1 A program reads data from a text file and stores it in a queue. The linear queue `Queue` is stored as a 1D array of up to 50 elements. The queue has the following pointers:

- `HeadPointer` – this stores the index of the first element in the queue, initialised to `-1`
- `TailPointer` – this stores the index of the last element in the queue, initialised to `-1`

(a) `Queue` is a global array of 50 integers with all elements initialised to `-1` in the main program.

The **two** pointers are declared as global variables and initialised to `-1`

Write program code to declare and initialise `Queue`, `HeadPointer` and `TailPointer`

Save your program as **Question1_J25**.

Copy and paste the program code into part **1(a)** in the evidence document.

[3]

(b) The function `Enqueue()`:

- takes an integer parameter to store in the next position in the queue
- returns `FALSE` if the queue is full and the parameter cannot be stored in the queue
- returns `TRUE` if the parameter is stored in the queue
- updates the pointers where appropriate.

Write program code for `Enqueue()`

Save your program.

Copy and paste the program code into part **1(b)** in the evidence document.

[6]

(c) The function `Dequeue()`:

- returns the next item in the queue, if the queue is **not** empty
- returns `-1` if the queue is empty
- updates the pointers where appropriate.

Write program code for `Dequeue()`

Save your program.

Copy and paste the program code into part **1(c)** in the evidence document.

[5]

(d) The text file `QueueData.txt` stores positive integers. Each integer is on a new line in the file.

The procedure `CreateQueue()`:

- opens the file `QueueData.txt`
- reads in each line from the text file and uses `Enqueue()` to insert each line into the queue
- outputs `"Queue full"` if any item cannot be inserted into the queue
- uses exception handling when opening and reading from the text file.

The procedure needs to work for a file that contains an unknown number of lines.

Write program code for `CreateQueue()`

Save your program.

Copy and paste the program code into part **1(d)** in the evidence document.

[6]

(e) **(i)** The main program needs extending to call `CreateQueue()`. The main program then adds together all of the integers stored in the queue, using `Dequeue()` to access each integer. The total is then output.

Write program code to extend the main program.

Save your program.

Copy and paste the program code into part **1(e)(i)** in the evidence document.

[5]

(ii) Test your program.

Take a screenshot of the output(s).

Save your program.

Copy and paste the screenshot into part **1(e)(ii)** in the evidence document.

[1]