COMPUTER SCIENCE	**2210/21**
Paper 1	**May/June 2019**
MARK SCHEME	
Maximum Mark: 50	

Published

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

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

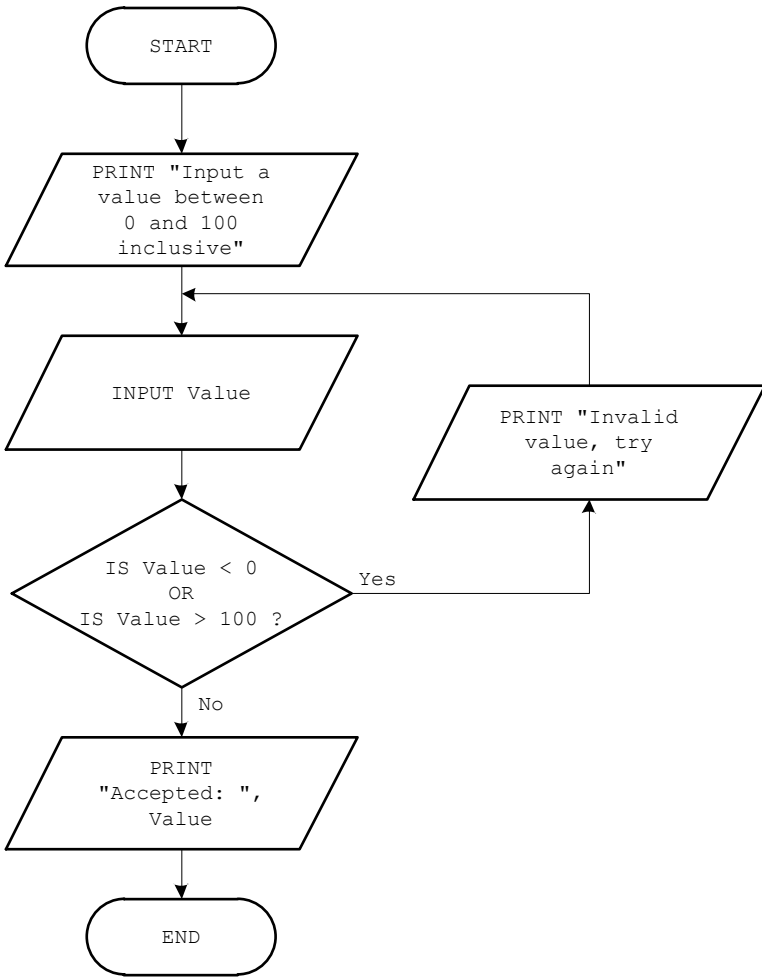| Question	Answer	Marks

1(b)	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
	``	

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**

Question	Answer		Marks
4(c)			**3**

Value	OUTPUT
	Input a value between 0 and 100 inclusive
200	Invalid value, try again
300	Invalid value, try again
−1	Invalid value, try again
50	Accepted: 50

1 mark – Value column
1 mark – OUTPUT column first line
1 mark – OUTPUT column lines two to five

Question	Answer	Marks
4(d)		5

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

Question	Answer	Marks
5(a)	Each data value is unique	1
5(b)	10 records	1
5(c)	Bev07 Lemonade Bev01 Cola 1 mark for each correct content 1 mark for each correct format 1 mark for correct order	3
5(d)	(see query grid below)	4

5(d)

Field:	BevNo	BevName	Calories			
Table:	BEVERAGES	BEVERAGES	BEVERAGES			
Sort:		Ascending				
Show:	☑	☑	☐	☐	☐	☐
Criteria:			>45			
or:						

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