

Revision Sheet

A-Level Computer Science · Unit 10 Data Types and Structures

10 vocabulary items · 32 marks of exam questions (4 questions)

Part 1 · Vocabulary review answer key. Section A wants the exact English spelling.

Question	Answer	Marks
1	array	1
2	file	1
3	text file	1
4	end of file	1
5	Abstract Data Type	1
6	queue	1
7	linked list	1
8	circular array	1
9	free list	1
10	record structure	1

11(a)	Linked list
11(b)	Start pointer
11(c)	<p>One mark for each:</p> <p>Name: Null pointer</p> <p>Meaning: There are no further nodes in the list</p>
11(d)	<p>One mark for:</p> <ul style="list-style-type: none"> Start Pointer pointing to 'Cat' node Remaining arrows: Cat → Dolphin → Elk → Fish

12(a)	<p>One mark per point:</p> <ol style="list-style-type: none"> Open the file (in read mode and subsequently close) Initialise an index variable (to 1 // 0) Repeat (the next three steps) until the end of file is reached Read a line from the file (into a string variable) Store the line / variable in the array at the index Increment the index in a loop <p>Note: max 5 marks</p>
12(b)	<p>One mark per point:</p> <p>Construct: a <u>conditional</u> loop</p> <p>Use: To keep repeating until the end of the file is reached</p> <p>ALTERNATIVE:</p> <p>Construct: a <u>selection</u> statement</p> <p>Use: To test / check the value returned by the <code>EOF()</code> function</p>

13(a)	<p>One mark per emboldened part:</p> <ol style="list-style-type: none"> 1 Open the file (Result.txt) in read mode 2 Loop until EOF(Result.txt) // EOF // end of file 3 Read a / the / next line (from the file) and store in <code>ThisLine</code> 4 Assign <code>LineY</code> to <code>LineX</code> 5 Assign <code>LineZ</code> to <code>LineY</code> 6 Assign <code>ThisLine</code> to <code>LineZ</code> 7 After the loop, close the file (Result.txt) 8 Output <code>LineX</code>, <code>LineY</code>, <code>LineZ</code>
13(b)	<p>Example answer:</p> <p>So that the last three lines of the file are output in the correct order</p> <p>A mark for mentioning one of:</p> <ul style="list-style-type: none"> • (Ensuring) the lines are output in correct order • Ordering the last <u>three</u> lines • Ordering the lines so they are in the same order as (they occur) <u>in the file</u> <p>Max 1 marks</p>
13(c)	<p>Two loop solution</p> <p>One mark per point:</p> <ol style="list-style-type: none"> 1 Loop until given line number (-1) / <code><parameter> (-1)</code> (lines have been read) 2 ... if end of file is reached then return FALSE 3 Loop for three lines // Read three lines // Repeat of 4 and 5 for three lines 4 Read a line and output it 5 ... if end of file is reached then return FALSE 6 After outputting the (required) lines return TRUE 7 Ordering of lines no longer needed <p>Max 4 marks</p> <p>Alternative solution</p> <p>One loop solution mark scheme that reads to given line number + 2</p> <p>One mark per point</p> <ol style="list-style-type: none"> 1 Loop until given line number + 2 / <code><parameter> + 2</code> (lines have been read) 2 OR end of file is reached 3 Return FALSE if EOF reached 4 Read a line from the file in the loop 5 Continue to order the last three lines read in the loop // Steps 4 to 6 stay the same 6 (If FALSE has not been returned) output the required three lines (in the correct order) 7 ... and Return TRUE <p>Max 4 marks</p>

14(a)(i)	<p>One mark per bullet point:</p> <ul style="list-style-type: none"> • Data from the arrays is written to the <u>files</u> at the end of the day / before the program is terminated / computer is switched off • Data can then be read from the <u>files</u> at the start of the next day and written to / stored in the <u>arrays</u> • No need to (re-)enter the data manually // only need to enter data once <p>Note: Max 2 marks</p>
14(a)(ii)	<ul style="list-style-type: none"> • The data is retained when the program is terminated / after the computer is switched off // data is stored permanently // non-volatile storage
14(a)(iii)	<p>One mark per bullet point:</p> <ul style="list-style-type: none"> • Data items are combined to form a single string / saved as a single line in the file • Data items are separated by a special character // make each data item a fixed length <p>ALTERNATIVE:</p> <ul style="list-style-type: none"> • Convert all data items / 'number of people' to strings • Consecutive / each line stores a separate data item
14(b)	<pre>PROCEDURE Preview (ThisFile : STRING) DECLARE LineNum : INTEGER DECLARE ThisLine : STRING OPENFILE ThisFile FOR READ IF EOF(ThisFile) THEN OUTPUT "Warning Message" ELSE LineNum ← 1 WHILE LineNum < 6 AND NOT EOF(ThisFile) READFILE Thisfile, ThisLine OUTPUT ThisLine LineNum ← LineNum + 1 ENDWHILE ENDIF CLOSEFILE ThisFile ENDPROCEDURE</pre> <p>Marks as follows:</p> <ol style="list-style-type: none"> 1 Procedure heading (including parameter) and ending 2 File OPEN and subsequently CLOSE 3 Check if file is empty and output a warning message if it is 4 Conditional Loop 5 Output line (including blank lines) and read next line in a loop