8. An online game measures a player's reaction time to seeing a green dot appear on the screen.

Each time someone plays the game their name, e-mail address, attemptID and reaction time to the nearest thousandth of a second is stored.

Sample data is shown below.

Lucy Scott	lscott@hmail.com	lscott8	0.215
Deaglán Mag Uidhir	deaghlan@nsm.com	deaghlan9	0.125
...	...	...	...

The game allows a maximum of 10 000 attempts each day. Players can play more than once, and each time their attemptID will be different.

(a) (i) Using a programming language of your choice, define a suitable record structure to store a player's data. **2**

(ii) Using a programming language of your choice, declare a variable that could be used to store details for 10 000 attempts. Your answer should make use of the record data structure defined in part (i). **2**

8. (continued)

(b) The game stores the number of times it has been played that day in a variable `numPlays`. At the end of each day, the game finds the fastest time.

(i) Using a programming language of your choice, write the code to find the fastest time. Your answer should use the record data structure and variable declared in part (a). **5**

8. (b) (continued) **MARKS**

(ii) Explain why the variable `numPlays` is needed instead of traversing the entire array of 10 000. **1**

____

____

____

(c) The reaction time of 0.125 is stored as a binary number as:

0.001

Convert the binary number above into floating-point representation.

There are 16 bits for the mantissa (including the sign bit) and 8 bits for the exponent. **3**

Space for working

sign	mantissa	exponent

[Turn over

8. (continued)

(d) A player can play the game multiple times each day. Each time that they play, the program allocates them an `attemptID` using all the characters before the '@' in their e-mail and the value of `numPlays`.

A function that returns the position of any character in a string is used to implement this feature.

The code for this function is shown below.

```
…
Line 47   FUNCTION findCharIndex(STRING value, STRING
          character) RETURNS INTEGER
Line 48     DECLARE positionChar INITIALLY -1
Line 49     FOR index FROM 0 TO length(value)-1 DO
Line 50        IF value[index]= character THEN
Line 51             SET positionChar TO index
Line 52        END IF
Line 53     END FOR
Line 54     RETURN positionChar
Line 55   END FUNCTION
…
Line 70   SET position TO <return value of findCharIndex>
Line 71   SET attemptID TO <characters before the '@'
          character concatenated with the value of numPlays>
```

(i) Using a programming language of your choice, write the code for line 70 to assign the location of the '@' character in the `email` variable to `position` by calling the function `findCharIndex`. **2**

8. (d) (continued)

(ii) Using a programming language of your choice, write the code for line 71 to assign all of the characters before the '@' character concatenated with the value of the variable `numPlays` to `attemptID`. **2**