2.2 Problem Solving and Programming: Test Pack

OCR H446 · Calculators not allowed · about 35 minutes

Total Marks

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 A local library is developing a simple program to manage overdue book fines.

(a) State the most appropriate primitive data type for storing each of the following: **(3)**

(i) the number of books a member has borrowed

(ii) whether a book is currently overdue

(iii) the title of a book

(b) A variable is declared as: **(2)**

```
DECLARE bookPrice : REAL
```

Explain why REAL is a more appropriate data type than INTEGER for storing the price of a book in pounds.

© Revision Library | revisionlibrary.co.uk

(c) A librarian writes the following algorithm to find the average fine charged, ignoring members with no fine: **(5)**

```
count <- 0
total <- 0
FOR i <- 1 TO 5
    INPUT fine
    total <- total + fine
    IF fine > 0 THEN
        count <- count + 1
    ENDIF
NEXT i
average <- total / count
OUTPUT average
```

The five values entered for fine, in order, are: 0, 2.50, 0, 1.20, 3.30

Complete the trace table below and state the final value output for average.

i	fine	total	count
1	0	___	___
2	2.50	___	___
3	0	___	___
4	1.20	___	___
5	3.30	___	___

(Total for Question 1 is 10 marks)

© Revision Library | revisionlibrary.co.uk

2 A weather station program reads temperatures until a sentinel value of -999 is entered, and finds the maximum temperature recorded:

```
maxTemp <- -999
INPUT temp
WHILE temp <> -999
    IF temp > maxTemp THEN
        maxTemp <- temp
    ENDIF
    INPUT temp
ENDWHILE
OUTPUT maxTemp
```

The values entered, in order, are: 14, 21, 9, 18, -999

(a) Complete a trace table showing the value of maxTemp after each temperature is processed, and state the final value output. **(5)**

(b) Explain what would happen if the very first value entered was -999, and why the output produced would be unhelpful. **(2)**

(c) A programmer rewrites the loop using REPEAT...UNTIL instead of WHILE. Write a correct REPEAT...UNTIL version of this algorithm, and explain one adjustment that had to be made compared with the WHILE version. **(3)**

(d) State the name given to the special value (-999) used to end the input loop. **(1)**

(Total for Question 2 is 11 marks)

© Revision Library | revisionlibrary.co.uk

3 A software company is developing a mobile app that manages a large, frequently-updated contact list (contacts are regularly added, deleted and searched for). The developers are deciding between using linear search or binary search, and between using a bubble sort or a merge sort, to organise and search the contact data.

Discuss the factors the company should consider when choosing appropriate algorithms and data structures for this application. Your answer should refer to time complexity, the need to keep data sorted for binary search, and the trade-offs of maintaining a sorted list when data changes frequently.

(Total for Question 3 is 8 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 2.2 Problem Solving and Programming: Test Pack

Question 1

(a) (i) INTEGER **B1**

(a) (ii) BOOLEAN **B1**

(a) (iii) STRING **B1**

(a) Answer: (i) INTEGER (ii) BOOLEAN (iii) STRING

(b) book prices can include a fractional/decimal part (pence), oe **B1**

(b) an INTEGER can only store whole numbers, so a price such as 4.99 would be stored inaccurately (e.g. truncated/rounded), oe **B1**

(b) Answer: REAL is needed because prices such as £4.99 have a decimal (pence) component that INTEGER cannot store accurately.

(c) i = 1: total = 0, count = 0 **B1**

(c) i = 2: total = 2.50, count = 1 **B1**

(c) i = 3 and i = 4: total = 2.50 then 3.70, count = 1 then 2 **B1**

(c) i = 5: total = 7.00, count = 3 **B1**

(c) average = 7.00 / 3 = awrt 2.33 (accept 7/3 exact), cao **B1**

(c) Answer: average = awrt 2.33

Question 2

(a) temp = 14: maxTemp becomes 14 **B1**

(a) temp = 21: maxTemp becomes 21 **B1**

(a) temp = 9: maxTemp stays 21 **B1**

(a) temp = 18: maxTemp stays 21 **B1**

(a) loop ends at temp = -999; OUTPUT is 21, cao **B1**

(a) Answer: 21

(b) the WHILE condition is false immediately, so the loop body never executes, oe **B1**

(b) maxTemp remains at its initial value of -999, which is output but is not a real/possible temperature reading, oe **B1**

(b) Answer: The loop would never run and -999 (an invalid placeholder, not a real temperature) would be output.

(c) correct REPEAT structure with INPUT temp inside the loop and UNTIL temp = -999 **M1**

(c) an IF guard added so the sentinel value itself is not compared/assigned to maxTemp, e.g. IF temp <> -999 THEN ... ENDIF around the comparison **M1**

(c) correct explanation that REPEAT checks its condition after the body has run at least once (post-condition loop), unlike WHILE (pre-condition loop), so extra logic is needed to avoid processing the sentinel **A1**

(c) Answer:

```
REPEAT
    INPUT temp
    IF temp <> -999 THEN
        IF temp > maxTemp THEN
            maxTemp <- temp
        ENDIF
    ENDIF
UNTIL temp = -999
OUTPUT maxTemp
```

Adjustment: REPEAT always executes at least once, so an IF guard is needed to stop the sentinel value being treated as a real temperature.

(d) sentinel value (accept rogue value) **B1**

(d) Answer: Sentinel value

© Revision Library | revisionlibrary.co.uk

Question 3

Level 1 Basic, largely isolated statements are made about search or sort algorithms (e.g. naming time complexities) with little or no reference to the specific context of a large, frequently-updated contact list. Limited or no use of technical terminology. **1-3**

Level 2 Relevant points are made linking algorithm/data structure choice to the context, but the discussion may be one-sided, contain minor inaccuracies, or not fully connect the trade-off between maintaining sorted data and the cost of frequent updates. Reasonable use of technical terminology. **4-6**

Level 3 A coherent, well-structured discussion that clearly links time complexity (e.g. O(log n), O(n), O(n^2), O(n log n)) to the specific demands of a large, frequently-updated contact list, evaluates the trade-off between search efficiency and update cost, and reaches a reasoned, justified conclusion or recommendation. Accurate and consistent use of technical terminology throughout. **7-8**

Indicative content:

- Binary search, O(log n), is far faster than linear search, O(n), for large data sets, but binary search requires the data to be sorted.
- Keeping a large contact list permanently sorted has a cost: inserting or deleting a contact into a sorted array may require shifting many elements, or the list must be re-sorted, adding overhead on every update.
- If updates (adds/deletes) are frequent relative to searches, the cost of continually maintaining sorted order may outweigh the benefit of faster binary searches; linear search with simple appends could be more efficient overall.
- Bubble sort, O(n^2), becomes very slow as the contact list grows large, making it a poor choice for repeatedly re-sorting a large list after updates.
- Merge sort, O(n log n), is much more efficient than bubble sort for large data sets, but it uses additional memory (not in-place) and is more complex to implement; this overhead may not be justified if the contact list is small or updates are infrequent.
- The company should consider the actual size of the contact list and how frequently searches versus updates occur before choosing; for small lists the difference between algorithms may be negligible in practice.
- An alternative data structure, such as a hash table (average O(1) lookup) or a self-balancing tree, could avoid the sorted-order maintenance problem altogether and may be more appropriate for frequently-updated data.
- A reasoned conclusion should weigh search speed against update/maintenance cost for this specific use case, rather than assuming the theoretically fastest search algorithm is always the best overall choice.

© Revision Library | revisionlibrary.co.uk