

Module 7 – Module Abstraction and Portability in C

Tutorial Questions

Objectives

To practise algorithm development for array processing tasks, and to begin working with modules & interfaces.

Activities

1. Sketch out an algorithm that will
 - Enable the user to enter a series of integer values in any order they choose. Suppose there is a limit of at most 10 integers.
 - Those integer values should be stored into an array so that after each is added the array is fully ordered in ascending numerical order (i.e. the insert should involve a search and shuffle-shift).
 - After the user has finished entering the integers the algorithm should next display the integer values as they now appear in the array (i.e. in ascending order)
2. Propose a set of interface routines for an `IntList` type that could serve as the basis for implementing the algorithm above in a modular manner.
3. Provide some type declarations for `IntList` that "encapsulate" all that you need to manipulate an ordered array of integers. Use the following skeleton as your starting point.

```
typedef struct
{
    /* ... */
} IntList;
```

4. Write some C code for to implement each of the interface routines for your `IntList` type. Be careful not to allow your code to overrun the bounds of the array being used.
5. Write a function `main()` which uses your `IntList` type to perform the original task outlined in 1 (above).
6. OPTIONAL: Modify the output so that the ordered sequence of integers is displayed as a comma separated list, with sequences of consecutive numbers are displayed as a hyphenated pair. For instance,