This document is part of the HTML publication "**An Introduction to the Imperative Part of C++**"

The original version was produced by **Rob Miller** at **Imperial College London**, September 1996.

Version 1.1 (modified by **David Clark** at **Imperial College London**, September 1997)

Version 1.2 (modified by **Bob White** at **Imperial College London**, September 1998)

Version 1.3, 1.4, 2.0, ..., 2.15 (modified by **William Knottenbelt** at **Imperial College London**, September 1999-September 2016)

Introduction to C++ Programming: Exercise Sheet 6

Question 1

Write a library of integer array functions with a header file "IntegerArray.h" and implementation file "IntegerArray.cpp", which contains the following functions:

- A function "`input_array(a,n)`" which allows the user to input values for the first n elements of the array a.
- A function "`display_array(a,n)`" which displays the values of the first n elements of the array a on the screen.
- A function "`copy_array(a1,a2,n)`" which copies the first n elements of a2 to the respective first n elements in a1.
- A function "`standard_deviation(a,n)`" which returns the standard deviation of the first n elements of a. (The function "`average(a,n)`" in the lecture notes may help. A formula for the standard deviation of n values is given in Exercise Sheet 3, Question 3.)

Test the functions in a suitably defined main program.

(EXAMPLE ANSWER: main program, IntegerArray.h, IntegerArray.cpp) (BACK TO COURSE CONTENTS)

Question 2

Adapt the function "`selection_sort(...)`" in the lecture notes into a single argument string function "`string_sort(...)`" which sorts the characters in a string alphabetically (but putting all upper-case letters before all lower-case letters). The function should leave the position of the sentinel character unchanged. Test the function in a suitable main program, which should be able to reproduce the following input/output:

```
Type in a string: Rob Miller
The sorted string is:  MRbeillor
```

(EXAMPLE ANSWER) (BACK TO COURSE CONTENTS)

Question 3

Write a function "`no_repetitions(...)`" which removes all repetitions of characters from a string. Test the function in a suitable main program, which should be able to reproduce the

following input/output:

```
Type in a string: This string contains repeated characters
The string without repetitions is: This trngcoaepd
```

Hint: *Like most programming problems, this exercise is* ***much*** *easier if you use functional abstraction.*

(EXAMPLE ANSWER) (BACK TO COURSE CONTENTS)

Question 4

Using two-dimensional arrays, write a function (and a corresponding program to test it) which multiplies an mxn matrix of integers by an nxr matrix of integers. Use global constant declarations before the main program to give test values for m, n and r. Example input/output might be:

```
INPUT FIRST (2x2) MATRIX:
Type in 2 values for row 1 separated by spaces: 3 4
Type in 2 values for row 2 separated by spaces: 5 7
INPUT SECOND (2x2) MATRIX:
Type in 2 values for row 1 separated by spaces: 1 1
Type in 2 values for row 2 separated by spaces: 2 2

           3     4
           5     7
TIMES
           1     1
           2     2
EQUALS
           11    11
           19    19
```

(EXAMPLE ANSWER) (BACK TO COURSE CONTENTS)
