Arrays

1. As an introduction to arrays, write a program that declares an array of 10 integers, and initialises the elements as follows:

   ```
   element 0 is initialised to 100
   element 1 is initialised to 101
   ...
   element 9 is initialised to 109
   ```

   Using a for loop, print the value of each element in the array, one element per line.

2. Write a program that declares an array of 5 doubles, but this time do not initialise the array beforehand. Instead, ask the user to supply the 5 array values, using a for loop to read the values in.
   Using another for loop, print the value of each element in the array.

3. Modify your solution to question 2, so that the array elements are printed in REVERSE ORDER.

4. Write a program that declares an array of 12 integers, and asks the user to supply the initial values. The program should then loop through the elements to detect the largest number, and should print the largest value and its position in the array

   e.g. if the user enters

   ```
   10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2100
   ```

   then the largest number is 2100, which is element 11 in the array

5. Write a prototype weather reporting program which obtains month-by-month weather statistics from the user, and then displays the information in a table.

   You will need at least four arrays of 12 doubles, i.e. for each month the following values need to be stored, rainfall, maximum temperature, minimum temperature and number of hours of sunshine.

   You are advised to write functions, i.e. one to get these values from the user (called four times, one per each of the arrays, and three 'prettyprint' functions to display the rainfall/max and min temperatures, the temperature range and the sunshine hours.

6. Write a program that declares 2 strings (arrays of char), both initialised with a memorable phrase and then using a buffer array of char (make it enough), experiment with strlen, strcpy, strncpy and isdigit. We have provided a loop using some I/O functions for you to get/put char from/to keyboard and screen (covered in the later I/O chapter).

   For example, the strings could be:

   ```
   char string1[] = "Once upon a time there was a noble Prince";
   char string2[] = "The quick brown fox jumps over the lazy dog";
   ```

   and the I/O loop would look something like:

   ```
   while ( (c = getch())  != '!')
   {
       printf ("The character %c ", c);
       /* process the digit */
   }
   ```