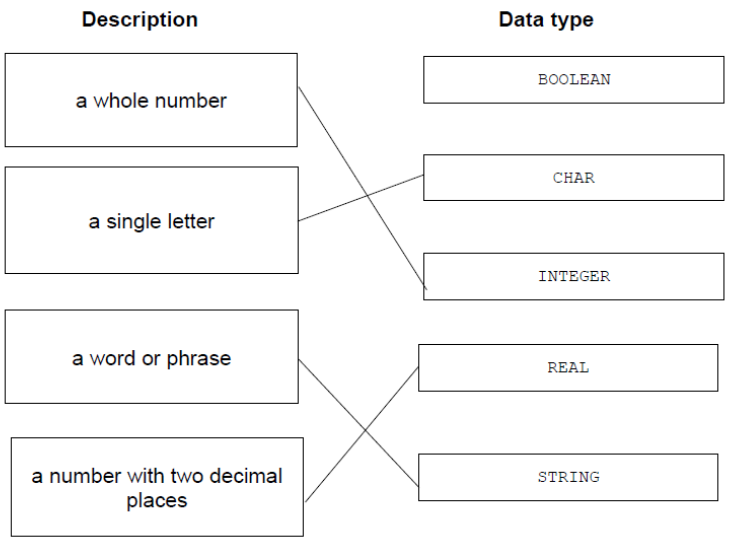
8 Algorithms & Programming
ANSWERS

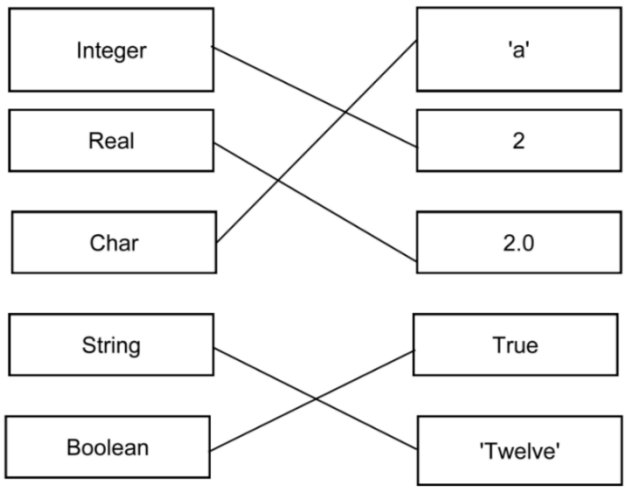
Question	Answer	Marks
1	C	1

Question	Answer	Marks
2	B	1

Question	Answer	Marks
3	One mark for each correct line from description to data type 	4

Question	Answer	Marks
4	One mark for each correct word • array • constant • variable	3

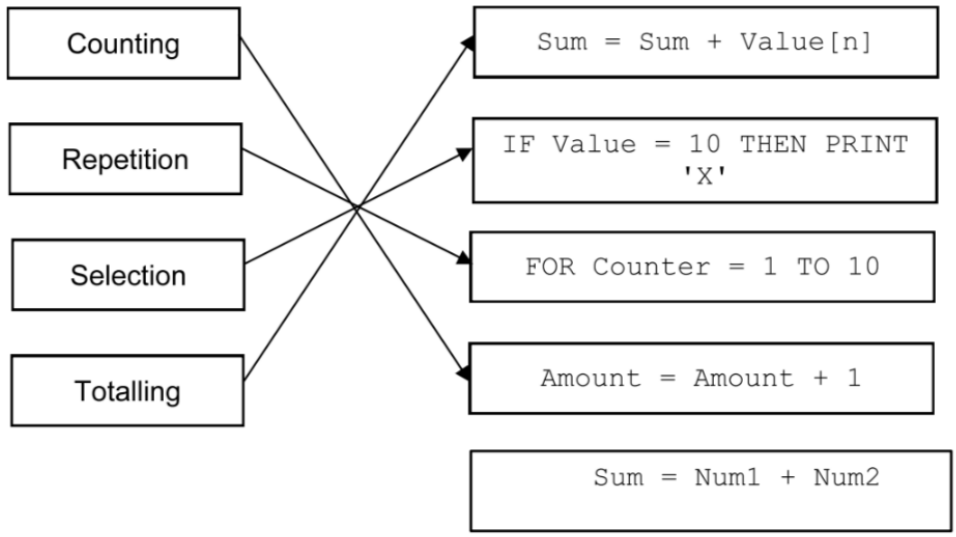
4 1 mark for each correct link, up to maximum of 4 marks



[4]

8 Algorithms & Programming
ANSWERS

4 1 mark for each correct line, two lines from one box not allowed

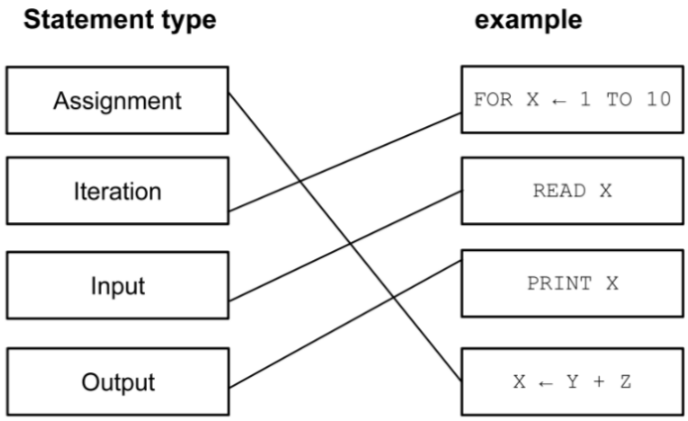


[4]

5
- `FOR (… TO … NEXT)…`
- … a set number of iterations
- `WHILE (… DO … ENDWHILE) …`
- … used where the loop may never be executed/whilst a specified condition exists

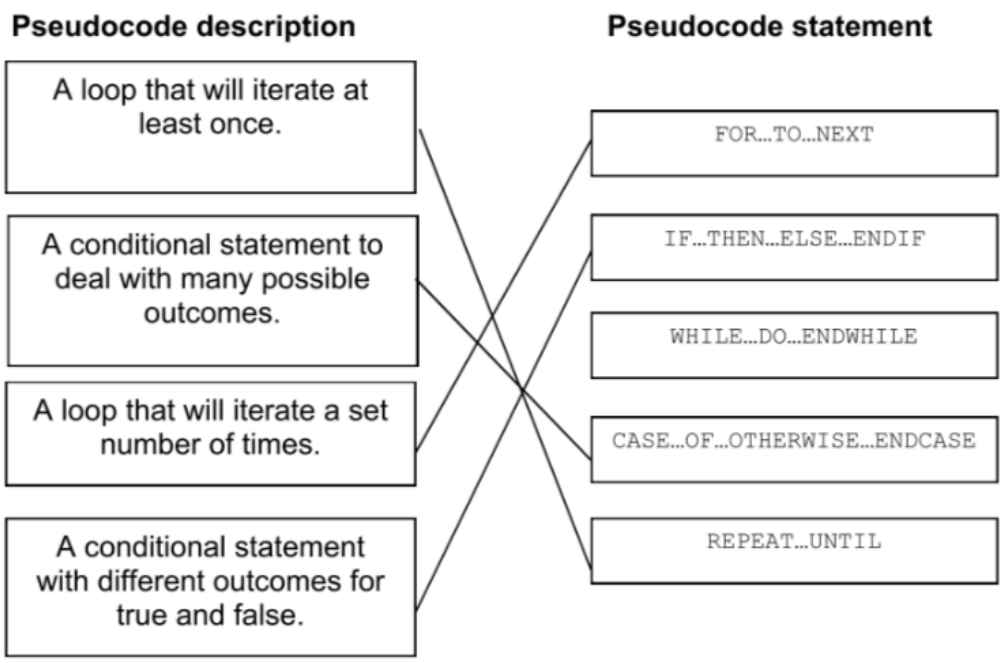
[4]

4 **1** mark for each correct line, maximum **3** (zero correct **0**, one correct **1**, two correct **2**, three or four correct **3**), each box must have only one connection.

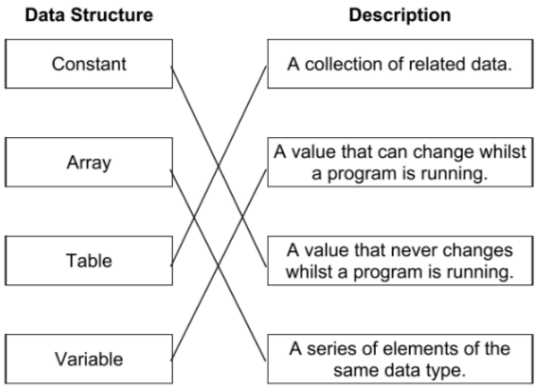


[3]

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
4(a)	1 mark for each correct line 	**4**

4(b)	1 mark per bullet:
	Appropriate loop controls
	Read from array
	Print from array (the last two points can be in one statement)
	Note reading and printing **MUST** be within the same loop
	Example algorithm:
	`Count ← 0`
	`WHILE Count < 50 DO`
	`  OUTPUT Name[Count]`
	`  Count ← Count + 1`
	`ENDWHILE`
		3

Question	Answer	Marks
3	1 mark for each correct line, max **3** marks. 	**3**

8 Algorithms & Programming
ANSWERS

4 There are many possible correct answers this is an example only.

Normal	e.g. 1.7
Extreme	0.5 **or** 2.0 only
Abnormal	e.g. one

[3]

5 – `IF (… THEN … ELSE … ENDIF)`
– `CASE (… OF … OTHERWISE … ENDCASE)` [2]

4 (a) (i) Normal

(ii) Acceptable data to test that the results are as expected. [2]

(b) **One** mark for the data set, **one** mark for the type and **one** mark for the matching reason
There are many possible correct answers this is an example only.

Set 1	–	Age 4, height 0.9
Type	–	Boundary/Extreme
Reason	–	Data to test the validation that is just within the limits of acceptability
Set 2	–	Age 10, height 1.4
Type	–	Abnormal
Reason	–	Data that should be rejected and produce an error message

[6]

Question	Answer	Marks

5(a)	– initialising counter outside the loop
	– updating counter inside loop
	– suitable exit value at start of loop
	– correct use of `WHILE … DO … ENDWHILE`
	
	Example:
	
	`INPUT Num`
	`Counter ← 1`
	`WHILE Counter <= 12 DO`
	`  Num ← Num * Counter`
	`  A [Counter] ← Num`
	`  Counter ← Counter + 1`
	`ENDWHILE`
		4

5(b)	– `WHILE` has criteria check at start / pre-test – may never run – `REPEAT UNTIL` has criteria check at end / post-test – will always run at least once	**4**

Question	Answer	Marks

4	2 marks for identification, 1 mark for description, 1 mark for reason.
	
	Identification:
	`CASE …`
	`… OF … OTHERWISE … (ENDCASE)` or
	`… OF … (OTHERWISE) … ENDCASE`
	
	Description:
	– a statement that allows for multiple selections // not any of the above
	
	Reason:
	– to simplify pseudocode/ make pseudocode more understandable etc.
		4

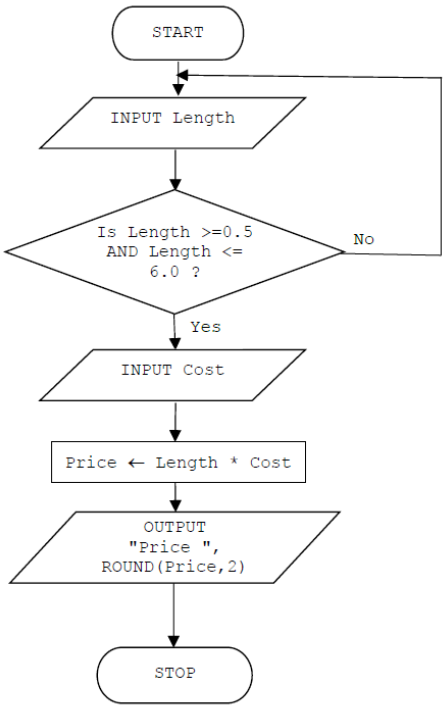
Question	Answer	Marks
6(a)	Displaying/sort 10 names in alphabetical order 9	**1**

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
6(b)	One mark for each point (max four) • Initialisation • inputting 10 names • storing the names in an array • sorting the names in alphabetical order using a bubble sort • displaying the 10 names • iteration	4

6(c)	**One** mark for a meaningful identifier for the array
	`A Names // ArrayNames`
	
	Two marks for 3 meaningful identifiers for variables
	One marks for 1 or 2 meaningful identifiers for variables
	`T Temp`
	`C Counter`
	`L Length`
		3

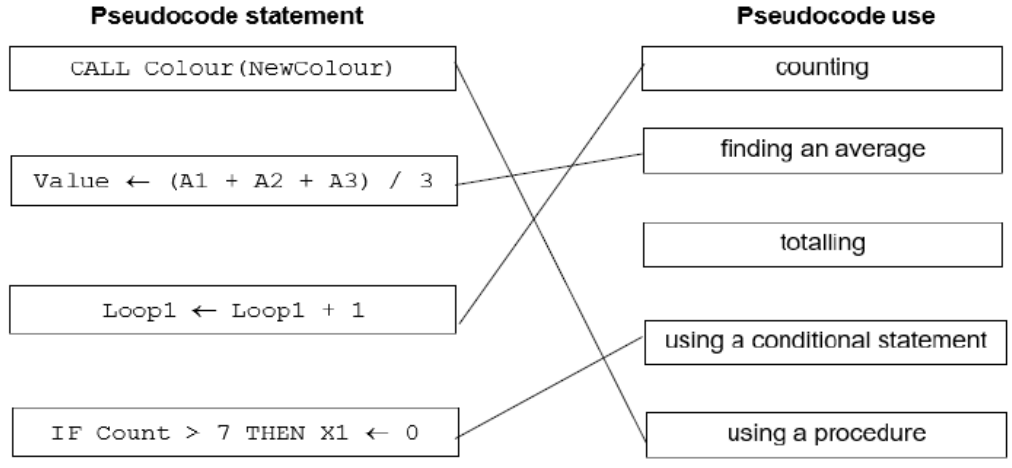
6(d)	One mark for each point (max two) • use of comments • use of procedures/functions • use of white space	2

Question	Answer	Marks
8(a)	Range check	1
8(b)	 **One** mark for each of the following points • correct use of flowchart symbols • working flow lines and complete • both inputs correct • working range check • working calculation • correct output rounded to two decimal places	6
8(c)	One mark for set of test data, one mark for purpose (max four) Example: 1 and 1 (1) normal data to ensure the algorithm accepts this test data (1) –1 and 1 (1) abnormal data for length to ensure that it is rejected (1)	4
8(d)	One mark for two correct headings Two marks for three correct headings Three marks for all headings correct and no other headings unless used in 8(b) **Length** | **Cost** | **Price** | **OUTPUT**	3
8(e)	**One** mark for each point (max two) • validate `Cost` … • … with a range/presence check • add another validation check for `Length`	2

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
1	A	1

Question	Answer	Marks
2(a)	Format check	1
2(b)	**One** mark for each appropriate test data, max **two** **One** mark for each correct accompanying reason, max **two** **For example:** Normal – 30/12/1960 … Reason – … (the date is written in the correct format and) **should be accepted**. Abnormal – 30/Dec/1960 … Reason – … (the month is not written in the correct format and) **should be rejected**.	4
2(c)	**One** mark per mark point, max **two** MP1 check that there are 10 **characters** in total MP2 if the date is **too long/short** it will be **rejected**	2

Question	Answer	Marks
3(a)	**One** mark for each correct line. 	4

Question	Answer	Marks

3(b)	**One** mark per mark point, max **four**
	MP1 initialise a variable to store the lowest number set to a high value (at least 100) / first value in the array
	MP2 loop structure to iterate 25 times
	MP3 use of `IF` to check if the array element is less than the current lowest value
	MP4 …if it is, set this to the lowest value
	MP5 output the result (with an appropriate message) after the loop
	
	Example answer:
	`Min ← 100`
	`FOR Count ← 1 TO 25`
	`  IF Temperatures[Count] < Min`
	`  THEN`
	`  Min ← Temperatures[Count]`
	`  ENDIF`
	`NEXT Count`
	`OUTPUT "The lowest temperature is ", Min`
		4

Question	Answer	Marks
5	**One** mark per mark point, max **three** MP1 variables and constants should have meaningful identifiers MP2 …so that programmers/future programmers are able to understand their purpose MP3 they are **both** used for data storage MP4 constants store values that never change during the execution of a program // by example MP5 variables contain values that have been calculated within the program / can change during the execution of the program // by example	**3**

Question	Answer	Marks

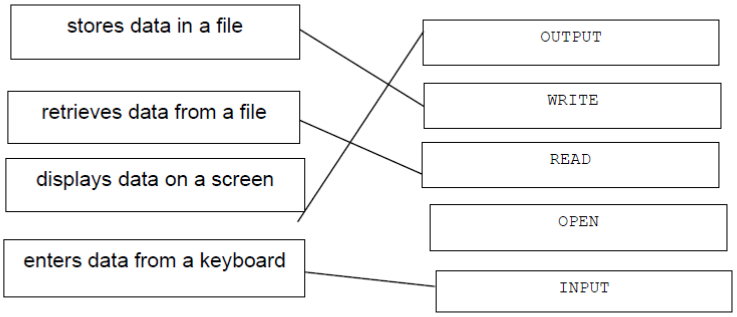
7	**One** mark per mark point, max **five**
	MP1 storing string in `Quote`
	MP2 correct assignment for `Start` // sending correct start value
	MP3 correct assignment for `Number` // sending correct number of characters
	MP4 use of `SUBSTRING` function with first parameter as `Quote`, or equivalent, plus two other parameters
	MP5 correct use of `LCASE` function
	MP6 two correct outputs
	
	For example:
	`Quote ← "Learning Never Exhausts The Mind"`
	`Start ← 25`
	`Number ← 8`
	`OUTPUT SUBSTRING(Quote, Start, Number)`
	`OUTPUT LCASE(Quote)`
		5

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
8	**One** mark per mark point, max **four** Procedures, max **three** MP1 to enable the programmer to write a collection of programming statements under a single identifier MP2 to allow modular programs to be created // to allow procedures to be re-used within the program or in other programs MP3 to make program creation faster because procedures can be re-used // to enable different programmers to work on different procedures in the same project MP4 to make programs shorter (than using the repeated code) / using less duplication of code // to make programs easier to maintain due to being shorter. Parameters, max **three** MP5 to pass values from the main program to a procedure / function MP6 …so that they can be used in the procedure / function MP7 allow the procedure / function to be re-used **with different data**.	**4**

Question	Answer	Marks
1	**A**	**1**

Question	Answer	Marks
2	**B**	**1**

Question	Answer	Marks
3(a)	**One** mark for each correct line from description to pseudocode keyword **Pseudocode description** / **Pseudocode keyword**  	**4**
3(b)	**One** mark for each point (max two) • data is stored permanently • data can be moved to another computer • another copy of data can be made and stored//accessed elsewhere // backup copy	**2**

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
4(a)	**One** mark for each point • type check • range check	**2**

4(b)	**One** mark for each point (max five)
	• use of loop for check
	• checking for whole number
	• checking for number greater than or equal to one
	• … and less than or equal to six
	• Appropriate error/reinput message
	• ability to reinput value
	
	Example:
	`WHILE Seats < 1 OR Seats > 6 OR Seats <> ROUND(Seats, 0) DO`
	`  OUTPUT "Please enter a valid number of seats "`
	`  INPUT Seats`
	`ENDWHILE`
		5

4(c)	**One** mark for correct test data, **one** mark for corresponding reason Example: 7, abnormal data to show that this value would be rejected	**2**

Question	Answer	Marks
7	**one** mark for first description **one** mark for matching difference **max four** • local variables - scope is a defined block of code/subroutine/procedure/function • global variables – scope is the whole program • local variables - value cannot be changed elsewhere in the program • global variables – value can be changed anywhere in the program	**4**

Question	Answer	Marks
2	A	**1**

Question	Answer	Marks
3(a)	**One** mark per mark point, max **two** • Validation is an automated check carried out by a computer • … to make sure the data entered is sensible/acceptable/reasonable	**2**
3(b)	**One** mark for each appropriate test data, max **three** **One** mark for each correct accompanying reason, max **three** **For example:** Normal – 75 Reason – the data lies within the required range **and** should be accepted Abnormal – Sixty Reason – this is the wrong data type **and** should be rejected Extreme – 200 Reason – the highest value in the required range that should be accepted	**6**

Question	Answer	Marks
4	**One** mark per mark point, max **four** `DIV`, max **two** • To perform integer division • Meaning only the whole number part of the answer is retained • Example of `DIV` For example `DIV(9,4) = 2` `ROUND`, max **two** • To return a value rounded to a specified number of digits / decimal places • The result will either be rounded to the next highest or the next lowest value • … depending on whether the value of the preceding digit is >=5 or <5 • Example of `ROUND` for example, `ROUND(4.56, 1) = 4.6`	**4**

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
6	**One** mark for each correct feature, max **two** **One** mark for each correct accompanying reason, max **two** **For example:** Meaningful identifiers – to enable the programmer (or future programmers) to easily recognize the purpose of a variable / array / constant // to enable easy tracking of a variable / constant / array through the program Use of comments – to annotate each section of a program so that a programmer can find specific sections / so that the programmer knows the purpose of that section of code Procedures and functions – to make programs modular and easier to update / add functionality	4

Question	Answer	Marks

8(a)	**One** mark per mark point, max **three**
	• Storing string in `Phrase`
	• Correct use of `LENGTH` function
	• Correct use of `UCASE` function
	• Correct outputs of `LENGTH` and `UCASE`
	
	For example:
	
	`Phrase ← "The beginning is the most important part"`
	`OUTPUT LENGTH(Phrase)`
	`OUTPUT UCASE(Phrase)`
		3

8(b)	**One** mark for each correct line, max **two**
	
	`40`
	`THE BEGINNING IS THE MOST IMPORTANT PART`
		2

Question	Answer	Marks
4(a)	**One** mark for each point (**max three**). • range check with acceptable values is (greater than) zero **and** less than 1000 • presence check to ensure the program will not continue until a value has been entered • type/character check to ensure that a number is entered • length check to ensure there are no more than 3 digits entered	3
4(b)(i)	To verify the data / for verification / as a verification check // to make sure that no changes are made to the data on entry	1

4(b)(ii)	**One** mark for each point (**max three**).
	• use of iteration
	• use of two inputs
	• to check that the two inputs are the same / different
	• use of the given variable `Measurement`
	
	For example
	`REPEAT`
	`  OUTPUT "Please enter measurement "`
	`  INPUT Measurement`
	`  OUTPUT "Please re-enter measurement "`
	`  INPUT MeasurementCheck`
	`UNTIL Measurement = MeasurementCheck`
		3

Question	Answer	Marks
6	**One mark** for each feature and **one mark** for corresponding example (**max six**) • ensuring that all identifiers have meaningful names … • … example using `Total` to store a running total • using comments to explain how the program works … • … example `// all values are zeroed before the next calculation` • using procedures and functions for the tasks within a program … • … example `CalculateInterest(Deposit, Rate)`	6

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks

11(a)	**One** mark for any two correct lines
	
	`DECLARE P : STRING`
	`P ← "The world"`
	`DECLARE Q : CHAR`
	`Q ← 'W'`
		2

Question	Answer	Marks

11(b)	**One** mark for each point (**max four**)
	• converting `P` to upper case
	• finding the length of `P`
	• using a loop to check for position of `Q`
	• using the string operation substring
	• storing the loop counter in `Position` if the value is found
	
	For example:
	`P ← UCASE(P)`
	`Counter ← 1`
	`Position ← 0`
	`REPEAT`
	`  IF SUBSTRING(P, Counter, 1) = Q`
	`  THEN`
	`  Position ← Counter`
	`  ENDIF`
	`  Counter ← Counter + 1`
	`UNTIL Position <> 0 OR Counter = LENGTH(P)`
		4

11(c)	5	**1**

Question	Answer	Marks
1	B	**1**

Question	Answer	Marks
2	**One** mark per mark point, max **four** `MOD`, max **two** • To perform (integer) division when one number is divided by another • … and find the remainder • Allow example e.g. `7 MOD 2 = 1` `RANDOM`, max **two** • To generate (pseudo) random numbers • …(usually) within a specified range • Allow example e.g. `RANDOM() * 10` returns a random number between `0` and `10`	**4**

Question	Answer	Marks
3	**One** mark per mark point, max **three** MP1 A **call statement** is used in order to make use of a function // the function is called **using its identifier** MP2 Parameters are / may be passed (from the main program) to the function (to be used within the function) MP3 The function performs its task … MP4 … and returns a value / values to the main program	**3**

Question	Answer	Marks
4(a)	**One** mark per mark point, max **two** • To ensure that data has been accurately copied // to ensure that changes have not been made to the values originally intended when data is copied • … from one source to another	**2**

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks
4(b)	**One** mark for each appropriate verification check, max **two** **One** mark for each correct accompanying use, max **two** **For example:** Verification check 1 – Visual check Use – the user looks through the data that has been entered and confirms that no changes have been made. Verification check 2 – Double data entry Use – data is entered twice, the two entries are compared and if they do not match, a re-entry is requested.	4

Question	Answer	Marks
5(a)	**One** mark for each correct line. **Description** to check that the data entered is an integer to check that some data has been entered to check that the data entered has an appropriate number of characters to check that an identification number contains no errors **Check** check digit format check length check presence check type check	4

Question	Answer	Marks

5(b)	**One** mark per mark point, max **three**
	• appropriate `REPEAT` / `WHILE` loop begin and end
	• input of `Length`
	• appropriate input prompt / error message
	• correct loop exit/entry condition / selection
	
	Example answers:
	
	WHILE Loop
	
	`OUTPUT "Enter a number between 15 and 35 inclusive"`
	`INPUT Length`
	`WHILE Length <15 OR Length > 35 (DO)`
	`  OUTPUT "Your number must be between 15 and 35 inclusive`
	`  INPUT Length`
	`ENDWHILE`
	
	REPEAT Loop
	
	`REPEAT`
	`  OUTPUT "Enter a number between 15 and 35 inclusive"`
	`  INPUT Length`
	`UNTIL Length >= 15 AND LENGTH <= 35`
		3

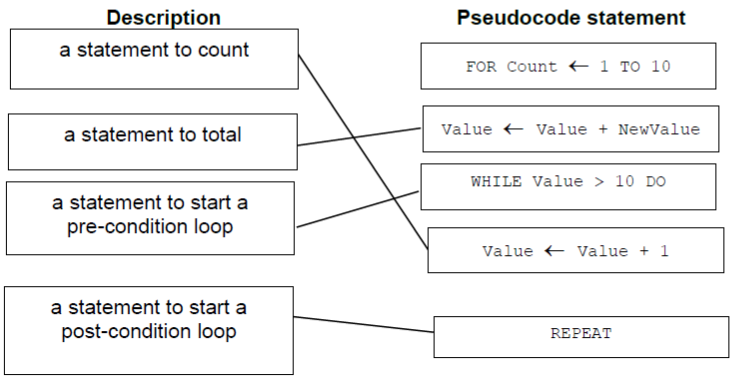
Question	Answer	Marks
9(a)	`DECLARE Saying : STRING`	1

8 Algorithms & Programming
ANSWERS

Question	Answer	Marks

9(b)	**One** mark per mark point, max **five**
	MP1 input a string into `Saying`
	MP2 correct use of `OPENFILE` to write data
	MP3 correct use of `WRITEFILE` to write `Saying`
	MP4 correct use of `CLOSEFILE`
	MP5 correct use of filename `Quotations.txt` throughout
	
	For example:
	
	`INPUT Saying`
	`OPENFILE "Quotations.txt" FOR WRITE`
	`WRITEFILE "Quotations.txt", Saying`
	`CLOSEFILE "Quotations.txt"`
		5

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

8 Algorithms & Programming
ANSWERS

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

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