CAMBRIDGE INTERNATIONAL EXAMINATIONS

GCE Advanced Subsidiary Level and GCE Advanced Level

MARK SCHEME for the October/November 2013 series

9691 COMPUTING

9691/23 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.

CAMBRIDGE
International Examinations

1 (a) (i) – arithmetic on LHS of assignment statement

– `NoOfBalances = NoOfBalances + 1` [2]

(ii) – incorrect condition

– `IF NoOFPipettes > NOOfBeakers OR NoOfPipettes < NoOfBottles THEN OUTPUT "Check the numbers"` [2]

(iii) – structure is unplanned
– cannot view the problem/comparable reason [1]

(b) – easier to understand the problem
– when dealing with smaller problems
– simpler to debug
– small parts at a time
– easier to maintain
– changing small sections
– can re-use his modules
– in this and future work [max 6]

(c) (i) – it will show the relationship between different parts of the program
– will help break up modules into smaller sections [2]

(ii) – header/title box
– 3rd level [2]

(d) 3 boxes under 1st alternative module
– correctly named
– in correct order [2]

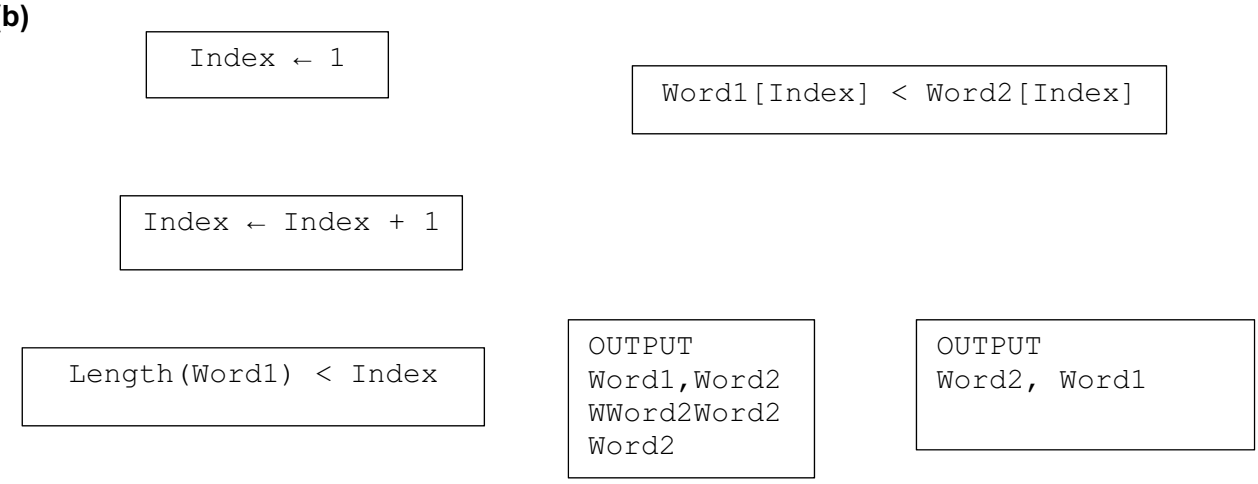
(e) (i) – equipment ID [1]

(ii) – passed as a parameter
– by reference or value
OR
– global variable
– accessible throughout the program [2]

2 (a) (i) – as a numerical/binary value
– according to some standard code/ASCII [2]

(ii) – numerical order corresponds to alphabetical order [1]

(b)



[6]

(c) e.g. C#

```
string[ ]    Equipment = new string[500] ;
for    (int  i = 0;  i < 500; i ++ )  {
   Equipment [ i ] =  " " ;
}
```

– declaring an array size 500
– of type string
– FOR loop x 500
– Initialising elements to a reasonable value, say " " [4]

(d)
– condition for those <=2000
– correctly nested
– IF dealing with values ending in zero
– other 2 conditions
– assigning place
– indentation [6]

(e) e.g. VB6

```
Dim EquipID As Integer
EquipID = InputBox ("Enter Equipment ID;  ")
Select Case EquipID
   Case 1 To 2000
      MsgBox("Physics Lab")
   Case 2001 To 4000
      MsgBox("Biology Lab")
   Case 4001 To 8000
      If EquipID Mod 10 = 0 Then
         Console.WriteLine("Chemistry Lab – locked cabinet")
      Else
         Console.WriteLine("Chemistry Lab")
      End If
   Case Else
      Console.WriteLine("Invalid Equipment ID")
End Select
```

– CASE/SELECT header
– correct CASE labels (NO =)
– dealing with sub-condition
– method of finding values ending in 0
– ending CASE/SELECT [5]

3 **(a)** – prompt for name of equipment
– drop-down menus
– extra menu if Chemistry chosen
– displays for ID and keeping place
– controls/buttons
– use of full space [6]

(b) – title
– date
– heading for ID
– grouping
– heading for keeping place
– use of all page [max 5]

(c) – indentation
– comments/annotation
– Capitalisation
– meaningful variable names [4]

(d) – correct condition within loop
– correct starting value
– indented
– sensible names
– comments
– correct logic [6]

(e) **(i)** – error in the design of the program/producing wrong results
– example [2]

(ii) – error detected during execution
– example [2]

4 **(a)** – within the function [1]

(b) – line 5 [1]

(c) – Calc(3)
– Calc(1)
– Calc(-1) [3]

(d) – 12 [1]