2 A program stores 20 unique random integers between 0 and 100 (inclusive) in a 1D array that is local to the main program.

(a) Write program code to declare the array local to the main program and store 20 **unique** random numbers between 0 and 100 (inclusive) in the array.

Save your program as **Question2_N25**.

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

[3]

(b) The procedure `PrintArray()` takes an integer array as a parameter. The procedure outputs the array contents on a single line with a space between each integer.

Write the program code for `PrintArray()`

Save your program.

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

[3]

(c) The function `BubbleSort()`:

- takes an integer array as a parameter
- sorts the data into ascending order using a bubble sort
- returns the sorted array.

The function needs to work for an array of any length.

Do **not** use an inbuilt sorting method.

Write program code for `BubbleSort()`

Save your program.

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

[5]

(d) (i) The main program:

- outputs the contents of the array using `PrintArray()`
- sorts the array using `BubbleSort()`
- outputs `"Sorted"`
- outputs the contents of the sorted array using `PrintArray()`

Write program code for the main program.

Save your program.

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

[3]

(ii) Test your program.

Take a screenshot of the output(s).

Save your program.

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

[1]

(e) The recursive function `RecursiveBinarySearch()` takes four parameters:

- an integer array
- the lower bound of the array
- the upper bound of the array
- the value to find in the array.

The recursive function performs a binary search to find the index of the value in the array.

The function returns the index of the value if it is found. The function returns `-1` if the value is not found.

Write program code for `RecursiveBinarySearch()`

Save your program.

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

[6]

(f) (i) The main program:

- prompts the user to enter an integer
- takes the integer as input
- calls `RecursiveBinarySearch()` with the sorted array, appropriate lower bound, appropriate upper bound and the user's input as parameters
- outputs `"Not found"` if the input is not within the array
- outputs `"Found at position"` and the index if the input is within the array.

Write program code to amend the main program.

Save your program.

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

[3]

(ii) Test your program **three** times with each of the inputs described:

Test 1: the smallest number in the array

Test 2: the largest number in the array

Test 3: a number **not** in the array

Take a screenshot of each output.

Save your program.

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

[2]