Question 1

Design Description:

The purpose of the application is to prompt a user for two integers, of either negative or positive signage and of maximum length 50 digits long, then subtract the second from the first to form a mathematical answer.

The method by which I chose to develop the application was by using a single array construction, allowing for both integers to be stored in the same construct. The items were segregated by index value signifying the beginning of the integer, with a length used to calculate the end point. With this design, the inclusion of classes was deemed surplus to requirement due to the simplicity.

The logic flow of the general use case is as follows:

-User is prompted for the first integer number.

-Input accepted as string & examined for validity.

-Should the input not meet requirements, the user is prompted to attempt another input.

(Process repeats until valid input.)

-String is examined for signage & result stored. (If signage present, it is discarded during array import).

-Process is repeated for the second integer.

-Integers are extracted from the array by index & length, then appended to a string. String converted to integer value and returned.

-Negative signage returned to the integer, if present before import.

-Calculation performed and returned to user.

Testing Strategy:

Testing involved inserting data that would be both expected, and unexpected by the application, as well as combinations of both. Validation was used to ensure that data being input was only processed if it was of the correct type. As subtracting a negative number results in a positive sum, this also had to be accounted for when performing unit tests.

Test Cases & Results:

The values listed in the table are a transcript of those input into the application at various stages & those that were returned:

Input #1	Input #2	Result
<blank>		Input rejected: User prompted for valid input.
0	<blank>	Input rejected: User prompted for valid input.
0	0	0
1	1	0
-1	4	-5
1	-4	5
abc		Input rejected: User prompted for valid input.
1	abc	Input rejected: User prompted for valid input.
1a4		Input rejected: User prompted for valid input.
1	1a4	Input rejected: User prompted for valid input.

A special case to note is the inclusion of validation for input format such as '1a4', as inferior forms of validation would allow the first integer character through, then discard the rest, or attempt to use this as input for a later user prompt.