked list.
   2. Initialize carry to zero.
   3. Traverse up to the end of both the lists.
      1. Add the individual digits and the carry
      2. If the sum is greater than 9

Add (sum % 10) to the linked list

Set carry

Else

Add sum to linked list

Reset carry

* 1. If carry is set
     1. Add carry to list
  2. Return list

1. Sub() :
   1. Initialize an empty linked list.
   2. Reset carry
   3. Traverse up to end of the linked list
      1. If carry is set
         1. Value of list1 = value of list1 – 1
      2. If value of list1 < value of list 2
         1. If value of list 1 is 0
            1. Value of list1 = 9
            2. Set carry
         2. Value of list1 = value of list1 + 10
         3. Set carry
      3. Find difference
      4. Add diff to list
   4. If list 2 is not empty and carry is set
      1. Return 0
2. Multiplication():
   1. Initialize a linked list result to 0
   2. Traverse through every digit of list1 (Iterator i)
      1. Multiply list2 to digit. (Add i number of zeroes to the result)
      2. Add the above product to result
   3. Return result
3. Power():
   1. Initialize a linked list result to 0.
   2. If list2 is zero, return 1.
   3. While value of list2 is no zero
      1. prod = list1 \* list1
      2. result = result + prod
      3. decrement list2
   4. Return result

TSP130130\_LinkedListManipulationsDriver is the driver program which illustrates these functionalities.