CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Advanced Level

MARK SCHEME for the October/November 2014 series

9691 COMPUTING

9691/31 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 2014 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

CAMBRIDGE
International Examinations

1 (a) (i) a b + 6 / [1]

(ii) 3 x y * 3 + * [2]
1

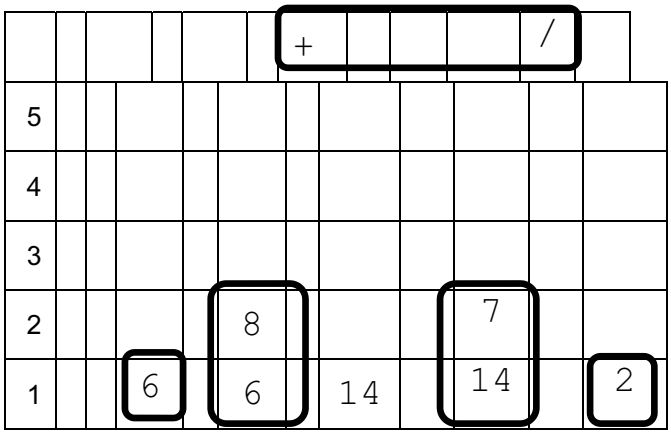
(b) (i) 3 * (x + y + z) [1]

(ii) $(7^y + 6) / 2$

1 mark only for:
- 7^y or…
- (7^y = 6) / 2 [2]

(c) (i) Last item added is the first to leave // first add will be the last to leave
Last in – First out // First in – Last out
Refuse: LIFO [1]

(ii)



[max 4]

2 (a) The main memory is divided into page frames
The program is divided into pages
Only some of the pages of the program are loaded to start execution of the program
The operating system must manage the allocation of pages to page frames
The Page (Map) table shows the mapping of pages to page frames [max 3]

(b) 'Priority' which is well explained and clear × 2
E.g. Anticipated shortest time to complete
Shortest remaining time to complete

[max 2]

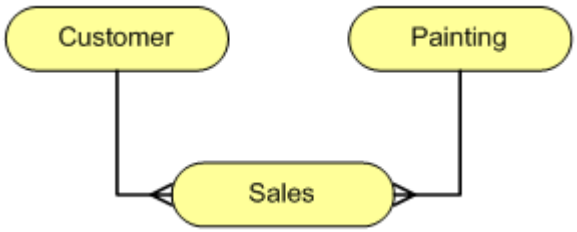
(c) *Mark as follows…*

B – D – F – A Scores full 4
or…
C and E are excluded 1

B 1
D 1
F 1
A each in the correct position 1 [max 4]

3 (a) (i) `Sales(SalesID, CustomerID, PaintingID, PurchaseDate)` [2]

(ii)



2 X correct relationship [2]

(iii) A customer can never purchase more than one painting on the same date [1]

(b) (i) `Not in 2NF…` – Sales 1

`CustomerName` is known from only `CustomerID` //
`CustomerName` will be known by only knowing part of the primary key 1

`Sales(CustomerID, PurchaseDate, PaintingID)` 1 [3]

(ii) *`Not in 3NF…`* – `Painting` 1

There are non-key attributes which are dependent.
Or by example…
`DateBorn/DateDied/Nationality` are all dependant on `ArtistName` 1

```
Painting(PaintingID, Description, PaintingDate,
ArtistName, Price)

Artist(ArtistName, ArtistDateBorn, ArtistDateDied,
ArtistNationality)
```

Mark as follows:
All except `ArtistName` removed from table `Painting` 1
New table `Artist` 1
`Artist` containts at least three of the correct attributes 1 [5]

(c)
```
UPDATE Customer                                    1
SET TelNo = "0123 456789"                          1
```

```
WHERE CustomerID = "065"
```
1 [3]

4 (a) (i) ACC = 77 1
Show contents of 203 copied to ACC 1 [2]

(ii) ACC = 65 1
Show 150 used as a forwarding address 1
Contents of 200 copied to ACC 1 [3]

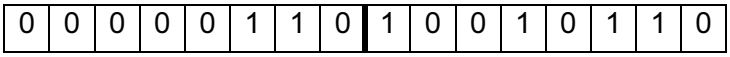
(b) (i) 256 different instructions [1]

(ii) Store the ACC contents 1
at address 65 // 01000001 1 [2]

(iii) Fewer digits to write // less chance of an error in writing the code // easy conversion to/from binary code [1]

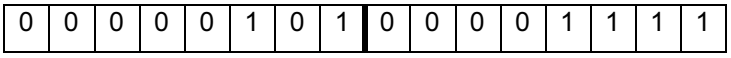
(iv) 1041 hex [1]

(v) `LDI 150`



Opcode 1
Operand 1 [2]

(vi) `LDV 15`



Opcode 1
Operand 1 [2]

(vii) True
`OUTCH / IN // END` or using a good explanation (only) of either [2]

(c)

ACC	Location 150	OUTPUT
65		A
200		
201	201	
76		L
201		
202	202	
65		A
202		
203	203	
77		M
203		
204	204	

Mark as shown [5]

5 **(a)** a single processor
program consists of a sequence of stored instructions 1
instructions + data make up a 'program' 1
are stored in a continuous block of main memory 1
instructions are executed in sequence 1 [max 2]

(b) 1. The (contents of) the program counter/PC are copied to the Memory Address Register
2. The contents of the Program Counter are incremented
3. Identify the address in the Memory Address Register. Go to this address and copy its contents to the Memory Data Register
4. The (contents of) the Memory Data Register are copied to the Current Instruction Register [4]

(c) **(i)** Control bus [1]

(ii) read/write
interrupt
reset
clock signal
bus request/bus grant [max 1]

(d) (i) Case 1 1
The operand number is already held in the CIR 1 [2]

(ii) Case 2 1
The instruction is for directed addressing
The address bus is loaded with address 35 1 [2]

6 (a) (i) All the keyboards which make up the syntax of the language 1
A token for each keyword 1 [2]

(ii) DECLARE, CONSTANT, CALL, REPEAT (any three) [1]

(iii) A list of all the identifiers used by the program 1
A pointer to where their value is stored in memory 1 [2]

(iv) `i, Customer, Address, DiscountRate, InitialiseCustomerData`
(any three…) [1]

(v) *Lexical analysis*
remove any whitespace from the source file 1
remove any comment statements 1
check for obvious errors in the use of identifiers (names) e.g. they do not exceed 64 characters 1
replace all language keywords with their token (by searching for the appropriate keyword in the keyword table) 1
place an identifier names in the symbol table 1
search for the appropriate identifier in the symbol table – the identifier name is replaced in the source code by a pointer value 1 [5]

(b) (i) *Code optimisation*
the process of taking the final executable code produced by the compiler and changing it in some way 1

in order that it will use fewer resources // less memory 1
Refuse: reduced in size

it will execute faster 1
removes redundant code 1 [max 2]

(ii) 203 [1]

7 **(a)** 3 (days) [1]

(b) Error [1]

(c) 2 (months) [1]

(d) Error [1]

(e) Error [1]

(f) Built-in functions are those provided (as a part of the programming language) // accept by example 1
User defined functions are designed and coded by the programmer 1 [2]