CAMBRIDGE INTERNATIONAL EXAMINATIONS

Cambridge International Advanced Level

MARK SCHEME for the October/November 2015 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 2015 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

CAMBRIDGE
International Examinations

1 (a) The table has a repeated group of attributes //
There are several orders for the same customer/`CustomerID` [1]

(b) (i) `CUSTOMER(CustomerID, ……..)`
`PRODUCT(ProductID)` [1]

`ORDER(CustomerID, OrderDate,……..)` [1]

(ii)

[2]

1 mark for each correct one-to-many relationship

(iii) Primary key //`CustomerID` in the `Customer` table [1]
Links to foreign key (`CustomerID`) in the `ORDER` table [1]

(c)
```
SELECT ProductID                                          [1]
FROM PRODUCT                                              [1]
WHERE RetailPrice>=100 AND RetailPrice<=200               [1]
```

(d)

Creates a new record in the `ORDER` table	
Amends an existing record in the `ORDER` table	✓
Assigns the `Dispatched` attribute a `TRUE` value	✓
Creates a new attribute `Dispatched`	
Changes all the existing records for customer 647	
Changes one record for customer 647	✓

[1] [1] [1]

Remove 1 mark for each additional tick.

(e) (i)
```
INSERT INTO ORDER                                         1
(CustomerID, OrderDate, ProductID, Dispatched,
DispatchDate)                                             1
VALUES (447, #17-10-15#, 982, FALSE, (NULL))              1   [3]
```

(ii) Attempt to add a record in `ORDER` table 1
But, no corresponding `ProductID` in the `PRODUCT` table 1
Or:
// Delete a record in the `PRODUCT` table 1
and, matching records in `ORDER` table remain 1 [2]
Or: Similar explanation with `ORDER` and `CUSTOMER` and the `CustomerID` attribute // Allow use of the term 'update' if mentions a change to `TutorID`/foreign key attribute

2 (a) (i) Building a model of the system // Models the behaviour of the system 1
The model records over time the result of changing parameters/conditions/ circumstances // predicts outcomes for the real-world scenario 1 [2]

(ii) A computer program can be written to build the model 1
The computer system can process results very quickly // can change the time frame // Can process large volumes of data 1
Use of the computer avoids possible health and safety issues 1 [max 1]

(b) Temperature sensor 1
Air pressure sensor 1
Moisture sensor 1 [max 2]

(c) Wind tunnel requires that an actual physical model is built 1
The modelling of the weather is only an abstraction realised by the computer software 1 [2]

3 (a) (i) 0101 1000 1
0111 1101 1 [2]

(ii) 16 [1]

(b) (i) Action
Description
`MAR ← [PC]`
The contents of the Program Counter are copied to the Memory Address register [1]

`PC ← [PC] + 1`
The contents of the Program Counter are incremented [1]

`MDR ←[[MAR]]`
The contents of the address currently in the Memory Address Register are copied to the Memory Data Register [1]

`CIR ← [MDR]`
The contents of the Memory Data Register are copied to the Current Instruction Register [1]

(ii)

Fetch stage	Special purpose registers (Contents shown in hex)				Buses	
	PC	MAR	MDR	CIR	Address bus	Data bus
	58					
MAR ← [PC]		58			✓	
PC ← [PC] + 1	59					
MDR ←[[MAR]]			867A			✓
CIR ← [MDR]				867A		

[max 5]

4 (a)

	Register	
Instruction	**ACC**	**Index Register (IX)**
LIX 400		3
LDD 401	616	
LDI 401	96	
LDX 401	63	

[1]

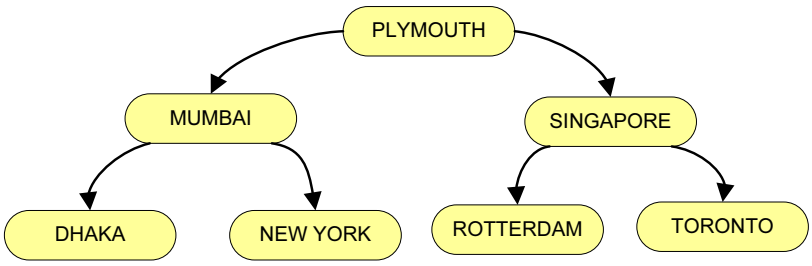
[1]

[1]

[1]

(b) **Use the text editor to write the assembly language program**
PROG.ASM 1
REPEAT
PROG.ASM is input to the assembler software 1
IF errors reported
THEN
Amend PROG.ASM using the text editor 1
ENDIF
UNTIL No errors reported
Produce the PROG.EXE executable file 1
Run PROG.EXE 1 [max 4]

5 **(a)** **(i)**



Root correct 1
Left subtree correct 1
Right subtree correct 1 [3]

(ii) *Labelling ....*
Root [1]
Left subtree // FT for their tree [1]

(iii) 4 // FT for their tree [1]

(b) **(i)**
```
INTEGER
ARRAY[1 : 2000] OF STRING
```
[2]

(ii)

RootPtr	1					
1	3			LIMA		2
2	4			PARIS		5
3	6			KARACHI		0
4	0			MELBOURNE		0
5	0			WARSAW		0
6	0			CAPE TOWN		7
7	0			EDINBURGH		0

[4]

(c)
```
//binary tree search
INPUT SearchCity                                                    [1]
IsFound ← FALSE
Current ← RootPtr
REPEAT
   IF City[Current] = SearchCity                                    [1]
      THEN
         //found
         OUTPUT "Found"
         IsFound ← TRUE                                             [1]
      ELSE
         IF SearchCity > City[Current]
            THEN
               // move right
               Current ← RightPtr[Current]                          [1]
            ELSE
               Current ← LeftPtr[Current]
         ENDIF
      ENDIF
UNTIL Current = 0 OR IsFound = TRUE                                 [1]

IF IsFound = FALSE                                                  [1]
   THEN
      OUTPUT SearchCity "Not Found"
ENDIF
```

6 (a) (i) `SumRange` 1
`ThisInteger1, ThisInteger2, Flag` 1
Must be identifiers only … [2]

(ii) 6 [1]

(iii) ERROR [1]

(iv) ERROR [1]

(v) 11 [1]

(vi) ERROR [1]

7 **(a)** More than one program loaded into memory at the same time [1]

(b) Anything reasonable …
printer drivers
spooler
linker
loader
compiler / assembler
backup software

R. "System software" and "Utilities" [max 2]

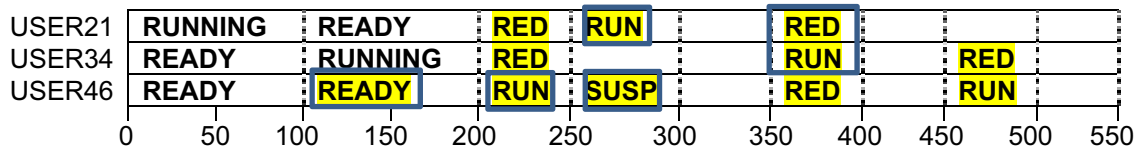
(c)

All the (data) is processed together/at the same time	1	
There is a time delay before processing	1	
Output is generated as a batch	1	
Processing cannot start until all data has been collected/input	1	
There is no user involvement // the process runs until completion	1	[max 3]

(d) **(i)**

Each program can use the processor in turn	1	
For a time of 100 milliseconds // for the fixed time slice	1	[2]

(ii)



1 mark each [5]

(iii) Input/output request [1]

8 **(a)** *The diagram includes the following*
One or more communication links to ….

A modem // router	1	
Firewall	1	
Laser printer	1	
File server // database server	1	[max 4]

Penalise once only the omission of a comms. link line

(b) Twisted pair 1
Description 1
Or …
Coaxial cable 1
Description 1
Or …
Optical fibre 1
Description 1

Allow descriptors CAT 5, CAT 6 [max 2]