Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER			

2831013016

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **February/March 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

INFORMATION

- The total mark for this paper is 75.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Any blank pages are indicated.

DC (LK/SG) 334324/4

1 Tick (✓) **one** box to show which word accurately describes the scope of a variable declared in a procedure.

A Function ☐

B Global ☐

C Local ☐

D Subroutine ☐

[1]

2 **(a)** **Four** descriptions and **five** pseudocode statements are shown.

Draw **one** line to link each description to its most appropriate pseudocode statement. **Not** all pseudocode statements will be used.

Description	Pseudocode statement
a statement to count	`FOR Count ← 1 TO 10`
a statement to total	`Value ← Value + NewValue`
a statement to start a pre-condition loop	`WHILE Value > 10 DO`
a statement to start a post-condition loop	`Value ← Value + 1`
	`REPEAT`

[4]

(b) Write an algorithm in pseudocode, using a single loop, to output the average of 50 numbers that have been stored in the array `Number[]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

3 Describe the purpose of test data. Include an example of a type of test data in your answer.

Description .....

.....

.....

.....

Example .....

.....

[3]

4 Describe how variables and constants are used in programming.

.....

.....

.....

.....

.....

.....

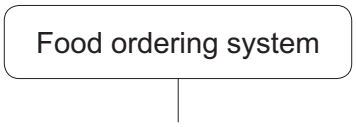
.....

..... [3]

5 A food ordering system is an example of a computer system that is made up of sub-systems.

The food ordering system:
- allows the user to enter the details of the food they want to order and to pay for the order
- displays food available as pictures or as a list.

Complete the structure diagram for the given parts of the food ordering system.



[4]

6 The energy efficiency of an electrical appliance is the percentage of useful energy out compared with the total energy in.

An algorithm has been written in pseudocode to calculate the energy efficiency of an appliance. Values for total energy in and useful energy out are input. The efficiency is calculated and output as a percentage.
The entry of the number –1 for either value stops the algorithm.

```
01 REPEAT
02     OUTPUT "Enter total energy in "
03     INPUT TotalEnergyIn
04     OUTPUT "Enter useful energy out "
05     OUTPUT UsefulEnergyOut
06     IF TotalEnergyIn <> -1 AND UsefulEnergy <> -1
07       THEN
08         Efficiency ← (UsefulEnergyOut / TotalEnergyIn) * 100
09         OUTPUT "Efficiency is ", Efficiency, "%"
10     ENDIF
11 UNTIL TotalEnergyIn <> -1 OR UsefulEnergyOut <> -1
```

(a) Identify the **three** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(b) Write pseudocode to check for an efficiency of 92% or over for this appliance and to output "A-rated" if the efficiency is 92% or over.

.....

.....

.....

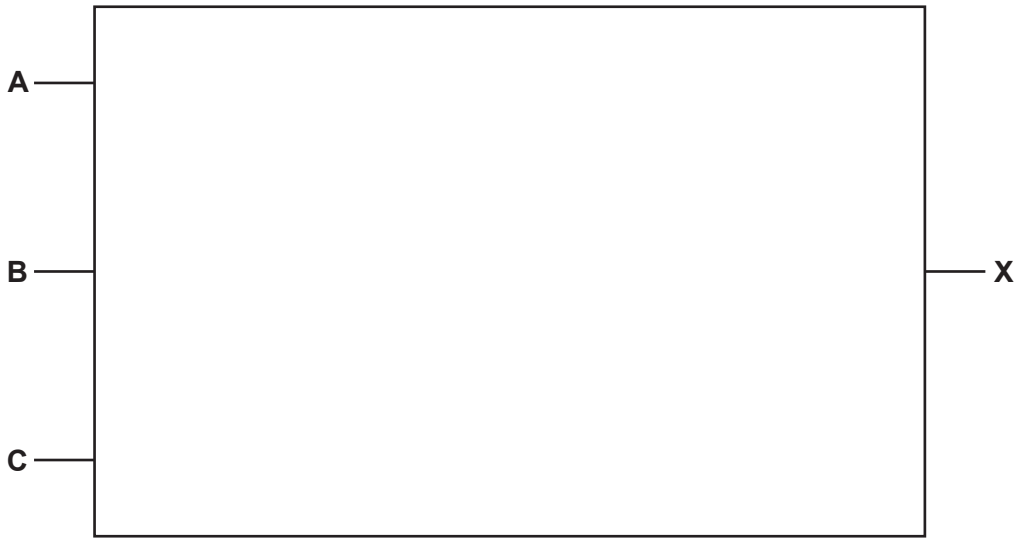
.....

..... [2]

7 Consider this logic expression.

X = (**A** OR NOT **B**) AND (**B** AND NOT **C**)

(a) Draw a logic circuit for this logic expression. Each logic gate must have a maximum of **two** inputs. Do **not** simplify this logic expression.



[5]

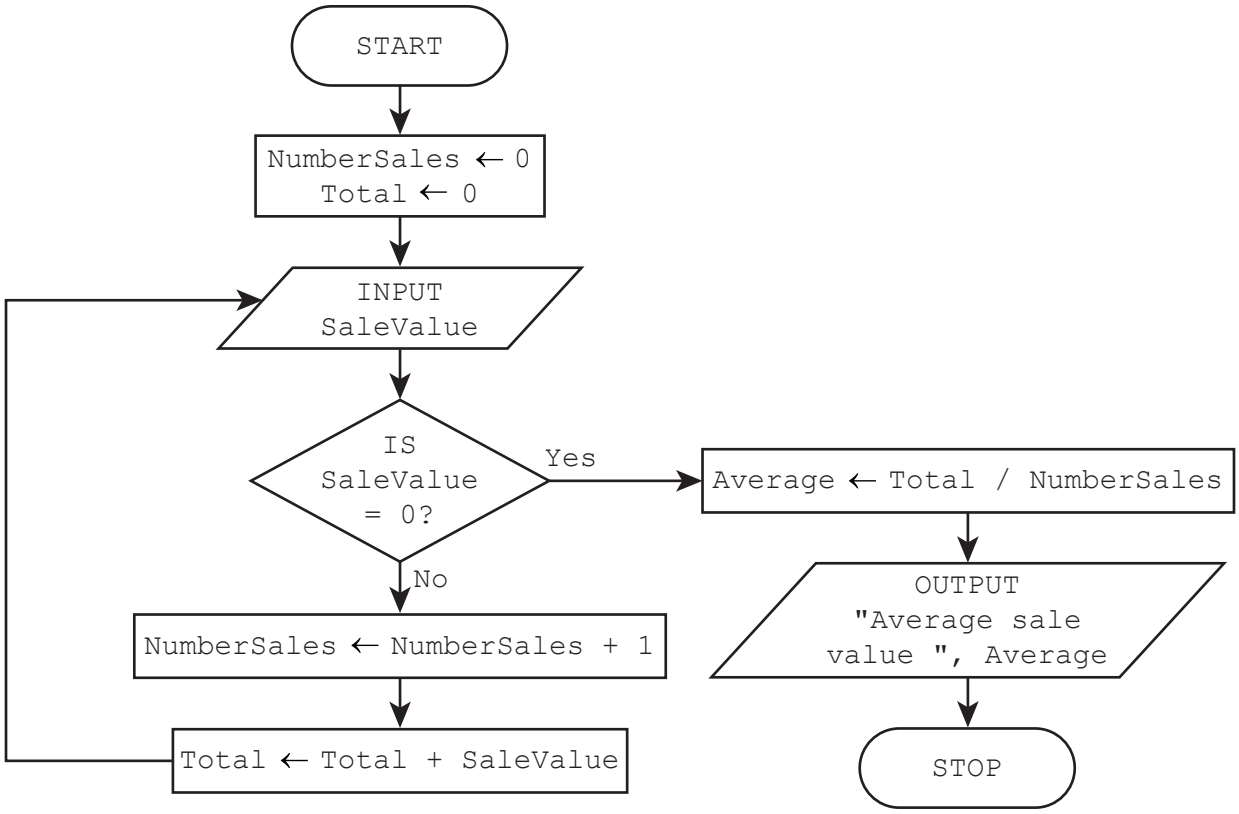
(b) Complete the truth table from the given logic expression.

A	**B**	**C**	Working space	**X**
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

[4]

BLANK PAGE

8 This flowchart represents an algorithm to find the average value of a number of sales.



(a) Complete the trace table using this data:
5.50, 3.40, 6.25, 3.85, –11.00, 0

NumberSales	Total	SaleValue	Average	OUTPUT

[4]

(b) Identify the error in the algorithm and describe how to correct it.

Error .....

.....

Correction .....

.....

.....

.....

[3]

9 A shop that sells books has set up a new database table called `BookList` to store book details. Part of this table is given.

CatNo	Title	Fiction	Author	PaperBack	Price	StockLevel
BK01	The Princes' Story	Yes	B Penn	Yes	4.50	500
BK02	The Princesses' Story	Yes	B Penn	Yes	4.50	350
BK03	Computer Science	No	Way Yu	Yes	19.99	20
BK04	The Modern World	No	P Patel	No	25.00	5
BK05	The Ancient World	Yes	P Patel	No	25.00	5
BK06	Computer Science	No	R Dale	Yes	27.35	8
BK07	The Princes' Story	Yes	B Penn	No	12.50	3
BK08	The Princesses' Story	Yes	B Penn	No	12.50	0
BK12	Famous Five	Yes	E Bly	Yes	2.75	45
BK15	Secret Seven	Yes	E Bly	Yes	2.75	25
BK16	The Last Knight	Yes	P Mann	Yes	5.99	7
BK17	The Dark Tower	Yes	P Mann	Yes	5.99	5
BK19	The Final Chase	Yes	P Mann	Yes	5.99	5
BK21	Maths Today Part 1	No	B Ward	Yes	6.75	25
BK22	Maths Today Part 2	No	B Ward	Yes	6.75	15
BK23	Maths Today Part 3	No	B Ward	Yes	6.75	10
BK26	Maths Today Workbook	No	B Ward	Yes	6.75	30
BK27	Knitting for Beginners	No	A Smith	Yes	6.99	3
BK30	Woodwork for Beginners	No	A Smith	Yes	6.99	4
BK31	Networking for Beginners	No	A Smith	Yes	6.99	0

(a) State the number of records in this part of the database table.

..... [1]

(b) **(i)** Give the name of the field that would be used for the primary key.

..... [1]

(ii) State the reason for choosing this field for the primary key.

.....

..... [1]

(c) Complete the table to identify the most appropriate data type for each field based on the data shown in the table `BookList`

Field	Data type
`CatNo`	
`Title`	
`Fiction`	
`Price`	

[2]

(d) Write the output from this structured query language (SQL) statement.

```
SELECT CatNo, Title, Author

FROM BookList

WHERE StockLevel = 0;
```

.....

.....

.....

..... [2]

(e) Complete this SQL statement to display all the titles by the author B Penn.

```
SELECT .....

FROM .....

WHERE .....;
```

[2]

10 The variables `X`, `Y` and `Z` are used in a program: `X` stores a whole number, `Y` stores a decimal number and `Z` stores a flag that can be set to `TRUE` or `FALSE`

(a) Write pseudocode statements to declare the variables `X`, `Y` and `Z`

.....

.....

..... [3]

(b) The function `Same(A,B)` returns `TRUE` if the value of `A` is the same as the value of `B` when `B` is rounded to the nearest whole number and `FALSE` otherwise.

Write pseudocode statements to:
- define the function
- call the function with `X` and `Y` and store the return value in `Z`

Function definition .....

.....

.....

.....

.....

.....

.....

.....

Function call .....

.....

.....

.....
[6]

(c) State the difference between defining and calling a function.

.....

..... [1]

BLANK PAGE

11 The one-dimensional (1D) array `TeamName[]` contains the names of teams in a sports league. The two-dimensional (2D) array `TeamPoints[]` contains the points awarded for each match. The position of each team's data in the two arrays is the same. For example, the team stored at index 10 in `TeamName[]` and `TeamPoints[]` is the same.

The variable `LeagueSize` contains the number of teams in the league. The variable `MatchNo` contains the number of matches played. All teams have played the same number of matches.

The arrays and variables have already been set up and the data stored.

Each match can be played at home or away. Points are recorded for the match results of each team with the following values:

- 3 – away win
- 2 – home win
- 1 – drawn match
- 0 – lost match.

Write a program that meets the following requirements:

- calculates the total points for all matches played for each team
- counts the total number of away wins, home wins, drawn matches and lost matches for each team
- outputs for each team:
  - name
  - total points
  - total number of away wins, home wins, drawn matches and lost matches
- finds and outputs the name of the team with the highest total points
- finds and outputs the name of the team with the lowest total points.

You must use pseudocode or program code **and** add comments to explain how your code works.

You do **not** need to declare any arrays, variables or constants; you may assume that this has already been done.

All inputs and outputs must contain suitable messages.

You do **not** need to initialise the data in the arrays `TeamName[]` and `TeamPoints[]` or the variables `LeagueSize` and `MatchNo`

.....

.....

.....

.....

.....

.....

.....

.....

.....

[15]

Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **February/March 2023**

MARK SCHEME

Maximum Mark: 75

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 February/March 2023 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

PUBLISHED

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.

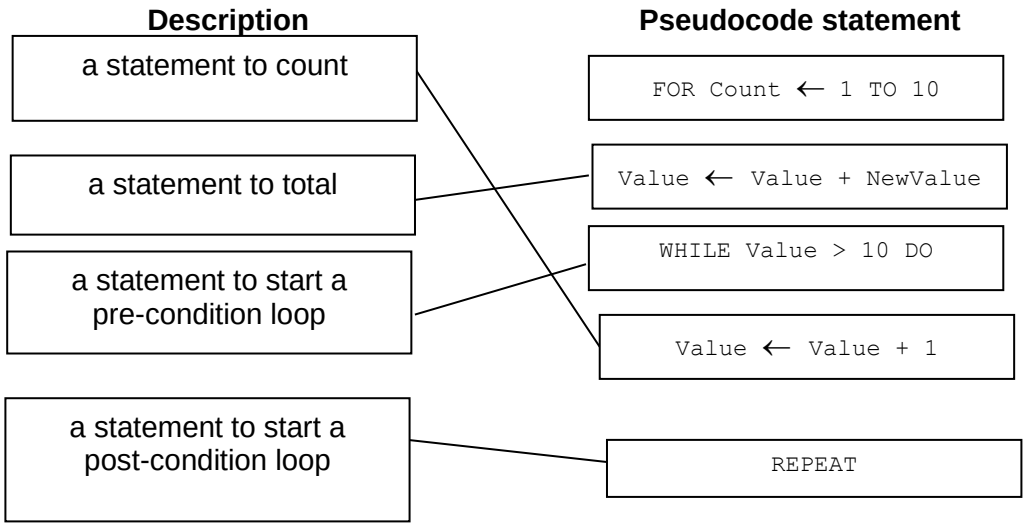
GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Mark scheme abbreviations

/	separates alternative words / phrases within a marking point
//	separates alternative answers within a marking point
underline	actual word given must be used by candidate (grammatical variants accepted)
max	indicates the maximum number of marks that can be awarded
( )	the word / phrase in brackets is not required, but sets the context

Note: No marks are awarded for using brand names of software packages or hardware.

Question	Answer	Marks
1	**C**	**1**

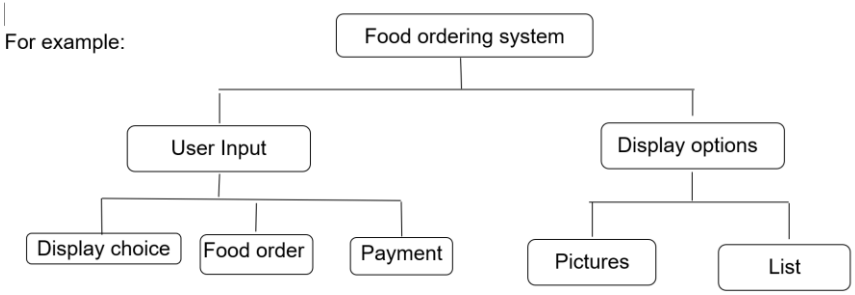
Question	Answer	Marks
2(a)	**One** mark for each **single** correct line from a description to statement 	**4**

Question	Answer	Marks

2(b)	**One** mark for each point:
	• Initialisation of total to zero before loop
	• appropriate loop controls
	• totalling statement inside the loop, must use array `Number[]` with an index
	• calculation of average outside loop
	• output of average outside loop
	
	Example:
	`Total ← 0`
	`FOR Count ← 1 TO 50`
	`  Total ← Total + Number[Count]`
	`NEXT Count`
	`Average ← Total / 50`
	`OUTPUT "The average is ", Average`
		5

Question	Answer	Marks
3	**One** mark for each point **max two**. • check that the program works as expected • check for **logic/runtime** errors • check that the program **rejects any invalid** data that is input • check that the program **only accepts** reasonable data **One** mark for example Normal // erroneous // abnormal // extreme // boundary	**3**

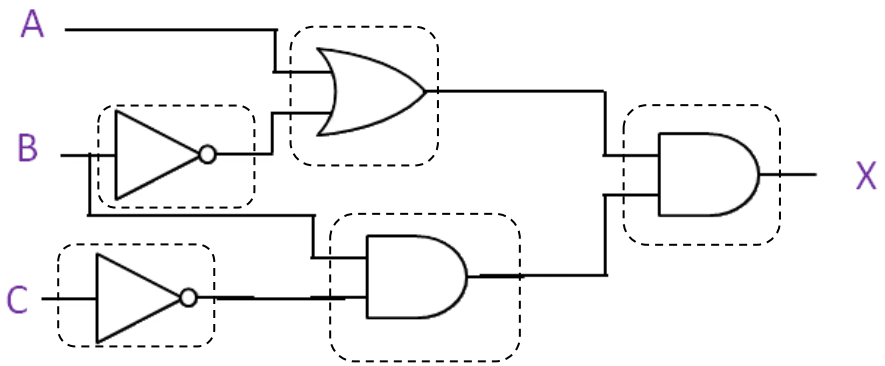
Question	Answer	Marks
4	**One** mark for each point **max three**. • variables / constants are used to store items of data • the data stored in variables / constants are accessed by an identifier // named data stores • the value of a variable may change during the execution of a program • the value of a constant will remain the same during the execution of a program	**3**

Question	Answer	Marks
5	**One** mark for a suitable hierarchical structure **One** mark for suitable names for the sub systems for user input and display options **One** mark for sub systems for user inputs, (choice of display,) food order and payment **One** mark for sub systems for display output types, pictures and list For example:  	**4**

Question	Answer	Marks

6(a)	**One** mark for each error identified and correction
	• Line `05 OUTPUT UsefulEnergyOut` should be `INPUT UsefulEnergyOut`
	• Line `06 IF TotalEnergyIn <> -1 AND UsefulEnergy <> -1` should be:
	`IF TotalEnergyIn <> -1 AND UsefulEnergyOut <> -1`
	• Line `11 UNTIL TotalEnergyIn <> -1 OR UsefulEnergyOut <> -1` should be:
	`UNTIL TotalEnergyIn = -1 OR UsefulEnergyOut = -1`
		3

6(b)	**One** mark for checking for >= 92
	One mark for outputting "**A-rated**" only if the condition is met
	For example
	`IF Efficiency >= 92`
	`  THEN`
	`  OUTPUT "A-rated"`
	`ENDIF`
		2

Question	Answer	Marks
7(a)	**One** mark for each correct gate, with the correct inputs as shown. 	**5**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>7(b)</td><td>
<table>
<tr><th>A</th><th>B</th><th>C</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
4 marks for 8 correct outputs
3 marks for 6/7 correct outputs
2 marks for 4/5 correct outputs
1 mark for 2/3 correct outputs
</td><td>4</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>8(a)</td><td>
<table>
<tr><th>NumberSales</th><th>Total</th><th>SaleValue</th><th>Average</th><th>OUTPUT</th></tr>
<tr><td>0</td><td>0</td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>5.50</td><td>5.50</td><td></td><td></td></tr>
<tr><td>2</td><td>8.90</td><td>3.40</td><td></td><td></td></tr>
<tr><td>3</td><td>15.15</td><td>6.25</td><td></td><td></td></tr>
<tr><td>4</td><td>19.00</td><td>3.85</td><td></td><td></td></tr>
<tr><td>5</td><td>8.00</td><td>-11.00</td><td></td><td></td></tr>
<tr><td></td><td></td><td>0</td><td>1.6</td><td>Average sale value 1.6</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
One mark for each column NumberSales, Total, SaleValue
One mark for columns Average and OUTPUT
</td><td>4</td></tr>
<tr><td>8(b)</td><td>Error – one mark for identification of error
for example including negative numbers / not differentiation between negative and positive values
Correction
One mark for correct placement and one mark for appropriate action
For example – after the input box insert a decision box to reject negative numbers</td><td>3</td></tr>
</table>

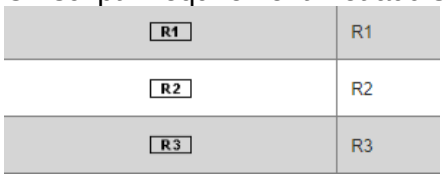
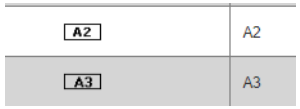
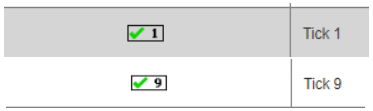
<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(a)</td><td>20</td><td>1</td></tr>
<tr><td>9(b)(i)</td><td>CatNo</td><td>1</td></tr>
<tr><td>9(b)(ii)</td><td>It is a unique identifier // no repeated values</td><td>1</td></tr>
<tr><td>9(c)</td><td>Two marks for 4 correct data types or one mark for 2 or 3 correct data types
<table>
<tr><th>Field</th><th>Data type</th></tr>
<tr><td>CatNo</td><td>Text/Alphanumeric</td></tr>
<tr><td>Title</td><td>Text/alphanumeric</td></tr>
<tr><td>Fiction</td><td>Boolean</td></tr>
<tr><td>Price</td><td>Real</td></tr>
</table></td><td>2</td></tr>
<tr><td>9(d)</td><td>One mark for each correct row
BK08 The Princesses’ Story B Penn
BK31 Networking for Beginners A Smith</td><td>2</td></tr>
<tr><td>9(e)</td><td>One mark if two correct or two marks if completely correct
Title
BookList
Author = "B Penn" // Author = 'B Penn' // Author Like "B Penn"</td><td>2</td></tr>
</table>

Question	Answer	Marks

10(a)	**One** mark for each correct line
	`DECLARE X : INTEGER`
	`DECLARE Y : REAL`
	`DECLARE Z : BOOLEAN`
		3

10(b)	**One** mark for using `FUNCTION` and `ENDFUNCTION` and `RETURNS BOOLEAN`
	One mark for naming the function `Same`
	One mark for defining the two parameters correctly
	One mark for comparing the two parameters using `ROUND`
	One mark for correctly returning `TRUE` and `FALSE`
	One mark for correct function call
	
	Example definition:
	`FUNCTION Same(A : INTEGER, B : REAL) RETURNS BOOLEAN`
	`  IF A = ROUND(B,0)`
	`  THEN`
	`  RETURN TRUE`
	`  ELSE`
	`  RETURN FALSE`
	`  ENDIF`
	`ENDFUNCTION`
	
	Example call:
	`Z ← Same(X,Y)`
		6

10(c)	A function is defined once **and** called many times **or** Define – setting up the function **and** call is using a function	**1**

Question	Answer	Marks
11	Read and understand the question before starting to mark any scripts. Read the whole answer before marking a script: Check if each requirement listed below has been met. Requirements may be met using a suitable built-in function from the programming language used (Python, VB.NET or Java) On script if requirement met add seen, NE if partial attempt, cross if no attempt (see marked scripts).  Use the tables for A02 and A03 below to award a mark in a suitable band using a best fit approach, then add up the total. Marks are available for: • AO2 (maximum 9 marks) • AO3 (maximum 6 marks)   **Data Structures required** with names as given in the scenario Arrays or lists `TeamName`, `TeamPoints` Variables `LeagueSize`, `MatchNo` **Requirements (techniques)** **R1** calculates total points for all matches played by each team (nested iteration, totalling) **R2** counts and outputs, with the team's name, for each team the total number of away wins, home wins, drawn matches and lost matches (nested iteration, counting, output) **R3** finds and outputs the name of the team with the highest number of points and the name of the team with the lowest number of points. (output, selection)	**15**

Question	Answer	Marks

11	***Example 15-mark answer in pseudocode:***
	`// meaningful identifier names and appropriate data structures to store the data required`
	`DECLARE TeamCounter : INTEGER`
	`DECLARE MatchCounter : INTEGER`
	`FOR TeamCounter ← 1 to LeagueSize // zero totals for each club's results`
	`  TotalPoints[TeamCounter] ← 0`
	`NEXT TeamCounter`
	
	`FOR TeamCounter ← 1 TO LeagueSize`
	`  AwayWinNo ← 0  // zero totals for each club's result details`
	`  HomeWinNo ← 0`
	`  DrawNo ← 0`
	`  LostNo ← 0`
	`  FOR MatchCounter ← 1 TO MatchNo`
	`  TotalPoints[TeamCounter] ← TotalPoints[TeamCounter] +`
	`  TeamPoints[TeamCounter, MatchCounter]`
	`  CASE OF TeamPoints[TeamCounter, MatchCounter]`
	`  3 : AwayWinNo ← AwayWinNo + 1`
	`  2 : HomeWinNo ← HomeWinNo + 1`
	`  1 : DrawNo ← DrawNo + 1`
	`  0 : LostNo ← LostNo + 1`
	`  ENDCASE`
	`  NEXT MatchCounter`
	
	`  OUTPUT "Team ", TeamName[TeamCounter]  // Output details of a team's results`
	`  OUTPUT "Total points ", TotalResult[TeamCounter]`
	`  OUTPUT "Away wins ", AwayWinNo`
	`  OUTPUT "Home wins ", HomeWinNo`
	`  OUTPUT "Draws ", DrawNo`
	`  OUTPUT "Losses ", LostNo`	

Question	Answer	Marks

11	`// Check for highest and lowest results`
	`  IF TeamCounter = 1`
	`  THEN`
	`  HighestResult ← TotalPoints[TeamCounter]`
	`  LowestResult ← TotalPoints[TeamCounter]`
	`  ENDIF`
	
	`  IF TotalPoints[TeamCounter] > HighestResult`
	`  THEN`
	`  HighestResult ← TotalPoints[TeamCounter]`
	`  TopTeam ← TeamCounter`
	`  ENDIF`
	`  IF TotalPoints[TeamCounter] < LowestResult`
	`  THEN`
	`  LowestResult ← TotalPoints[TeamCounter]`
	`  BottomTeam ← TeamCounter`
	`  ENDIF`
	
	`NEXT TeamCounter`
	
	`// output names of the teams with the highest and lowest number of points`
	`OUTPUT "Top Team ", TeamName[TopTeam]`
	`OUTPUT "Bottom Team ", TeamName[BottomTeam]`	

Marking Instructions in italics			
AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems			
0	**1–3**	**4–6**	**7–9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check list of techniques needed.*	The range of programming techniques used is appropriate to the problem. *All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

Marking Instructions in italics			
AO3: Provide solutions to problems by: **evaluating computer systems**		**making reasoned judgements**	**presenting conclusions**
0	**1–2**	**3–4**	**5–6**
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented
	Some identifier names used are appropriate. *Some of the data structures used have meaningful names.*	The majority of identifiers used are appropriately named. *Most of the data structures used have meaningful names.*	Suitable identifiers with names meaningful to their purpose have been used throughout. *All of the data structures used have meaningful names.*
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. *Solution contains few lines of code with errors that attempt to perform a task given in the scenario.*	The solution contains parts that are inaccurate. *Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.*	The solution is accurate. *Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.*
	The solution attempts at least one of the requirements. *Solution contains lines of code that attempt at least one task given in the scenario.*	The solution meets most of the requirements. *Solution contains lines of code that perform most tasks given in the scenario.*	The solution meets all the requirements given in the question. *Solution performs all the tasks given in the scenario.*