MARK SCHEME for the October/November 2013 series

9691 COMPUTING

9691/22　　　Paper 2 (Written Paper), maximum raw mark 75

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)
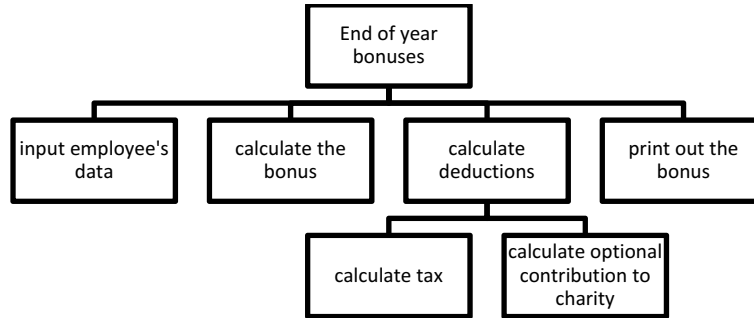
1 mark for top level
1 mark for 4 boxes at 2ⁿᵈ level in order incl. a verb
1 mark for 2 boxes below 'calculate deductions' [3]

(b) Any three points + **good** explanation [6]

– use of indentation …
– … to show structure

– comments/annotation …
– … to know what each line/block is doing // to help understanding of this line/section/block

– meaningful variable names …
– … to convey the purpose of the variable

– capital keywords/reserved words // capitalised identifiers
– … to make these words stand out

(c) (i) – local variables
– only have scope <u>in the module</u> they are <u>declared</u> in [2]

(ii) – parameters
– pass/return values/parameters into a module [2]

(d) Value + reason (<u>different</u> reasons within each type) × 5 [5]

2 (a) *mark as follows:*

    1 mark for declaration
    1 mark for size (at least 150 × 2)
    1 mark for type specification [3]

(b) (i) – to ensure that known values are stored
    // no rogue values found later
    // to identify unused locations [1]

(ii) *mark the structure as follows:*

    1 mark for 2 FOR <u>loops</u> or equivalent
    1 mark for working through <u>all</u> the array
    1 mark for initialising an array element with a sensible value (e.g. empty string, NULL)
    (must give 2 indices) [3]

(c) Any three points from:

    – declare a record type // description of a record
    – declare a file type
    – create a new file of this file type
    – repeatedly assign data from array to record
    – save the record to file
    – close file
    – open and read file [3]

2 (d)

Identifier	Description	Data type	Field size (Bytes)
EbookID	a unique ebook ID, a whole number between 1 and 500	INTEGER	2–9
BookTitle	the title of the ebook	STRING	20–50
Author	the author of the ebook	STRING	10–30
DateBought	date bought	DATE/DateTime /STRING /INTEGER	6–12
Cost	price paid	CURRENCY /REAL/FLOAT	5–8
Fiction	fiction or non-fiction	BOOLEAN	1

[8]

3 **(a)** Any five from:

– prompt
– space for password entry
– indication of number of attempts
– space for successful/unsuccessful message
– button/similar to cancel process
– uses space available   [5]

3 **(b) (i)**

Attempt	Password	Password = "Aisha"	Attempt > 3	Password = "Aisha" OR Attempt > 3	Output
0					
	Aisha				
1					
		True`			
			False		
				True	
					Password correct

  1 mark   **1 mark**   **1 mark**

[3]

3   (b) (ii)

Attempt	Password	Password = "Aisha"	Attempt > 3	Password = "Aisha" OR Attempt > 3	Output
0					
	aisha				
1					
		False	False	False	
	Asha				
2					
		False	False	False	
	AISHA				
3					
		False	False	False	

 1 mark   1 mark   1 mark   1 mark   1 mark

[5]

3 **(c)** **(i)** Logic error [1]

(ii) – until password = "Aisha" OR attempt = 3
// until password = "Aisha" OR attempt >= 3
// until password = "Aisha" OR attempt > 2 [1]

(d) Too obvious/simple //easy to guess // easily hacked
// *comparable reason* [1]

(e) – a longer string/password
– mixture of different character types // letters and numbers
– random characters // do not use a dictionary word // don't use accessible personal data [2]

(f) **(i)** *mark as follows:*

1 mark for inputting <u>numerical</u> value
1 mark for validation of input between 1 and 4
1 mark for error message
1 mark for 3 or 4 IF statements
1 mark for correct nesting
1 mark procedure call implementation [6]

(ii) *mark as follows:*

correct opening of CASE/SELECT statement
correct label
correct action
correct end of CASE/SELECT statement [4]

4 (a) (i)

Line number	x	y	RETURN
0	8	2	
5	0	(2)	
6			Divisor(2,0)
0	2	0	
3			2

1 mark for line 5
1 mark for line 6 // second line 0 + line 3   [2]

(ii)

Line number	x	y	RETURN
0	38	7	
5	3	(7)	
6			Divisor(7,3)
0	7	3	
5	1	(3)	
6			Divisor(3,1)
0	3	1	
5	0	(1)	
6			Divisor(1,0)
0	1	0	
3			1

1 mark for 1st line 5
1 mark for candidate's 1st line 0
1 mark for 2 more line 5s and line 0s
1 mark for output   [4]

© Cambridge International Examinations 2013

4 **(b)** **(i)** (Function) <u>terminating</u> condition [1]

   (ii) The function would never end // infinite loop // stack overflow [1]

(c) – at the first recursive call // 1st time line 6 is reached
   – the call reverses the values of x and y
   – function carries on normally [3]