MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

159.201 Algorithms & Data Structures
S1 2022

Assignment 4

Write a C++ program to implement the addition operation for **big numbers**. A big number is a positive, whole number that can be of any arbitrary size, and therefore will not necessarily fit into an existing C++ type.
You ***must*** implement each big number as a **List** of single decimal digits and you must also set up the template class List based on a *linked-list* (use the sample codes). You will need to define and implement the appropriate methods for big numbers.

Your program must be laid out in the following way:
Section A : the template class List (with extra methods if you wish to do so)
Section B : the class BigNumber (which includes a List of digits)
Section C : program (main plus functions)

The program ***must read in two big numbers from a txt file***, add them and save the result as another big number instance. Then all three big numbers must be displayed on the screen using the **format**:

```
3245678911234123400 0
+
651234431221341234 45
=
9758023223447535744 5
```

Notes:

- You must get the two big numbers from a 2 lines text file, one big number in each line (***get the sample code***)
- The "digits" in the List can be stored in any type you want: `int, short int, char` etc. You probably realise that any type is wasting a bit of memory, as you would only need 10 digits (even `char` can represent 256 things).
- Remember that when a sum of two nodes goes over the base size minus 1 (in base 10, any number that is bigger than 10-1, then you need to carry part of the result to the next node (grade school arithmetic).
- You need to **read** the big number as a string, but do not store the big number as a string, integer or float at any other stage of the program. You need to use your own function that **converts** a **string** representing a decimal number into a **List**. Every big number is represented by several base-10 ***digits***, one digit ***per node***.
- Once the addition operation is implemented, make sure to carry out the following tests (*shown here in one line due to limited space, for the program follow the format described above*):

000000000000 + 000000000001 = 000000000001	(a list with zeros)
99 + 1 = 100	(check that the carry works)
12345678901234567890 + 9876543210987654321 = 22222222112222222211	(check mix of digits)
99999999999999999999 + 99999999999999999999 = 199999999999999999998	(check that the carry goes all the way)
99999999999999999999 + 1 = 100000000000000000000	(same thing)
100000000000000000000 + 1 = 100000000000000000001	(make sure a single digit is added)

Use our virtual machine to test your submissions (host name **vm000296**). The input/output requirements are essential, please follow them carefully to avoid losing marks. Spaces matter and text is case sensitive.
After you are satisfied with the performance of your code as tested in the virtual machine, submit a **one source file** code on Stream by **Friday 8 of April 2022.** Your **name** and **ID number** must appear **on top of the file as comments.** If you are working in a group, then ***all*** names and IDs must appear on top of the file as comments, but you still need to **submit individually** in both the virtual machine and Stream.