Lab 2: Arbitrary precision integers

Aim: The aim of this lab class is to design and implement a arbitrary precision integer library for C. You are required to design the library before starting to code.

You should use the provided skeleton code for the final submission.

Task1:
You would first create a *bigint_t* (which is the data type for arbitrary precision integers) using a *suitable* data type. There are some sample code provided in *bigint.h* header file which you should modify according to your design.

In addition to *bigint.h* there is another header file called *helper.h* and a corresponding c file called *helper.c*. When implementing *bigint_t* one might need other data structures; of cause which would depend on your design. For example on might need a linked list or an array etc. These data structures are supposed to be implemented in the *helper.c* find and their interface should be defined in the *helper.h* file. You are completely free to decide what should go into the *helper.h* and *helper.c* files.

However you may decide to implement *bigint_t* they should provide the following functions are described in the *bigint.h* header.

Function header	Description
bigint_t new_bigint(int);	Create a new *bigint_t* and return the reference.
void free_bigint(bigint_t);	Free all the memory used by the given bigint.
int add(bigint_t sum, bigint_t a, bigint_t b);	Add the two *bigint_t* a and b and store the results in *bigint_t* sum. Should return 0 to indicate success.
void show_bigint(bigint_t);	Display the given *bigint_t.*

Task 2:
Your next task is to use your implementation to display the first 100 Fibonacci numbers. You may see the implementation of Fibonacci sequence given in *main.c* and modify the code to use *bigint_t* instead of *int* as done now.

Submission: You should use the provided skeleton code for the final submission. *Tar* the whole folder and submit before 20th July 2015.