MARK SCHEME for the October/November 2013 series

9691 COMPUTING

9691/33   Paper 3 (Written Paper), maximum raw mark 90

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the October/November 2013 series for most IGCSE, GCE Advanced Level and Advanced Subsidiary Level components and some Ordinary Level components.

1 **(a) (i)** `p q + 2 /`   [1]

    (ii) `6 3 5 p * + /`
                 1   [1]
    2nd mark for completely correct   [1]

  (b) Evidence for 6 and 2   [1]
    Final answer 3   [1]

  (c) (i) In-order traversal // (Traverse all subtrees in the order) left-root-right   [1]

    (ii) `A 3.14 r 2 ^ * =`   [1]

    (iii) Post-order traversal // (Traverse all subtrees in the order) left-right-root   [1]

[Total: 8]

2 **(a)** Security is better managed   [1]
Program-data independence // Changing a field does not require an applications program   [1]
Better managed data integrity/data validation // Validation code does not need to be present in all applications programs   [1]
Queries and reports quickly produced   [1]
If implemented with a DBMS it will allow concurrent access to the database   [1]
Reduced data duplication // data inconsistencies   [1]
MAX 3

  (b) (i) many cars are hired to many customers // many-to-many // M:m   [1]

    (ii) one depot has based there many cars // one-to-many // 1:M   [1]

  (c)

    Link table shown...   [1]
    2 X one-to-many relationships   [1]

  (d) (i) (Yes) since there is a not a repeated group of attributes   [1]

    (ii) (Yes) Since there is only a single attribute primary key //
        There are no partial dependencies //
          All no-key attributes are dependant on the primary key   [1]

    (iii) Two of the non-key attributes are dependant //
       `DepotManager` and `DepotAddress` are dependant on `DepotID`   [1]

    (iv) CAR(<u>CarRegistrationNo</u>, CarMake, CarModel, HirePriceCode, DepotID)

                                                                [1]

          DEPOT(<u>DepotID</u>, DepotAddress, DepotManager)   [1]

          If the primary key is no indicated, penalise once only

(e) avoids data duplication   [1]
      avoids data inconsistencies   [1]

(f) SELECT HireID, CustomerID   [1]
    FROM HIRE   [1]
    WHERE CustomerID = 'C674' AND CarRegistration = '456431'   [1]

                                                     [Total: 19]

3 **(a)** Temporary storage location   [1]
       Inside the (micro)processor   [1]

   (b) **(i)** 127   [1]

       (ii) 123   [1]

      (iii) less digits used to represent any number   [1]
          Less likely to make a mistake when copying/converting a digit string   [1]
          Easy conversion between binary and hex (vice versa) than binary and denary   [1]
                                                            MAX 1

   (c) **(i)** 2 bytes   [1]

       (ii) MAR ← [PC]   // MAR given the contents of the PC   [1]
             PC ← [PC] + 1 // PC is incremented   [1]
             MDR ← [[MAR]] // The contents of the address in MAR is copied to MDR [1]
             CIR ← [MDR]   // The contents of MDR are copied to CIR   [1]

             OR, if the candidate uses the suggested instruction ….
             MAR is given value 40 // PC contents of 40 are copied to MAR   [1]
             7324/The contents of address 40 is copied to the MDR   [1]
             PC is incremented from 40 to 41   [1]
             7324/contents of location 40 is copied to CIR   [1]
                                                            MAX 5

(d)

ACC	Memory address	
	153	160
13		0
13		
	13	
150		
151		
		151
23		
36		
	36	
151		
152		
		152

[4]

[Total: 15]

4 (a) Rules are: 15 and 25 [1]

(b) (i) `Who = zhen`
`Who = kong` [1]

(ii) `false` [1]

(iii) `false` [1]

(c) (i) `has_licence(X) AND passed_theory_Test(X) AND`
`                passed_driving_test(X, motorbike)`
each clause scores 1 [3]
use of two AND operators [1]
MAX 3

(ii)
```
9 ?- passed_theory_test(Who), not(passed_driving_test(Who, car)),
 not(passed_driving_test(Who, motorbike)).
Who = yin ;
```

OR *(using the anonymous variable) ...*
```
10 ?- passed_theory_test(Who), not(passed_driving_test(Who, _)).
Who = yin ;
```
[3]

(d) `has_licence(ho)` returns TRUE // clause 11 [1]
`age(ho, A)` returns 15 // A=15 [1]
`minimum_age(motorbike, L)` returns L=15 // clause 2 [1]
`A >= L` returns FALSE [1]
`able_to_drive(ho, motorbike)` returns false [1]
MAX 3

[Total: 12]

5 **(a)** BOOLEAN   [1]
Flags when the book title is found   [1]
STRING (for SearchBook)   [1]

```
OPENFILE Book.txt for Output
INPUT SearchBook                                                                                  [1]
IsFound ← FALSE

REPEAT
    READ next book data value and assign to NextBook
    IF  NextBook = SearchBook                                                                     [1]
        THEN
            IsFound ← TRUE
            OUTPUT "FOUND"
    ENDIF
UNTIL (IsFound = TRUE) OR EOF                                                                      [1]

IF IsFound = FALSE // NOT IsFound                                                                  [1]
    THEN
        OUTPUT "Book title was NOT FOUND"
ENDIF
CLOSEFILE                                                                                          [1]
```

(b) The search will read on average 125 records   [1]

(c) **(i)** The data items must be in order   [1]

   (ii) The function makes a call to itself (in two places)   [1]

   (iii)
```
BinarySearch(BookTitle, "Tortoise Care", 1, 11)
     High < Low is FALSE
     Middle = 6
     BookTitle[6] > "Tortoise Care" is FALSE
     BookTitle[6] < "Tortoise Care" is TRUE
     BinarySearch(BookTitle, "Tortoise Care"  7, 11)                                              [1]
```
```
     High < Low is FALSE
     Middle = 9                                                                                   [1]
     Booktitle[9] > "Tortoise Care" is FALSE
     Booktitle[9] < "Tortoise Care" is TRUE
     BinarySearch(BookTitle, "Tortoise Care" 10, 11)                                              [1]

         High < Low is FALSE
         Middle = 10
         BookTitle[10] > "Tortoise Care" is FALSE                                                 [1]
         Booktitle[10] < "Tortoise Care" is FALSE
         RETURN 10
     ENDFUNCTION

     ENDFUNCTION                                                                                  [1]

ENDFUNCTION
```
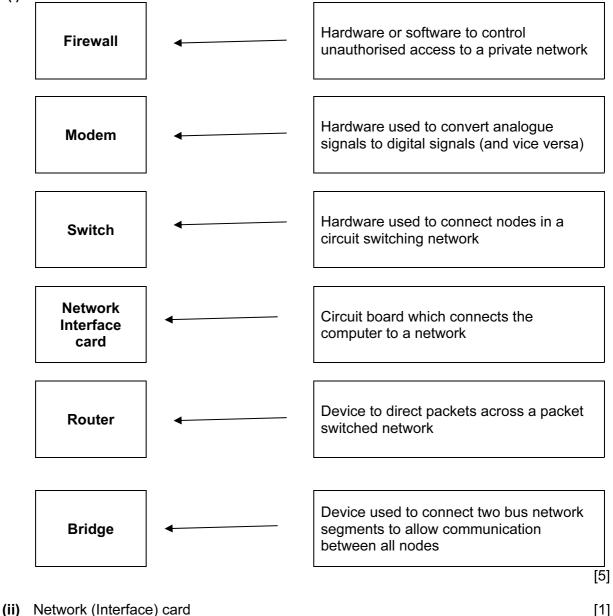
[Total: 16]

6 (a) *Boot file ...*
Stored in the BIOS/ROM [1]
The initial sequence of instructions run when the computer is powered on [1]
Information on which drive to look for the operating system [1]
Triggers the loading of the operating system [1]
MAX 3

(b) (i) *An interrupt*
a signal from some device [1]
to indicate that some event has occurred [1]
the device is seeking the attention of the processor [1]
MAX 1

(ii) *Hardware generated ...*
reset [1]
multiprogramming 'end of time slice'
other valid answers ...

Software generated ... [1]
Division by zero error
Other valid answers ...
MAX 2

(c) RUNNING
The process currently has use of the processor [1]

READY
The process would like to use the processor when the current process releases the processor [1]

SUSPENDED
The process cannot currently use the processor// or by example, the job is currently using an I/O device [1]

[Total: 9]

7 (a) (i)

Firewall	←	Hardware or software to control unauthorised access to a private network
Modem	←	Hardware used to convert analogue signals to digital signals (and vice versa)
Switch	←	Hardware used to connect nodes in a circuit switching network
Network Interface card	←	Circuit board which connects the computer to a network
Router	←	Device to direct packets across a packet switched network
Bridge	←	Device used to connect two bus network segments to allow communication between all nodes

[5]

(ii) Network (Interface) card [1]

(b) (i) Copper wire/coaxial/twisted pair
Wire conducts electricity // changing current denotes different signals
Optic fibre cabling
Separate fibres used for separate signal
Data travels very fast
Signal transmitted as light pulses/travels at the speed of light
Radio/Microwave signals
Wireless communication // allows for mobile communication
Mark as 2 × 2   MAX 4

(ii) Maximum possible distance [1]
Speed of communication // data transfer rate [1]
MAX 1

[Total: 11]

© Cambridge International Examinations 2013