Cambridge
O Level

Cambridge International Examinations
Cambridge Ordinary Level

COMPUTER SCIENCE **2210/22**

Paper 2 **October/November 2016**

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 October/November 2016 series for most Cambridge IGCSE®, Cambridge International A and AS Level components and some Cambridge O Level components.

® IGCSE is the registered trademark of Cambridge International Examinations.

© UCLES 2016

Section A

1 (a) (i) Many correct answers, they must be meaningful. This is an example only.
– `Choice`, integer, choice of charity
– `Cost`, real, cost of shopping
– `Donation`, real, donation calculated from cost of shopping [3]

(ii) – Array
– a set of (similar) variables grouped together// description **or** array declaration applied to the scenario e.g. DonationTotals[1:3]
– allows for more efficient programing e.g. use of indices//each charity total can be identified by an index

or

– List
– a set of variables grouped together// description **or** list declaration applied to the scenario e.g. DonationTotals[]
– allows for more efficient programing e.g. use of a loop to update each charity

or

– Variables
– storage locations that can be changed// description **or** declaration applied to the scenario e.g. DonationTotal1, DonationTotal2 and DonationTotal3
– e.g. as there are only 3 charities so there is no need to use an array

[3]

(b) Any **five** from:
– Prompt for input of charity choice // prompt for input of value of shopping
– input charity choice
– check for input of 1, 2, or 3
– input value of shopping
– calculate donation
– add donation to the appropriate total
– output name of charity and amount/total amount donated [5]

Sample Answer.

```
REPEAT
  PRINT 'Please enter choice of charity 1, 2. or 3 '
  INPUT Choice
UNTIL Choice = 1 or Choice = 2 or Choice = 3
PRINT 'Please enter value of shopping bill '
INPUT BillValue
Donation ← BillValue * 0.01
Total (Choice) ← Total (Choice) + Donation
PRINT 'Charity ', CharityName (Choice), ' has received a donation
of ', Donation
```

© UCLES 2016

(c) Maximum **six** marks in total for question part
Description (may include reference to program statements)
– when charity choice = -1
– display total donation for **each** charity …
– … with corresponding charity name…
– description of method for selecting descending order of totals
– evidence that the method works
– calculate grand total from 3 totals / sum of all donations
– output 'GRAND TOTAL DONATED TO CHARITY' and grand total

[6]

(d) Any **three** from:
– input number of charities
– store the number of charities as a variable
– change the upper value of the choice input
– change the array bounds for total donations etc.// add new variables to hold extra values
– the need to change the code…
– … to allow for differing number of charities

[3]

© UCLES 2016

Section B

2 1 mark for identifying each error, 1 mark for the corresponding change

– line 2 `or Counter = 100`
– `Counter = 0`

– line 6 or `UNTIL Num < 0`
– `UNTIL Num >= 0`

– line 7 or `Total = Total + 1`
– `Total = Total + Num`

– line 8 or `Counter = Counter + Num`
– `Counter = Counter + 1` [8]

3 Trace table for input value 33

X	A	B	OUTPUT
33	4	1	1
4			4

← (1 mark) → (1 mark)

Trace table for input value 75

X	A	B	OUTPUT
75	9	3	3
9	1	1	1
1			1

← (1 mark) → (1 mark)

[4]

© UCLES 2016

4 For each example: **1 mark** for **correct structure, 1 mark** for **appropriate content, 1 mark** for **the reason.** There are many correct answers these are only examples

```
IF X > 0 AND X <= 10
  THEN PRINT 'In Range'
  ELSE PRINT 'Out of Range'
ENDIF
```

– e.g. checking a condition that may be complex//uses relational operators// checking for a range of values// only 2 options

```
CASE X OF
  1 : PRINT 'Option 1'
  2 : PRINT 'Option 2'
  3 : PRINT 'Option 3'
  OTHERWISE PRINT 'Incorrect choice'
ENDCASE
```

– e.g. checking for discrete/large number/more than 2 of values

[6]

5 **(a)** – 6 [1]

(b)

– Play	text
– No Seats Stalls	number
– Price Stalls Seats $	currency

[3]

(c) **1 mark** for **correct plays**, **1 mark** for **correct dates** with each play and no extra fields or text, **1 mark** for the **order**

As You Like It	01/07/2016
Julius Caesar	22/07/2016
Macbeth	14/07/2016

[3]

© UCLES 2016

(d)

Field:	Play	Performance Date	Number Seats Circle	Price Circle Seats $
Table:	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION
Sort:		Ascending/ Descending		
Show:	☑	☑	☐	☑
Criteria:			>=6	
or:				
	(1 mark)	(1 mark)	(2 marks) 1 for Criteria 1 for correct Field & Table & Sort & Show & or	(1 mark)

[5]

© UCLES 2016