10. Lewis uses a smart watch to track the distance he walks each day for a week. His target is to walk 5 kilometres or more each day.

The program will find the highest the number of consecutive days Lewis meets this target.

For example, if the data was 5.2, 4.2, 4.0, 4.8, 5.8, 5.2, 6.4

The expected output should be 3.

The following code has been developed to find the highest number of consecutive days Lewis has met his target but there is an error.

```
…
Line 09   FUNCTION consecutiveDays (ARRAY OF REAL values)
          RETURNS INTEGER
Line 10     DECLARE counter INITIALLY 0
Line 11     DECLARE longestStreak INITIALLY - 1
Line 12     FOR index FROM 0 TO LENGTH(values) - 1 DO
Line 13       SET counter TO 0
Line 14       IF values[index] >= 5.0 THEN
Line 15          SET counter TO counter + 1
Line 16       ELSE
Line 17          IF counter > longestStreak THEN
Line 18             SET longestStreak TO counter
Line 19          END IF
Line 20          SET counter TO 0
Line 21       END IF
Line 22     END FOR
Line 23     IF counter > longestStreak THEN
Line 24          SET longestStreak TO counter
Line 25     END IF
Line 26     RETURN longestStreak
Line 27   END FUNCTION
…
Line 46   SET distances TO [5.2, 4.2, 4.0, 4.8, 5.8, 5.2, 6.4]
Line 47   SET daysMet TO consecutiveDays(distances)
…
```

10. (continued)

MARKS | DO NOT WRITE IN THIS MARGIN

(a) Complete the trace table for the first two iterations of the loop.

The trace table should indicate the line number where a variable changes value and the new value of that variable. **3**

Line number	counter	index	longestStreak
10	0		
11			−1
12		0	
13	0		
15			
12			
13			
18			
20			

(b) State the line number of the code that should be removed to correct the algorithm. **1**

(c) Name and describe a debugging technique that could be used during execution of the code. **2**

(d) Explain why the scope of the variable `counter` is local. **1**

MARKS DO NOT WRITE IN THIS MARGIN

10. (continued)

(e) Describe how modular programming increases the efficiency and maintainability of the code. **2**

Efficiency

Maintainability