

# Develop a C library of an integer of arbitrary length (intal)

## Problem Definition:

Develop a C-library of an integer of arbitrary length, let us call it as "intal" in short. The functionalities to be implemented in the library are declared in the header file given at the end of the document.

Library "intal", short for integer of arbitrary length, a library of nonnegative integers of arbitrary length. The given header file "intal.h" declares the functionalities the library is expected to provide except that there is no definition of the "intal" itself. That is left to the implementation file, which should declare the structure of the intal along with defining the functionalities declared in intal.h.

Client treats an intal (an integer of arbitrary length) as an object pointed by a pointer "void\*". An intal can be created by intal\_create() by providing a char string of a nonnegative integer provided in decimal digits. Some intals are created out of some functionalities like intal\_add(), which creates a new intal. A new intal created must have allocated a dynamic memory (may be by a malloc() call). Responsibility of destroying the intals created lies with the client by calling intal\_destroy(), which will free whatever memory allocated during the creation of intal. Client sees an intal as a "void\*". It could be a pointer to char array, int array, long int array, double array, or a struct array. There is no theoretical limit to the size of the integer, but memory limitations of the process (Operating System). If the OS allows, your library should be able to hold the largest prime number known, which is 23,249,425 digits long (as of Feb 2018).