Cambridge IGCSE

Cambridge International Examinations
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/22**

Paper 2 **May/June 2017**

MARK SCHEME

Maximum Mark: 50

Published

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 May/June 2017 series for most Cambridge IGCSE®, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

This syllabus is approved for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **6** printed pages.

CAMBRIDGE
International Examinations

Question	Answer	Marks
1(a)(i)	**One** variable name **MUST** relate to the cost of the outing in Task 1 - Variable name (1) - Data type to match variable (1) - Description of the use of the given variable (1) Many correct answers, they must be meaningful. This is an example only. - `NoSeniorCitizens` (1), integer (1), number of senior citizens that want to go on the outing (1)	**3**
1(a)(ii)	**Two** constants required, for each constant - Name (1) - Value (1) - Use (1) Many correct answers, they must be meaningful. These are examples only. - `MinNoSeniorCitizens` (1), 10 (1), minimum number of senior citizens that can go on the outing (1) - `MaxNoSeniorCitizens` (1), 36 (1), maximum number of senior citizens that can go on the outing (1) Max 6 marks	**6**
1(b)	- calculate cost of carers // if more than 24 senior citizens on the trip cost is 60 otherwise cost is 40 - add to the cost of the outing	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(c)</td><td>

Any **five** from:

- loop for number of senior citizens on the trip
- input with prompts name and amount paid
- store name and amount paid in appropriate place in arrays
- total the amount paid
- check if spare places are available
- if spare place is required remove a spare place//fill spare places
- add name(s) to list in appropriate place(s)
- store names of two carers
- If number of senior citizens > 24 store name of third carer

Max 5 marks

Example

```
TotalPaid ← 0
FOR Counter ← 1 TO NoSenCit
  PRINT "Please Enter Name"
  INPUT SenCitName[Counter]
  PRINT "Please Enter amount paid"
  INPUT SenCitAmount[Counter]
  TotalPaid ← TotalPaid + Amount
NEXT Counter
Extras ← TRUE
WHILE NoSenCit < 36 and Extras
  PRINT "Do you want to add another person? Y/N"
  INPUT Answer
  IF Answer = "Y"
    THEN
      NoSenCit ← NoSenCit + 1
      PRINT "Please Enter Name"
      INPUT SenCitName[NoSenCit]
    ELSE Extras ← FALSE
  ENDIF
ENDWHILE
PRINT "Please Enter Name of First Carer"
INPUT Carer1
PRINT "Please Enter Name of Second Carer"
INPUT Carer2
IF NoSenCit > 24
  THEN
    PRINT "Please Enter Name of Third Carer"
    INPUT Carer3
ENDIF
```

</td><td>5</td></tr>
<tr><td>1(d)</td><td>

Explanation (any programming statements must be fully explained)

- check total cost
- ……against total amount paid
- if total cost < total amount paid <u>display/show</u> profit
- if total cost = total amount paid <u>display/show</u> break even

</td><td>4</td></tr>
</table>

Question	Answer	Marks

2(a)	award full marks for any working solution
	- Input three numbers (1)
	- Attempt to select largest number (1)
	- Working method (1)
	- print out largest number (1)
	
	Sample algorithm
	`INPUT Num1, Num2, Num3`
	`IF (Num1 > Num2) AND (Num1 > Num3) THEN PRINT Num1`
	`ENDIF`
	`IF (Num2 > Num1) AND (Num2 > Num3) THEN PRINT Num2`
	`ENDIF`
	`IF (Num3 > Num1) AND (Num3 > Num2) THEN PRINT Num3`
	`ENDIF`
	or
	
	`INPUT Num1`
	`Big ← Num1`
	`INPUT Num2, Num3`
	`IF Num2 > Big THEN Big ← Num2 ENDIF`
	`IF Num3 > Big THEN Big ← Num3 ENDIF`
	`PRINT Big`
		4

2(b)	1 mark for each data set and 1 mark for the matching reason. There are many possible correct answers, these are examples only. *Test data set 1:* 30, 29, 28 *Reason:* first number is the largest Test data set 2: x, y, z Reason: abnormal data, should be rejected Max 4 marks	**4**

Question	Answer	Marks
3		5

Weight	Reject	Total Weight	OUTPUT
	0	0	
13		13	
17		30	
26	1		
25		55	
5		60	
10		70	
15		85	
35	2		
20		105	
		85	Weight of items 85 Number of items rejected 2
(1mark)	(1 mark)	(1 mark to 1st 85) (1 mark 105, 85)	(1 mark)

Question	Answer	Marks
4(a)		2

```
Error          - Count ← 0
Correction     - Count ← 1
or
Error          - UNTIL Count > 100
Correction     - UNTIL Count >= 100 or UNTIL Count = 100
or
                 UNTIL Count > 99
```

Question	Answer	Marks
4(b)		3

- use of FOR with correct start and end values …
- … use of NEXT
- … removal of increment for Count

Sample algorithm

```
      Sum ← 0
      FOR Count ← 1 TO 100
        INPUT Number
        Sum ← Sum + Number
      NEXT // NEXT Count
   PRINT Sum
```

Question	Answer	Marks
5(a)		8

for each field name (1), data type and sample (1)

The following are examples there are many different correct answers.

- EarTag (1), text, EAR1011 (1)
- DOB (1), date, 4/3/2017 (1)
- Gender (1), text, M (1)
- Weight (1), number, 5.9 (1)

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(b)</td><td>EarTag</td><td>1</td></tr>
<tr><td>5(c)</td><td>
<table>
<tr><td>Field:</td><td>EarTag</td><td>Gender</td><td>Weight</td><td></td></tr>
<tr><td>Table:</td><td>SHEEP</td><td>SHEEP</td><td>SHEEP</td><td></td></tr>
<tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>Criteria:</td><td></td><td>='M'</td><td>> 10</td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>(1 mark)</td><td>(1 mark)</td><td>(1 mark)</td><td></td></tr>
</table>
</td><td>3</td></tr>
</table>