

Handy Array Library – Basics

Work to do

You have to write functions which will serve as a mini reusable library meant to help you handle arrays of integers. Your code will be spread in the following 3 different files;

- **hal.h** will contain the headers (aka declarations) of all the functions we will implement. This file will be included by any code which is using these functions.
- **hal.c** will contain the definitions of all the functions we will implement. This file starts by including hal.h to have all the functions' headers.
- **tests.c** will be used to test whether the functions in the other files are working. This file will contain the main function which will contain code that you will invent to test each of the functions implemented in hal.c.

To start this project, let's implement the following functions in hal.c;

- **void intarray_display(const int data[], int size);**
This function displays the contents of the integer array **data** which size is represented by the parameter **size**.
- **void intarray_scan(int data[], int size);**
This function scans as many integers from the user as necessary to fill the integer array **data** of size **size**.

Example(s)

n/a

Hints

- n/a

Testing

For each function you write in hal.c, you should add progressively more code in the tests.c file in order to use this function and make sure it is working properly.

For instance, as soon as you are done writing the **intarray_scan** function, you should call it from your main. Once **intarray_display** is also developed, you will be able to you're your main display the array it just scanned, thus enabling you to compile your code and run the main to make sure both functions are working properly.

You will still keep track of the input you use when running your main function in order to constitute a test harness.

Input	Output	
Input array	Expected	Observed