**8.3 Assignment 3, due Thursday 2018-02-22, at 11:00pm**

For this assignment, you are to implement a data type supporting addition and multiplication of large non-negative integers.

8.3.1 Your task

The file [num.h](http://www.cs.yale.edu/homes/aspnes/classes/223/examples/2018/hw/3/num.h), shown below, defines the interface to the data type. Your job is to provide a matching num.c file that implements these functions. You may also implement any other functions that would be helpful, but to be safe it would be best to declare any extra functions static.

A Num represents a possibly very large non-negative integer, and can be initialized by supplying a null-terminated string of ASCII digits to the numCreate function. You will need to choose an appropriate representation for Nums that allows a reasonably efficient implementation of the remaining functions.

A test harness that you can use to try out your code can be found in [testNum.c](http://www.cs.yale.edu/homes/aspnes/classes/223/examples/2018/hw/3/testNum.c). These files are also available in the directory /c/cs223/Hwk3 on the Zoo.

#ifndef \_NUM\_H

#define \_NUM\_H

#include <stdio.h>

*/\**

*\* High-precision arithmetic on non-negative number in base 10.*

*\*/*

**typedef** **struct** num Num;

*/\* constructs a Num from a string \*/*

*/\* string contains representation of the number*

*\* in base 10, e.g. numCreate("314159");*

*\* Returns a null pointer (0) if the string contains any non-digits.*

*\* Leading zeros are OK: numCreate("012") parses as 12.*

*\* Empty string parses as 0 \*/*

Num \* numCreate(const char \*s);

*/\* Free all resources used by a Num \*/*

void numDestroy(Num \*n);

*/\* Get the value of the i-th least significant digit of a Num.*

*\* Returns 0 if i is out of range.*

*\* Example:*

*\* n = numCreate("12345");*

*\* numGetDigit(n, 0) == 5*

*\* numGetDigit(n, 3) == 2*

*\* numGetDigit(n, 17) == 0*

*\* numGetDigit(n, -12) == 0*

*\*/*

int numGetDigit(const Num \*n, int i);

*/\* add two Nums, returning a new Num \*/*

*/\* does not destroy its inputs, caller must destroy output \*/*

Num \* numAdd(const Num \*x, const Num \*y);

*/\* multiply two Nums, returning a new Num \*/*

*/\* does not destroy its inputs, caller must destroy output \*/*

Num \* numMultiply(const Num \*x, const Num \*y);

*/\* Print the digits of a number to file.*

*\* Do not print any leading zeros unless n is zero. \*/*

void numPrint(const Num \*n, FILE \*f);

#endif */\* \_NUM\_H \*/*

[examples/2018/hw/3/num.h](http://www.cs.yale.edu/homes/aspnes/classes/223/examples/2018/hw/3/num.h)

8.3.2 Submitting your assignment

Submit your assignment as usual with /c/cs223/bin/submit 3 num.c. You can use /c/cs223/bin/testit 3 public to run the public test script on your submitted assignment, which is also available in /c/cs223/Hwk3/test.public.

8.3.3 Clarifications added after the original assignment was posted

**2018-02-12**

You may assume that you will never have to deal with a number with more than 231 − 1 decimal digits. This is suggested by the use of int for the index in numGetDigit, although in principle numPrint could extract more digits than this. But we will just declare this officially.