Bubble Sort of Numbers



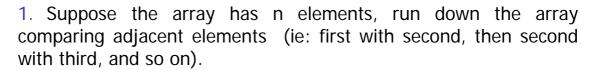


The listing here shows a well-written program (ExerciseB.cpp) to demonstrate the bubble sort algorithm. However, the actual sorting algorithm does not exist and you must write this yourself.



The algorithm can be described in English as follows:









2. If they are in the wrong order, swap them.



3. Repeat this process until you run through the whole array once without swapping anything. Then the array is sorted.



Read through the listing carefully and make sure that you understand everything. Ask a Lab Demonstrator if you have any question, no matter how trivial.

PYKC 1 Feb 2006

EE2/ISE1 Algorithms & Data Structures

Exercise B - Bubble Sort: Example of a good routine

Exercise B/2

Download from the course web page:

http://www.ee.ic.ac.uk/pcheung/teaching/ee2_software_engineering/

the source file *ExerciseB.cpp*, or type in the program yourself. The advantage of typing it in yourself is that you will be forced to think about the code carefully.

- Design the bubble sort routine in pseudocode.
- Code the routine and test it for various size of data arrays.
- Note that IN and OUT are replaced by nothing (i.e. removed) when the program is compiled. They are dummy keywords to make the code interface better documented.
- Apply the principles covered during the lecture in writing this routine.
- cMaxDataSize is defined as 100 in this code. You can alter just this single value to anything (say 100000) to change the upper limit for the entire program.

ExerciseB.cop

```
#include <stdlib.h>
#include <comio.h>
#include <iostream.h>
#define cMaxDataSize 100
#define TN
#define OUT
//-----
// Function to generate an array of random integer numbers
void GenerateData (
    IN int size.
                       // number of data values to generate
   OUT int *array
                       // data array returned as reference
                       // local index
   int i:
   randomize():
                       // function to initialize
                       //random number generator
    // Fill array with random values
   for( i=0; i<size; i++ ) {
       array[i] = rand();
// Function to output the array on console window
void DisplayArray (
   IN int size,
                       // number of data values to generate
   IN int *array
                       // data array containing data to display
   ) {
   int i;
                       // local index
    // Output data one value per line to console window
   for( i=0; i<size; i++ )
       cout << array[i] << endl;</pre>
   cout << endl;
                    // output a blank line
// Function to sort array into ascending order
void BubbleSort (
   IN int size,
                        // number of data values to sort
   IN OUT int *array // original data array on entry
                       // sorted array on exit
/* THE FOLLOWING THREE LINES ARE DUMMY CODE (called Stubs). */
/* These lines do not perform sorting, but allow the
/* rest of the code to be tested.
/* REPLACE this FOR-LOOP with your bubble sort code.
   int i;
   for ( i=0; i<size; i++ )
       array[i] = array[i];
// Main Program - does not take any input argument
int main()
int DataSize;
                               // no of data in arrays
int DataArray[cMaxDataSize];    // original data array
int SortedArray[cMaxDataSize];    // array with sorted data
    // input no of data required. must be less than maximum
   cout << "Enter size of array (0 - cMaxDataSize): ";</pre>
   cin >> DataSize;
    // if requesting too many data values, report error
   if (DataSize > cMaxDataSize)
           cout << "Data size too large - abort program" << endl:
    // otherwise if one or more data value exists
   else if (DataSize > 0) {
       // generate an array of random data
```

```
GenerateData( DataSize, DataArray );

// display original array in console window
cout << "Original Data\n";
DisplayArray( DataSize, DataArray );

// sort the data array
BubbleSort( DataSize, DataArray );

// display both original and sorted array in console window
cout << "Sorted Data\n";
DisplayArray( DataSize, DataArray );
}
cout << "Press any key to terminate"; getch();
return 0;
}</pre>
```