1.4 Data Types, Data Structures and Boolean Algebra: 30-minute Pack

OCR H446 · Calculators not allowed · about 30 minutes

Total Marks

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 This question is about character encoding.

(a) State the name of the character encoding standard that can represent characters and symbols from all the world's writing systems, in contrast to standard ASCII, which only covers a limited character set. **(1)**

(b) Given that the ASCII code for the character 'A' is 65, state the ASCII code for the character 'H'. **(1)**

(c) A company's customer database must store names containing accented and non-Latin characters (for example, an umlaut in Muller, or a Chinese name). Explain one advantage of storing this data using Unicode rather than ASCII. **(2)**

(d) A new messaging app must support 200 different characters using a fixed-length binary code. Calculate the minimum number of bits required per character, showing your working. **(2)**

(Total for Question 1 is 6 marks)

© Revision Library | revisionlibrary.co.uk

2 This question is about two's complement representation and binary addition.

(a) Represent the denary number -45 as an 8-bit two's complement binary number. Show your working. **(2)**

(b) Represent the denary number 77 as an 8-bit two's complement binary number. **(1)**

(c) Add your answers to parts (a) and (b) together in binary, showing your working, and state the resulting bit pattern. **(3)**

(d) A programmer adds the 8-bit two's complement numbers 01111111 and 00000001. State the resulting bit pattern, and explain why this result demonstrates arithmetic overflow. **(2)**

(Total for Question 2 is 8 marks)

© Revision Library | revisionlibrary.co.uk

3 A program uses the following pseudocode:

TYPE StudentRecord

```
    DECLARE studentID : INTEGER
    DECLARE name : STRING
    DECLARE grade : CHAR
```

ENDTYPE

```
DECLARE student1 : StudentRecord
student1.studentID <- 10234
student1.name <- "Amara Okafor"
student1.grade <- 'B'
```

(a) State the term used to describe a data structure, such as StudentRecord above, that groups together related data items, which may be of different data types, under a single identifier. **(1)**

(b) State the value stored in student1.name after the code above has executed. **(1)**

(c) Write pseudocode to (i) declare an array called cohort of 30 StudentRecord elements, and (ii) set the grade field of the 5th student in the array to 'A'. **(3)**

(d) Explain one advantage of using a single array of StudentRecord elements, rather than three separate arrays (one for IDs, one for names and one for grades), to store the same cohort data. **(2)**

(Total for Question 3 is 7 marks)

© Revision Library | revisionlibrary.co.uk

4 This question is about floating point representation.

(a) A computer represents floating point numbers using an 8-bit mantissa and a 4-bit exponent, both stored in two's complement, with the mantissa normalised so that its first two bits differ. Represent the denary integer 22 in this floating point form, showing your working for both the mantissa and the exponent. **(3)**

(b) State what is meant by a normalised mantissa, and explain why floating point numbers are stored in normalised form. **(2)**

(c) Many decimal fractions, such as 0.1, cannot be represented exactly in binary floating point. Explain why this happens, and the effect it can have on a program that repeatedly adds 0.1 to a running total. **(2)**

(d) State the effect on the floating point number system of allocating (i) more bits to the exponent and (ii) more bits to the mantissa, keeping the total number of bits used unchanged. **(2)**

(Total for Question 4 is 9 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 1.4 Data Types, Data Structures and Boolean Algebra: 30-minute Pack

Question 1

(a) Unicode **B1**

(a) Answer: Unicode

(b) 72 cao **B1**

(b) Answer: 72

(c) Unicode includes code points for accented characters and non-Latin scripts, whereas standard 7-bit ASCII only encodes basic Latin letters, digits and symbols **B1**

(c) so names can be stored and displayed accurately rather than being corrupted or losing information **B1**

(c) Answer: Unicode covers accented and non-Latin characters that ASCII cannot represent, so names are stored and displayed correctly.

(d) 2^7=128 (insufficient, since 128<200) and 2^8=256 (sufficient, since 256≥200) tested **M1**

(d) 8 bits cao **A1**

(d) Answer: 8 bits

Question 2

(a) 45 correctly represented in binary (00101101) and correctly inverted (11010010) **M1**

(a) 11010011 cao (invert then add 1) **A1**

(a) Answer: 11010011

(b) 01001101 cao **B1**

(b) Answer: 01001101

(c) correct bit-by-bit binary addition attempted with carries shown **M1**

(c) final carry out of the 8th bit correctly discarded **M1**

(c) 00100000 cao (= 32, since -45 + 77 = 32) **A1**

(c) Answer: 00100000 (= 32)

(d) 10000000 stated **B1**

(d) explanation that both operands are positive (sign bit 0) but the result has sign bit 1 (negative), which is impossible when adding two positive numbers, so overflow has occurred **B1**

(d) Answer: 10000000 (overflow, since 127+1=128 cannot be represented as a positive 8-bit two's complement number)

Question 3

(a) a record **B1**

(a) Answer: A record.

(b) "Amara Okafor" cao **B1**

(b) Answer: Amara Okafor

(c) array declared with correct bounds for 30 elements of type StudentRecord, e.g. DECLARE cohort : ARRAY[0:29] OF StudentRecord **M1**

(c) correct index used for the 5th element (index 4, since arrays are zero-indexed) **M1**

(c) correct dot notation used to access and assign the field, e.g. cohort[4].grade <- 'A' (any correct logic, ignore minor syntax) **A1**

(c) Answer:

```
DECLARE cohort : ARRAY[0:29] OF StudentRecord
cohort[4].grade <- 'A'
```

(d) all the data relating to one student is kept together under one index, so there is no risk of the arrays becoming out of step (e.g. after sorting or deleting an entry) **B1**

(d) makes the code easier to read and maintain, since related fields are accessed via one structure (e.g. cohort[i].grade) rather than needing to keep several array indices synchronised **B1**

(d) Answer: Keeps each student's data together under one index, avoiding the arrays becoming mismatched, and makes the code clearer to maintain.

© Revision Library | revisionlibrary.co.uk

Question 4

(a) 22 written in binary (10110) and expressed as a normalised fraction, e.g. 0.10110×2^5 **M1**

(a) exponent 5 correctly converted to 4-bit two's complement (0101) **M1**

(a) mantissa 01011000 and exponent 0101 both correct (oe equivalent padding) **A1**

(a) Answer: mantissa = 01011000, exponent = 0101

(b) normalised means the mantissa's first two bits differ (no redundant leading 0s or 1s before the significant digits) **B1**

(b) normalisation gives each number a single, unique representation and maximises the number of significant bits available, maximising precision **B1**

(b) Answer: Normalised means the first two mantissa bits differ; this is used to maximise precision and give a unique representation for each value.

(c) 0.1 is a recurring (non-terminating) binary fraction, so it must be truncated/rounded to fit the fixed number of mantissa bits **B1**

(c) the small rounding error accumulates over repeated additions, causing the running total to drift from the mathematically expected value **B1**

(c) Answer: 0.1 cannot be represented exactly in binary so it is rounded; repeated addition accumulates this rounding error, causing the total to drift from the true value.

(d) (i) more exponent bits increases the range of values that can be represented **B1**

(d) (ii) more mantissa bits increases the precision (accuracy) of the values represented **B1**

(d) Answer: (i) increases range; (ii) increases precision.

© Revision Library | revisionlibrary.co.uk