

2 A program sorts the data in the 1D array `DataArray` and searches `DataArray` for specific values.

`DataArray` stores 14 integer values and is declared local to the main program.

(a) Write program code to create `DataArray` and initialise it with the following data values in the order they are written:

0 3 4 56 67 44 43 32 31 345 45 6 54 1

Save your program as **Question2_J25**.

Copy and paste the program code into part **2(a)** in the evidence document.

[1]

(b) The function `InsertionSort()` takes an array of integers as a parameter. The function sorts the data in the array into ascending numerical order using an insertion sort. The function returns the sorted array.

Write program code for `InsertionSort()`

You must **not** use any inbuilt sorting functions for your programming language.

Save your program.

Copy and paste the program code into part **2(b)** in the evidence document.

[5]

(c) The procedure `OutputArray()` takes an array of integers as a parameter. The procedure outputs each element in the array from the first element to the last element. The output is on one line with a space between each number.

An example output is:

"0 3 4 56 67 44 43 32 31 345 45 6 54 1"

Write program code for `OutputArray()`

Save your program.

Copy and paste the program code into part **2(c)** in the evidence document.

[2]

(d) (i) The main program needs extending to:

- output the content of the unsorted array using `OutputArray()`
- sort the array using `InsertionSort()`
- output the content of the sorted array using `OutputArray()`

Write program code to extend the main program.

Save your program.

Copy and paste the program code into part **2(d)(i)** in the evidence document.

[2]

(ii) Test your program.

Take a screenshot of the output(s).

Save your program.

Copy and paste the screenshot into part **2(d)(ii)** in the evidence document.

[1]

(e) The function `Search()` performs a binary search to find `ItemToFind` in `DataArray`

The function takes **two** parameters:

- `DataArray`, an array of integers
- `ItemToFind`, an integer to find in `DataArray`

The function returns:

- the index of `ItemToFind` if it is in `DataArray`
- `-1` if `ItemToFind` is **not** in `DataArray`

Write program code for `Search()`

You must **not** use any inbuilt searching functions for your programming language.

Save your program.

Copy and paste the program code into part **2(e)** in the evidence document.

[6]

(f) (i) The main program needs extending to call `Search()` with the sorted array **four** times:

- the first time to find the index of the integer 0
- the second time to find the index of the integer 345
- the third time to find the index of the integer 67
- the fourth time to find the index of the integer 2

If the integer is found in the array, output an appropriate message that includes the index. If the integer is **not** found, output that it was **not** found.

Write program code to extend the main program.

Save your program.

Copy and paste the program code into part **2(f)(i)** in the evidence document.

[2]

(ii) Test your program.

Take a screenshot of the output(s).

Save your program.

Copy and paste the screenshot into part **2(f)(ii)** in the evidence document.

[1]