Cambridge IGCSE

Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education (9–1)

COMPUTER SCIENCE **0984/21**

Paper 1 **May/June 2019**

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 International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

This document consists of **9** printed pages.

Cambridge Assessment International Education

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks
1(a)(i)	Many correct answers, the name must be meaningful. Must relate to **Task 1** 1 mark per bullet point e.g.1 • Array `BusA` • Data type integer • Purpose storing the minutes late e.g.2 • Array `Day` • Data type string • Purpose storing the code for the day of the week	**3**
1(a)(ii)	Many correct answers, the name must be meaningful. Names shown are examples only. 1 mark per bullet point • Task 2 variable name `BusAMinsLate` • Data type real • Purpose used in calculation of average minutes late • Task 3 variable name `SearchDay` • Data type string • Purpose to input the day to be searched for	**6**
1(b)	1 mark for each point: MP1 Conditional statement to identify when a bus is late (punctuality < 0) MP2 Count the number of late days for at least one bus route MP3 Total the late minutes for at least one bus route MP4 Calculation of average minutes late 1 mark for each point (max **three** points): MP5 Initialisation of counting/totalling variables MP6 Iteration through days MP7 Checking all buses… MP8 Count late days and total minutes for all bus routes MP9 Output of number of late arrivals or average minutes late for at least one bus route… MP10 Output complete with all bus routes with late arrivals and average minutes late, with appropriate messages **Example algorithm on next page**	**6**

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

Example algorithm

```
CountA ← 0; CountB ← 0; CountC ← 0; CountD ← 0;
CountE ← 0; CountF ← 0
TotalA ← 0; TotalB ← 0; TotalC ← 0; TotalD ← 0;
TotalE ← 0; TotalF ← 0
FOR Days ← 0 to 19
   IF BusA[Days] < 0
      THEN
         CountA ← CountA + 1
         TotalA ← TotalA + BusA[Days]
   ENDIF
   IF BusB[Days] < 0
      THEN
         CountB ← CountB + 1
         TotalB ← TotalB + BusBA[Days]
   ENDIF
   IF BusC[Days] < 0
      THEN
         CountC ← CountC + 1
         TotalC ← TotalC + BusC[Days]
   ENDIF
   IF BusD[Days] < 0
      THEN
         CountD ← CountD + 1
         TotalD ← TotalD + BusD[Days]
   ENDIF
   IF BusE[Days] < 0
      THEN
         CountE ← CountE + 1
         TotalE ← TotalE + BusE[Days]
   ENDIF
   IF BusF[Days] < 0
      THEN
         CountF ← CountF + 1
         TotalF ← TotalF + BusF[Days]
   ENDIF
NEXT
PRINT "The number of late days for each bus route are:
Bus A "CountA", Bus B "CountB", Bus C "CountC", Bus D ",
CountD", Bus E ", CountE", Bus F "CountF
PRINT "The average number of minutes late for each route
are: Bus A "TotalA/20", Bus B "TotalB/20", Bus C
"TotalC/20", Bus D ", TotalD/20", Bus E ", TotalE/20",
Bus F "TotalF/20
```

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

Question	Answer	Marks
1(c)	Explanation of how the candidate's program performed the following: **Three** from: MP1 The input stored as a variable MP2 The method used to find the position of the day in the Day array that matches the input MP3 The array index is stored as a variable MP4 The index variable used as the array index for each bus array MP5 …and the contents of each array stored/output.	**3**
1(d)	**Two** from: MP1 Add a user input and prompt to enter the number of **weeks** required to record data on arrival times MP2 Store the user input for number of weeks as a variable MP3 Calculation to change number of weeks to number of days MP4 Replace the upper limit of the loop with a variable MP5 Increase the maximum size of the arrays to accommodate a higher number of weeks	**2**

Question	Answer	Marks
2	Many possible answers, those given are examples only. 1 mark for each correct description and 1 mark for each correct example **Char** Description: A single character (from the keyboard) Example: A / # / 2 **String** Description: An (ordered) sequence of characters Example: Hello world / #123?Y / 234 78963 **Boolean** Description: A data type with two possible values Example: TRUE / FALSE	**6**

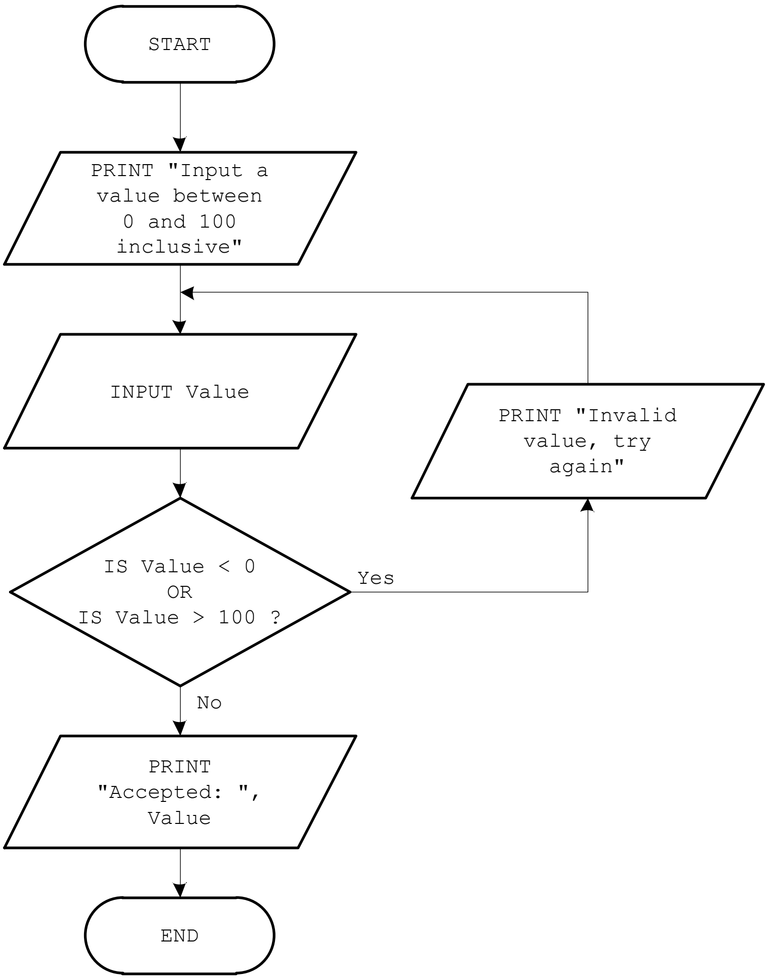
Question	Answer	Marks

3(a)	Many possible answers, those given are examples only.
	1 mark per bullet:
	• IF
	• Condition and outcome
	
	Example answer:
	`IF X < 0`
	`   THEN`
	`      PRINT "Negative"`
	`   ELSE`
	`      PRINT "Not negative"`
	`ENDIF`
	
	OR
	
	1 mark per bullet:
	• CASE
	• Condition and outcome
	
	Example answer:
	`CASE X OF`
	`   1: PRINT ("ONE")`
	`   2: PRINT ("TWO")`
	`   OTHERWISE PRINT ("Less than ONE or more than TWO")`
	`ENDCASE`
		2

3(b)	• To allow different routes through a program • dependent on meeting certain criteria	**2**

Question	Answer	Marks
4(a)	Range check	**1**
4(b)	**Two** from: • The entered number (`Value)` is being checked to see that it is not < 0 or not > 100 • If it is, it is rejected and the user has to enter another number / an error message is displayed • Otherwise the number is accepted, the word ‘Accepted’ is output along with the `Value`	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(c)</td><td>
<table>
<tr><th>Value</th><th>OUTPUT</th></tr>
<tr><td></td><td>Input a value between 0 and 100 inclusive</td></tr>
<tr><td>200</td><td>Invalid value, try again</td></tr>
<tr><td>300</td><td>Invalid value, try again</td></tr>
<tr><td>–1</td><td>Invalid value, try again</td></tr>
<tr><td>50</td><td>Accepted: 50</td></tr>
<tr><td></td><td></td></tr>
</table>
1 mark – Value column
1 mark – OUTPUT column first line
1 mark – OUTPUT column lines two to five
</td><td><b>3</b></td></tr>
</table>

Question	Answer	Marks
4(d)	 START PRINT "Input a value between 0 and 100 inclusive" INPUT Value IS Value < 0 OR IS Value > 100 ? Yes PRINT "Invalid value, try again" No PRINT "Accepted: ", Value END 1 mark – Input prompt and input value 1 mark – Correct decision box labelled sufficiently (Yes/No) – allow 2 decision boxes 1 mark – Remaining outputs correct 1 mark – All connecting lines and arrows to be complete and correct 1 mark – Standard flowchart symbols used	**5**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>Each data value is unique</td><td>1</td></tr>
<tr><td>5(b)</td><td>10 records</td><td>1</td></tr>
<tr><td>5(c)</td><td>Bev07 Lemonade
Bev01 Cola

1 mark for each correct content
1 mark for each correct format
1 mark for correct order</td><td>3</td></tr>
<tr><td>5(d)</td><td>
<table>
<tr><td>Field:</td><td>BevNo</td><td>BevName</td><td>Calories</td><td></td><td></td><td></td></tr>
<tr><td>Table:</td><td>BEVERAGES</td><td>BEVERAGES</td><td>BEVERAGES</td><td></td><td></td><td></td></tr>
<tr><td>Sort:</td><td></td><td>Ascending</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>☑</td><td>□</td><td>□</td><td>□</td><td>□</td></tr>
<tr><td>Criteria:</td><td></td><td></td><td>>45</td><td></td><td></td><td></td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
1 mark for correct Field row
1 mark for Table and Sort rows
1 mark for correct Show row
1 mark for correct Criteria rows</td><td>4</td></tr>
</table>