Cambridge Assessment International Education

Example Candidate Responses – Paper 2

Cambridge International AS & A Level Computer Science 9618

For examination from 2021

Cambridge Pathway

© Cambridge University Press & Assessment 2022 v1

Cambridge Assessment International Education is part of Cambridge University Press & Assessment. Cambridge University Press & Assessment is a department of the University of Cambridge.

Cambridge University Press & Assessment retains the copyright on all its publications. Registered centres are permitted to copy material from this booklet for their own internal use. However, we cannot give permission to centres to photocopy any material that is acknowledged to a third party even for internal use within a centre.

Contents

Introduction

The main aim of this booklet is to exemplify standards for those teaching Cambridge International AS & A Level Computer Science 9618, and to show how different levels of candidates' performance (high, middle and low) relate to the subject's curriculum and assessment objectives.

In this booklet candidate responses have been chosen from the June 2021 exam series to exemplify a range of answers.

For each question, the response is annotated with a clear explanation of where and why marks were awarded or omitted. This is followed by examiner comments on how the answer could have been improved. In this way, it is possible for you to understand what candidates have done to gain their marks and what they could do to improve their answers. There is also a list of common mistakes candidates made in their answers for each question.

This document provides illustrative examples of candidate work with examiner commentary. These help teachers to assess the standard required to achieve marks beyond the guidance of the mark scheme. Therefore, in some circumstances, such as where exact answers are required, there will not be much comment.

The questions, mark schemes and inserts used here are available to download from the School Support Hub. These files are:

9618 June 2021 Question Paper 22

9618 June 2021 Mark Scheme 22

Past exam resources and other teaching and learning resources are available on the School Support Hub:

www.cambridgeinternational.org/support

How to use this booklet

This booklet goes through the paper one question at a time, showing you the high-, middle- and low-level response for each question. The candidate answers are set in a table. In the left-hand column are the candidate answers, and in the right-hand column are the Examiner comments.

Example Candidate Response – high	Examiner comments

5 (a) A student is learning about arrays.

She wants to write a program to:

- declare a 1D array RNum of 100 elements of type INTEGER
- assign each element a random value in the range 1 to 200 inclusive
- count and output how many numbers generated were between 66 and 173 inclusive.

(i) Write pseudocode to represent the algorithm.

```
DECLARE RNum:ARRAY[1:100] OF INTEGER
DECLARE i,j, Count : INTEGER
FOR i ← 1 TO 100
RNum[i] ← INT(RAND(200))
NEXT i
```

1 The candidate is awarded marking point 1 for the array declaration.

2 The candidate is awarded marking point 3 for the correct array 'syntax'

Answers are by real candidates in exam conditions. These show you the types of answers for each level. Discuss and analyse the answers with your learners in the classroom to improve their skills.

Examiner comments are alongside the answers. These explain where and why marks were awarded. This helps you to interpret the standard of Cambridge exams so you can help your learners to refine their exam technique.

How the candidate could have improved their answer

- **(a)(i)** To be awarded marking point 4, the candidate needed to use the correct expression for generating the required random number. The statement `INT(RAND(200))` would generate a value in the range 0 to 199 rather than one in the range 1 to 200 inclusive, as required.
- **(a)(ii)** To be awarded marking point 6, the candidate needed to add more detail to their comment 'the following element', which was too vague. They could have put 'the next empty location' instead, which would have been acceptable.

This section explains how the candidate could have improved each answer. This helps you to interpret the standard of Cambridge exams and helps your learners to refine their exam technique.

Common mistakes candidates made in this question

- **(a)(i)** Some candidates made little or no attempt to answer this question.
- **(a)(i)** Few candidates used the `RAND()` function. Candidates needed to refer to the index provided for a description of pseudocode functions and operators and these definitions needed to be followed.
- **(a)(i)** Many candidates shortened the conditional expression `IF RNum >= 66 AND RNum <= 173` to the unacceptable form `IF RNum >= 66 AND <= 173`.

Often candidates were not awarded marks because they misread or misinterpreted the questions.

Lists the common mistakes candidates made in answering each question. This will help your learners to avoid these mistakes and give them the best chance of achieving the available marks.

Question 1

Example Candidate Response – high

Refer to the **insert** for the list of pseudocode functions and operators.

1 **(a)** **(i)** Complete the following table by giving the appropriate data type in each case.

Variable	Example data value	Data type
Name	"Catherine"	String
Index	100	Integer
Modified	FALSE	Boolean
Holiday	25/12/2020	Date

1

[4]

(ii) Evaluate each expression in the following table by using the initial data values shown in **part (a)(i).**

Expression	Evaluates to
Modified OR Index > 100	FALSE
LENGTH("Student: " & Name)	18
INT(Index + 2.9)	102
MID(Name, 1, 3)	"Cat"

102.9

2

[4]

(b) Each pseudocode statement in the following table contains an example of selection, assignment or iteration.

Put **one** tick ('✓') in the appropriate column for each statement.

Statement	Selection	Assignment	Iteration
Index ← Index + 1		✓	
IF Modified = TRUE THEN	✓		
ENDWHILE			✓

3

[3]

Examiner comments

1 All rows are correct.

Mark for (a)(i) = 4 out of 4

2 All rows are correct.

Mark for (a)(ii) = 4 out of 4

3 All rows are correct.

Mark for (b) = 3 out of 3

Total mark awarded = 11 out of 11

Example Candidate Response – middle

Refer to the **insert** for the list of pseudocode functions and operators.

1 (a) (i) Complete the following table by giving the appropriate data type in each case.

Variable	Example data value	Data type
Name	"Catherine"	STRING
Index	100	INTEGER
Modified	FALSE	BOOLEAN
Holiday	25/12/2020	CHAR

[4]

(ii) Evaluate each expression in the following table by using the initial data values shown in **part (a)(i)**.

Expression	Evaluates to
Modified OR Index > 100	0
LENGTH("Student: " & Name)	18
INT(Index + 2.9)	102
MID(Name, 1, 3)	Cat

[4]

(b) Each pseudocode statement in the following table contains an example of selection, assignment or iteration.

Put **one** tick ('✓') in the appropriate column for each statement.

Statement	Selection	Assignment	Iteration
Index ← Index + 1			✓
IF Modified = TRUE THEN	✓		
ENDWHILE			✓

[3]

Examiner comments

1 The candidate completes the first 3 rows correctly, but the last row should be type `DATE`.

Mark for (a)(i) = 3 out of 4

2 The candidate completes rows 2 and 3 correctly. In row 1, zero is not acceptable as a logical value. The answer in row 4 is of type `STRING` so it should be enclosed in quotation marks.

Mark for (a)(ii) = 2 out of 4

3 The candidate completes rows 2 and 3 correctly. Row 1 is an Assignment statement.

Mark for (b) = 2 out of 3

Total mark awarded = 7 out of 11

Example Candidate Response – low

Refer to the **insert** for the list of pseudocode functions and operators.

1 (a) (i) Complete the following table by giving the appropriate data type in each case.

Variable	Example data value	Data type
Name	"Catherine"	TEXT
Index	100	Number
Modified	FALSE	Boolean
Holiday	25/12/2020	date

1

[4]

(ii) Evaluate each expression in the following table by using the initial data values shown in **part (a)(i)**.

Expression	Evaluates to
Modified OR Index > 100	String
LENGTH("Student: " & Name)	String
INT(Index + 2.9)	string
MID(Name, 1, 3)	String

2

[4]

(b) Each pseudocode statement in the following table contains an example of selection, assignment or iteration.

Put **one** tick ('✓') in the appropriate column for each statement.

Statement	Selection	Assignment	Iteration
Index ← Index + 1	✓		
IF Modified = TRUE THEN		✓	
ENDWHILE			✓

3

[3]

Examiner comments

1 The candidate completes rows 3 and 4 correctly. Row 1 should be of type `STRING` and row 2 of type `INTEGER`.

Mark for (a)(i) = 2 out of 4

2 The candidate completes all rows incorrectly. They give a data type rather than an evaluated result.

Mark for (a)(ii) = 0 out of 4

3 The candidate only completes row 3 correctly. Row 1 is an assignment statement and row 2 a selection statement.

Mark for (b) = 1 out of 3

Total mark awarded = 3 out of 11

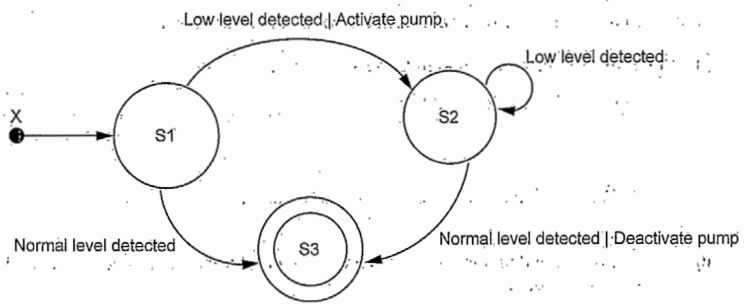
Common mistakes candidates made in this question

- Some candidates did not use `TRUE` or `FALSE` to denote logical values.
- Some candidates did not use quotation marks to denote values of data types `CHAR` or `STRING`.

Question 2

Example Candidate Response – high	Examiner comments

2 (a) Examine the following state-transition diagram.



(i) Complete the table with reference to the diagram.

	Answer
The number of transitions that result in a different state	3
The number of transitions with associated outputs	2
The label that should replace 'X'	Start pointer
The final or halting state	S3

[4]

(ii) The current state is S1. The following inputs occur.

1. Low level detected
2. Low level detected
3. Low level detected
4. Low level detected

Give the number of outputs and the current state.

Number of outputs ..... 3 .....

Current state ..... S3 .....

[2]

1 Rows 1, 2 and 4 are correct. Row 3 is not the correct term.

Mark for (a)(i) = 3 out of 4

2 Both of the candidate's answers are incorrect.

Mark for (a)(ii) = 0 out of 2

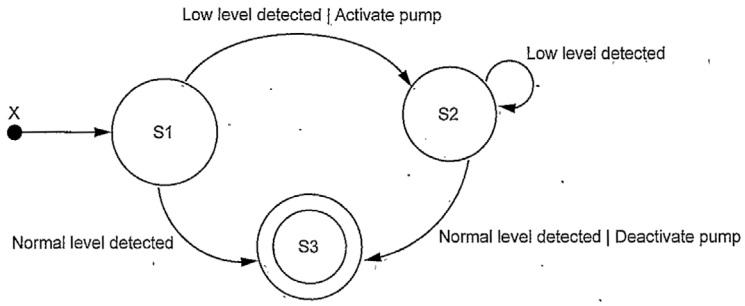
Example Candidate Response – high, continued	Examiner comments
(b) A system is being developed to help manage book loans in a library. Registered users may borrow books from the library for a period of time. **(i)** State **three** items of data that must be stored for each loan. 1 ~~name~~ Student ID 2 bookID 3 3 borrowdate [2] **(ii)** State **one** item of data that will be required in the library system but does not need to be stored for each loan. ~~Status~~ Studentname 4 [1] **(iii)** One operation that manipulates the data stored for each loan, would produce a list of all overdue books. Identify **two other** operations. Operation 1 produce list of ~~Student~~ names of Students currently borrowing a book. Operation 2 produce a list which shows how many of each book is present in the library 5 [2]	3 All parts of the candidate's answer are correct. Mark for (b)(i) = 2 out of 2 4 The candidate gives a correct answer. Mark for (b)(ii) = 1 out of 1 5 Both of the candidate's answers are correct. Mark for (b)(iii) = 2 out of 2 **Total mark awarded = 8 out of 11**

How the candidate could have improved their answer

(a)(i) The candidate needed to use the correct term for the label, which was 'Start'. The term 'pointer' had a special meaning and in this case the mark scheme was strictly applied.

Example Candidate Response – middle

2 (a) Examine the following state-transition diagram.



(i) Complete the table with reference to the diagram.

	Answer
The number of transitions that result in a different state	1
The number of transitions with associated outputs	2
The label that should replace 'X'	
The final or halting state	S_3

1

[4]

(ii) The current state is S1. The following inputs occur.

1. Low level detected
2. Low level detected
3. Low level detected
4. Low level detected

Give the number of outputs and the current state.

Number of outputs ..... 4 .....

Current state ..... S_3 .....

2

[2]

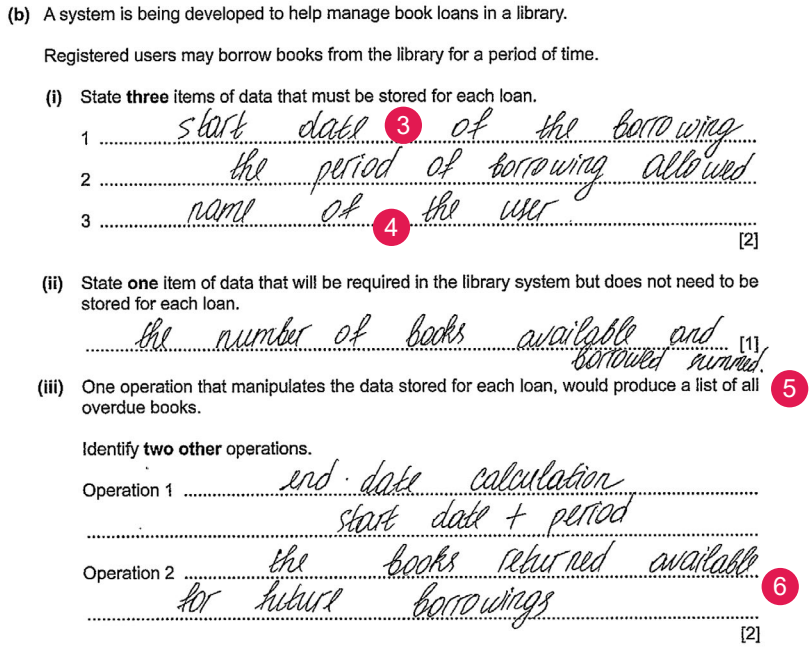
Examiner comments

1 The candidate completes rows 2 and 4 correctly.

Mark for (a)(i) = 2 out of 4

2 Both of the candidate's answers are incorrect.

Mark for (a)(ii) = 0 out of 2

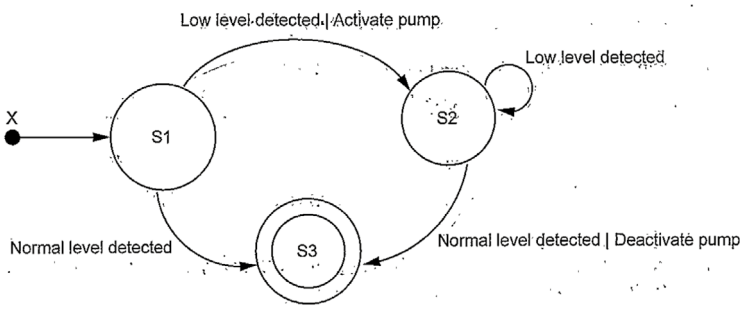
Example Candidate Response – middle, continued	Examiner comments
	3 The candidate's answer 'Start date' is correct. 4 The 'name of the user' may not be unique and so should not be used. Mark for (b)(i) = 1 out of 2 5 These items could reasonably be calculated when needed. The values would not need to be stored. Mark for (b)(ii) = 0 out of 1 6 The candidate is awarded both marks for these answers. Mark for (b)(iii) = 2 out of 2 **Total mark awarded = 5 out of 11**

How the candidate could have improved their answer

- **(b)(i)** The candidate needed to use a unique data item such as User ID (or similar) to identify a loan as many users may have shared the same name.
- **(b)(iii)** Best practice requires an item that could **not** be calculated from already stored data, such as is the case for the two answers given.

Example Candidate Response – low

2 (a) Examine the following state-transition diagram.



(i) Complete the table with reference to the diagram.

	Answer
The number of transitions that result in a different state	2
The number of transitions with associated outputs	3
The label that should replace 'X'	input
The final or halting state	S3

1

[4]

(ii) The current state is S1. The following inputs occur.

1. Low level detected
2. Low level detected
3. Low level detected
4. Low level detected

Give the number of outputs and the current state.

Number of outputs ..... 4 .....

Current state ..... S2 .....

2

[2]

Examiner comments

1 Only row 4 is correct.

Mark for (a)(i) = 1 out of 4

2 The candidate's answer for 'Current state' is correct.

Mark for (a)(ii) = 1 out of 2

Example Candidate Response – low, continued

(b) A system is being developed to help manage book loans in a library.

Registered users may borrow books from the library for a period of time.

(i) State **three** items of data that must be stored for each loan.

1 date on which the book was borrowed 3

2 period of time of which book has to be returned after it was borrowed

3 if the book was returned or not

[2]

(ii) State **one** item of data that will be required in the library system but does not need to be stored for each loan.

ID number of the book 4 [1]

(iii) One operation that manipulates the data stored for each loan, would produce a list of all overdue books.

Identify **two other** operations.

Operation 1 that is responsible for memory managment will optimise data storage for the books

Operation 2 sect that is responsible for security of the system (authorised access to the information) 5

[2]

or

1) operation that will make a list of most popular books

2) 6

Examiner comments

3 The candidate is awarded a mark for the first item. The second item would not need to be stored for each loan.

Mark for (b)(i) = 1 out of 2

4 The candidate gives an incorrect answer. Data is required for a loan.

Mark for (b)(ii) = 0 out of 1

5 The first two items the candidate gives do not relate to the library loan process.

6 The alternative answer the candidate gives could be awarded a mark, but the first two answers given are the only ones that can be considered when awarding marks in this style of question.

Mark for (b)(iii) = 0 out of 2

Total mark awarded = 3 out of 11

How the candidate could have improved their answer

- **(b)(i)** The mark scheme did not consider the period of the loan (the second item given) as something that would need to be recorded for each loan. The candidate could have improved their answer for the third item if they had given 'date the book was returned'. If they had done this, then the candidate could have given 'Book ID' as the second item and been awarded the second mark.
- **(b)(ii)** The ID number of the book was something that would need to be stored so that the actual copy could be identified. The mark scheme gives examples of items which could be required, but which are not required to be stored for each loan.

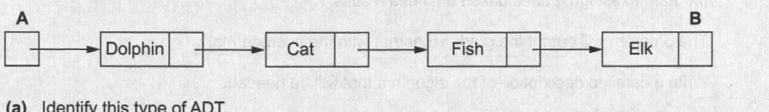
Common mistakes candidates made in this question

- **(b)(i)** Some candidates used 'Name of the user', which was too vague as it was possible for more than one user to have the same name. Candidates needed to use a unique item such as 'User ID' to identify the borrower.
- **(b)(ii)** Some candidates incorrectly stated that the library system would store items of data whose values could reasonably be calculated from already-stored data.

Question 3

Example Candidate Response – high

3 The following diagram represents an Abstract Data Type (ADT).



(a) Identify this type of ADT.

Linked lists 1 [1]

(b) Give the technical term for the item labelled **A** in the diagram.

Start Pointer 2 [1]

(c) Give the technical term for the item labelled **B** in the diagram.

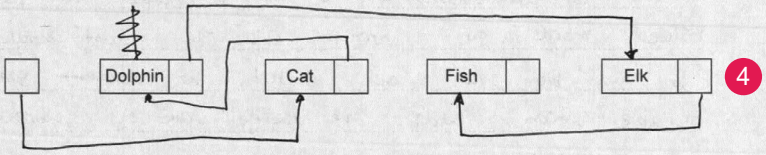
Explain the meaning of the value given to this item.

Term Null ~~Node~~ pointer

Meaning does not link to any other nodes

3 [2]

(d) Complete the diagram to show the ADT after the data has been sorted in alphabetical order.



4 [2]

Examiner comments

1 The candidate gives a correct answer.

Mark for (a) = 1 out of 1

2 The candidate gives a correct answer.

Mark for (b) = 1 out of 1

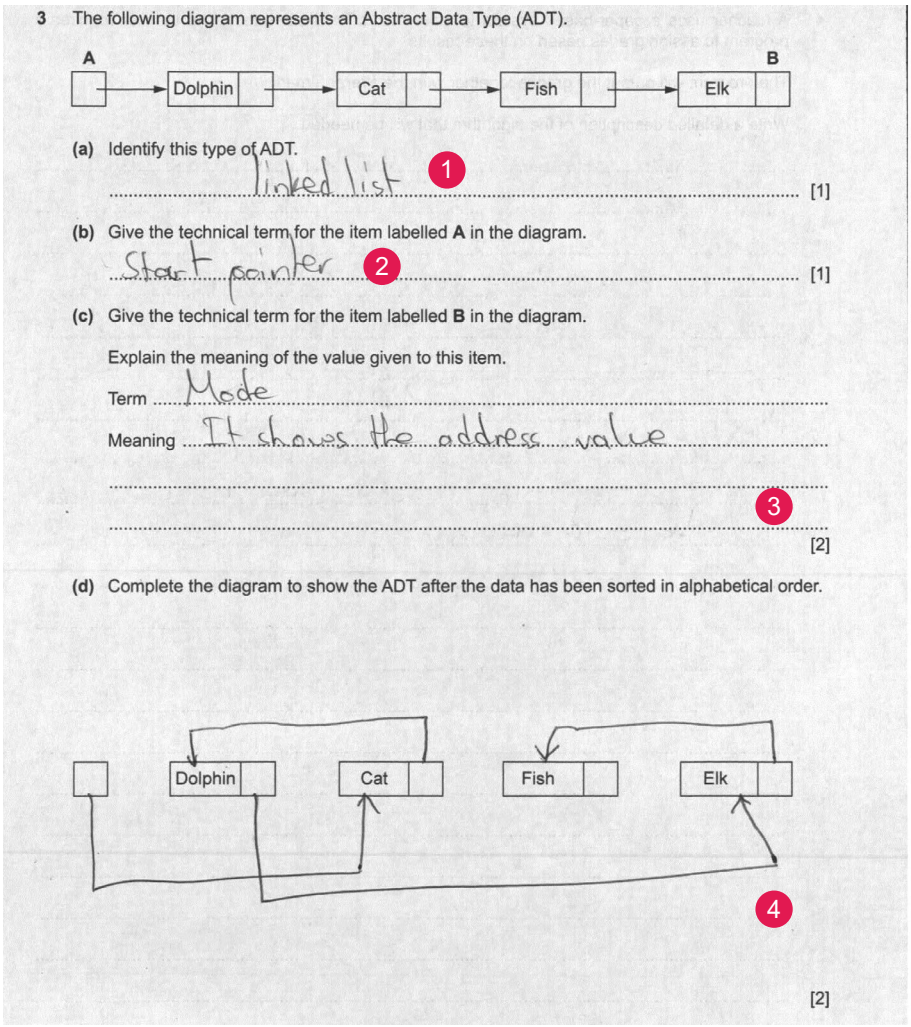
3 Both answers are correct.

Mark for (c) = 2 out of 2

4 The candidate completes the diagram correctly with arrows from each 'pointer' to the next node.

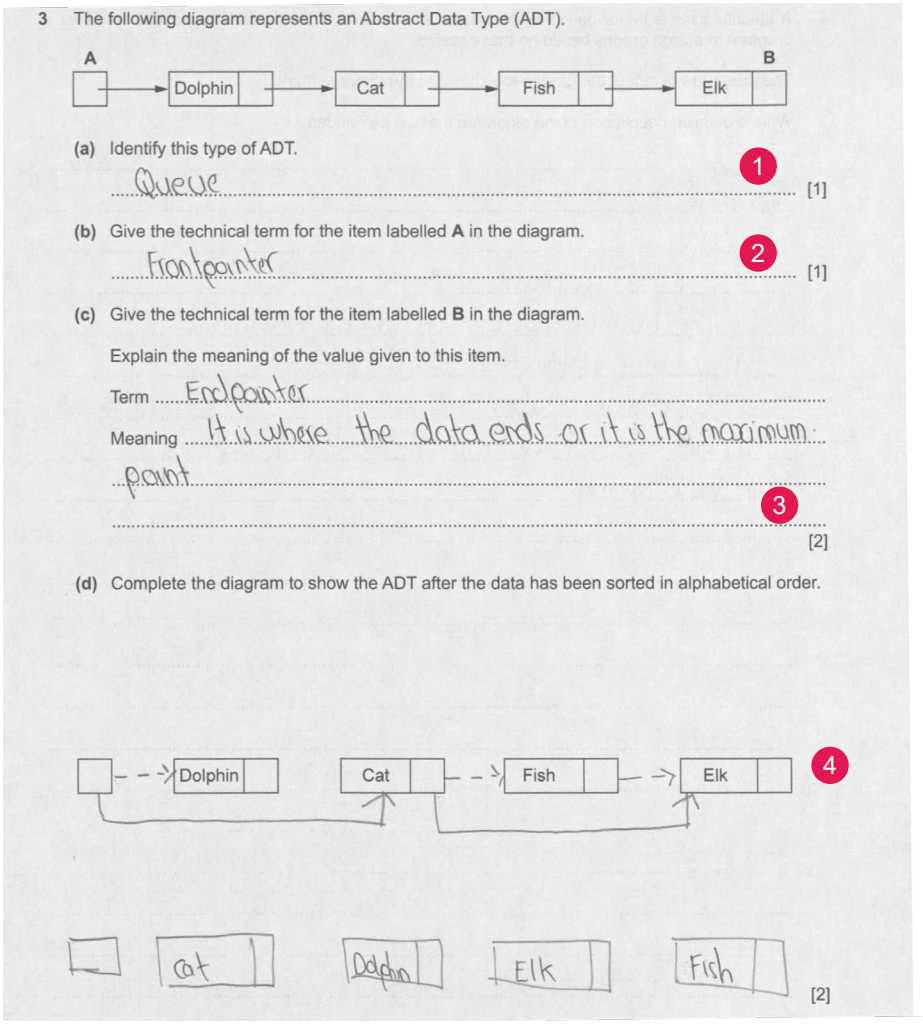
Mark for (d) = 2 out of 2

Total mark awarded = 6 out of 6

Example Candidate Response – middle	Examiner comments
	1 The candidate gives the correct answer. Mark for (a) = 1 out of 1 2 The candidate gives the correct answer. Mark for (b) = 1 out of 1 3 The candidate gives two incorrect answers. Mark for (c) = 0 out of 2 4 The candidate completes the diagram correctly with arrows from each 'pointer' to the next node. Mark for (d) = 2 out of 2 **Total mark awarded = 4 out of 6**

How the candidate could have improved their answer

(c) The candidate needed to use the term 'Null Pointer'. The answer the candidate gave for 'Meaning' suggested that they had an understanding of the use of pointers, but the particular meaning in the case of the Null Pointer would be that there were no more nodes in the list.

Example Candidate Response – low	Examiner comments
	1 The candidate gives an incorrect answer. Mark for (a) = 0 out of 1 2 The candidate gives another incorrect answer. Mark for (b) = 0 out of 1 3 The candidate uses an incorrect term and the phrase 'where the data ends' is too imprecise. Mark for (c) = 0 out of 2 4 The candidate is awarded the first mark for the arrow linking the Start Pointer to the node with 'Cat'. The remaining annotations are incorrect. Mark for (d) = 1 out of 2 **Total mark awarded = 1 out of 6**

How the candidate could have improved their answer

(c) The candidate needed to use the term 'Null Pointer'. The answer the candidate gave for 'Meaning' suggested an understanding of the linked list operation, but the expression 'where the data ends' was not sufficiently succinct.

Common mistakes candidates made in this question

Many candidates did not attempt parts of this question.

Question 4

Example Candidate Response – high

4 A teacher uses a paper-based system to store marks for a class test. The teacher requires a program to assign grades based on these results.

The program will output the grades together with the average mark.

Write a detailed description of the algorithm that will be needed.

1. The teacher has to create a global 1D Array MarkToGrade to store the maximum mark for each grade starting from the lowest. (one mark per array position) 1
2. INPUT student mark to StudentMarkTemp
3. ADD student mark to a totaling variable (ClassTotal) 2
4. Compare StudentMark with StudentMarkTemp
Check in a loop whether StudentMarkTem is lower or equal and if not increment array position until condition is met, storing grade 3
in an array of positions equal to STUDENTMARK
the students (to the corresponding student position) for an amount of
5. REPEAT steps 2 - 4 4
lines that is equal to the students
6. Divide ClassTotal by the amount of students and assign to AVGMark
7. In a loop output all array elements 5
of STUDENTMark and AVGMark along with it UNTIL the end of the array. 6
7

[6]

Examiner comments

1 The candidate is awarded marking point 3 for inputting a student mark within a loop (which the candidate describes in step 5).

2 The candidate is awarded marking point 6 because they add the mark to the total.

3 The candidate is awarded marking point 4 for their description of how each mark compares with grade thresholds.

4 The candidate refers to a loop so is awarded marking point 2.

5 The candidate is not awarded marking point 5 because the description outputs the mark, but not the grade.

6 The candidate is awarded marking point 7 for the output of the average, as they previously describe the calculation.

7 The candidate makes no reference to a variable used to store the count of students in the class so they are not awarded marking point 1.

Total mark awarded = 5 out of 6

How the candidate could have improved their answer

- To be awarded marking point 1, the candidate could have made a specific reference to the use of variables to store the number of students and the total of all the marks.
- To be awarded marking point 2, the candidate could have given a description of a loop used to input all of the marks.
- To be awarded marking point 5, the candidate needed to output each student grade.

Example Candidate Response – middle

4 A teacher uses a paper-based system to store marks for a class test. The teacher requires a program to assign grades based on these results.

The program will output the grades together with the average mark.

Write a detailed description of the algorithm that will be needed.

~~For loop to~~ 1
a count controlled loop with a count that is assigned
by the user is used. 2
As the count-controlled loop runs, the teacher inputs the marks
3 for the test, ~~the~~ The program stores the input value
into an array that goes through ~~the~~ the address locations
specified by the count. ~~Then the input~~ ~~sum variable~~
Then, the input value is added on to the sum
4 variable which is a variable with a real datatype.
After the count controlled loop is ran and all the marks
are inputted to the system, the ~~user~~ program
divides the sum by the ~~number of~~ count
given initially. This value is the average mark.
Another ~~for~~ count controlled loop is ran to output all the
5 stored marks in the array. Once the count-controlled
loop has been ran and exited, the ~~average~~ sum
variable which holds the average value of marks
is outputted. 6

[6]

Examiner comments

1 The candidate is not awarded marking point 1. Their reference to a variable to store the number of students ('a count') is vague and they do not mention a variable used to store the total.

2 The candidate mentions a loop so is awarded marking point 2.

3 The candidate is awarded marking point 3 awarded for the input of marks within a loop.

4 The candidate is awarded marking point 6 for adding the input value to a total.

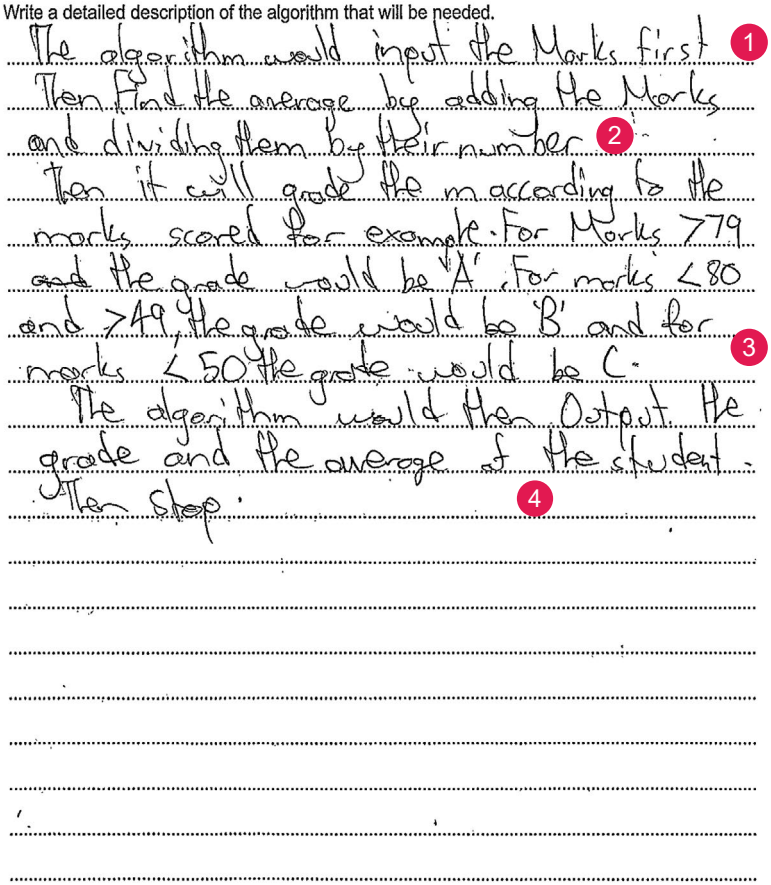
5 The candidate's solution needs to output the grade, but here it is the mark that is output, so they are not awarded marking point 5.

6 The candidate is awarded marking point 7 for the output of the average mark (they already describe the calculation of the average).

Total mark awarded = 4 out of 6

How the candidate could have improved their answer

- To be awarded marking point 1, the candidate could have made a specific reference to the use of variables to store the number of students and the total of all the marks.
- To be awarded marking point 4, the candidate could have given a description of how each mark was compared with threshold values to obtain the corresponding grade.
- To be awarded marking point 5, the candidate needed to output each student grade.

Example Candidate Response – low	Examiner comments
4 A teacher uses a paper-based system to store marks for a class test. The teacher requires a program to assign grades based on these results. The program will output the grades together with the average mark. Write a detailed description of the algorithm that will be needed.  The algorithm would input the Marks first 1 Then find the average by adding the Marks and dividing them by their number 2 Then it will grade the marks according to the marks scored for example. For Marks >79 and the grade would be 'A'. For marks <80 and >49 the grade would be 'B' and for marks <50 the grade would be C. 3 The algorithm would then Output the grade and the average of the student. Then Stop. 4	1 The candidate is not awarded marking point 3 despite this reference to marks being input because their description of the algorithm does not refer to a loop to input each mark in turn. 2 The candidate is not awarded marking point 6. The phrase 'adding the marks' is too vague and they make no specific reference to the required variables. 3 The candidate is awarded marking point 4 for their description of the mark being compared with threshold values. 4 The candidate is awarded marking point 7 because they refer to the output of the average mark. **Total mark awarded = 2 out of 6**

How the candidate could have improved their answer

- To be awarded marking point 1, the candidate could have made a specific reference to the use of variables to store the number of students and the total of all the marks.
- To be awarded marking point 2, the candidate could have referred to a loop containing the steps required for each student mark.
- To be awarded marking point 3, the candidate needed to refer to inputting the marks if there had also been a reference to a loop in the algorithm.
- The candidate could have been awarded marking point 5 if they had output each student grade.
- To be awarded marking point 6, the candidate could have made a specific reference to maintaining the total mark and the count of the number of students.

Common mistakes candidates made in this question

- Many candidates struggled with questions which asked for a description of an algorithm. Their descriptions tended to lack detail and in many cases just repeated phrases from the question.
- It might be helpful for candidates to imagine they were describing the algorithm to a fellow student, who would then use the description to produce a solution in pseudocode.
- Many candidates referred to features that were not required by the question, such as storing all names and test results in an array, converting each test mark into a percentage or sorting the marks into ascending order.

Question 5

Example Candidate Response – high

5 (a) A student is learning about arrays.

She wants to write a program to:

- declare a 1D array RNum of 100 elements of type INTEGER
- assign each element a random value in the range 1 to 200 inclusive
- count and output how many numbers generated were between 66 and 173 inclusive.

(i) Write pseudocode to represent the algorithm.

```
DECLARE RNum:ARRAY[1:100] OF INTEGER
DECLARE i, j, Count : INTEGER
FOR i ← 1 TO 100
    RNum[i] ← INT(RAND(200))
NEXT i

Count ← 0
FOR j ← 1 TO 100
    IF RNum[j] >= 66 AND RNum[j] <= 173 THEN
        Count ← Count + 1
    ENDIF
NEXT j
OUTPUT "The numbers between 66 and 173: " & Count
```

[6]

(ii) The student decides to modify the algorithm so that each element of the array will contain a unique value.

Describe the changes that the student needs to make to the algorithm.

When generating a random number, instead of directly assigning it to an element in the array, in a post-conditional loop, each of the assigned elements are compared to this new random number generated. If this element is unique, a boolean flag is changed and the program exits another post-conditional loop. Finally, this number is assigned to the following empty element in the array.

[3]

Examiner comments

1 The candidate is awarded marking point 1 for the array declaration.

2 The candidate is awarded marking point 3 for the correct array 'syntax'.

3 The candidate is not awarded marking point 4.

4 The candidate is awarded marking point 2 for the loop.

5 The candidate is awarded marking point 5 for the range check and subsequent increment of count variable.

6 The candidate is awarded marking point 6 for the output statement.

Mark for (a)(i) = 5 out of 6

7 The candidate is awarded marking point 2 because they refer to a conditional loop.

8 The candidate is awarded marking point 4 for their comparison of the new random number with an element from the array.

9 The candidate's reference to 'the following element' is vague so they are not awarded marking point 6.

Mark for (a)(ii) = 2 out of 3

Example Candidate Response – high, continued

(b) The following is a pseudocode function.

Line numbers are given for reference only.

```
01  FUNCTION StringClean(InString : STRING) RETURNS STRING
02
03     DECLARE NextChar : CHAR
04     DECLARE OutString : STRING
05     DECLARE Counter : INTEGER
06
07     OutString ← ""
08
09     FOR Counter ← 1 TO LENGTH(InString)
10        NextChar ← MID(InString, Counter, 1)
11        NextChar ← LCASE(NextChar)
12        IF NOT((NextChar < 'a') OR (NextChar > 'z')) THEN
13           OutString ← OutString & NextChar
14        ENDIF
15     NEXT Counter
16
17     RETURN OutString
18
19  ENDFUNCTION
```

(i) Examine the pseudocode and complete the following table.

	Answer
Give a line number containing an example of an initialisation statement.	07
Give a line number containing the start of a repeating block of code.	09
Give a line number containing a logic operation.	12
Give the number of parameters to the function `MID()`.	3

10

[4]

(ii) Write a simplified version of the statement in line 12.

IF NextChar >= 'a' OR NextChar <= 'z' THEN 11

[2]

Examiner comments

10 The candidate completed all rows correctly.

Mark for (b)(i) = 4 out of 4

11 Both of the candidate's comparisons are correct, but the logical operator is incorrect.

Mark for (b)(ii) = 1 out of 2

Total mark awarded = 12 out of 15

How the candidate could have improved their answer

- **(a)(i)** To be awarded marking point 4, the candidate needed to use the correct expression for generating the required random number. The statement `INT(RAND(200))` would generate a value in the range 0 to 199 rather than one in the range 1 to 200 inclusive, as required.
- **(a)(ii)** To be awarded marking point 6, the candidate needed to add more detail to their comment 'the following element', which was too vague. They could have put 'the next empty location' instead, which would have been acceptable.
- **(a)(ii)** The candidate needed to address marking point 1 and marking point 5.
- **(b)(ii)** The candidate needed to use the logical operator `AND` rather than `OR`.

Example Candidate Response – middle

5 (a) A student is learning about arrays.

She wants to write a program to:

- declare a 1D array `RNum` of 100 elements of type `INTEGER`
- assign each element a random value in the range 1 to 200 inclusive
- count and output how many numbers generated were between 66 and 173 inclusive.

(i) Write pseudocode to represent the algorithm.

```
1) DECLARE RNum : INTEGER (*) Count ← 0
3) RAND(201)
2) FOR Index ← 1 TO 100
4)     IF RAND(201)=INTEGER AND
       NOT RAND(201) =0 THEN
5)         RNum [Index] ← RAND(201)
6)         Index ← Index+1
7)         IF RAND(201) 765 AND
           RAND(201) < 174 THEN
8)             Count ← Count +1
9) ENDFOR (*4) ELSE RAND(201)
10) OUTPUT Count
```

[6]

(ii) The student decides to modify the algorithm so that each element of the array will contain a unique value.

Describe the changes that the student needs to make to the algorithm.

```
in line four
IF RAND(201)= INTEGER AND NOT RAND(201)=0
AND NOT RAND(201) = RNum [Index]
```

[3]

Examiner comments

1 The candidate is not awarded marking point 1 as this is not an acceptable array declaration.

2 The candidate is awarded marking point 3 for the correct array 'syntax'.

3 The candidate is not awarded marking point 4 as the statement using the `RAND()` function would not give the range of values required.

4 The candidate is awarded marking point 5 for the correct range check and subsequent increment of variable 'Count'.

5 The candidate is awarded marking point 2 for the loop structure.

6 The candidate is awarded marking point 6 for the final output statement.

Mark for (a)(i) = 4 out of 6

7 The candidate does not address the question.

Mark for (a)(ii) = 0 out of 3

Example Candidate Response – middle, continued

(b) The following is a pseudocode function.

Line numbers are given for reference only.

```
01  FUNCTION StringClean(InString : STRING) RETURNS STRING
02
03      DECLARE NextChar : CHAR
04      DECLARE OutString : STRING
05      DECLARE Counter : INTEGER
06
07      OutString ← ""
08
09      FOR Counter ← 1 TO LENGTH(InString)
10          NextChar ← MID(InString, Counter, 1)
11          NextChar ← LCASE(NextChar)
12          IF NOT((NextChar < 'a') OR (NextChar > 'z')) THEN
13              OutString ← OutString & NextChar
14          ENDIF
15      NEXT Counter
16
17      RETURN OutString
18
19  ENDFUNCTION
```

(i) Examine the pseudocode and complete the following table.

	Answer
Give a line number containing an example of an initialisation statement.	07
Give a line number containing the start of a repeating block of code.	09
Give a line number containing a logic operation.	12
Give the number of parameters to the function `MID()`.	2

8

[4]

(ii) Write a simplified version of the statement in line 12.

IF NextChar > 'a' AND NextChar < 'z' THEN 9

[2]

Examiner comments

8 The candidate completes the first 3 rows correctly, but the last row is incorrect.

Mark for (b)(i) = 3 out of 4

9 The candidate's use of the logical `AND` operator is correct, but their comparisons are incorrect.

Mark for (b)(ii) = 1 out of 2

Total mark awarded = 8 out of 15

How the candidate could have improved their answer

- **(a)(i)** To be awarded marking point 1, the candidate needed to specify that the array contained 100 elements in their declaration statement. They gave the name of the array and the data type correctly but the asterisk '(*)' was not acceptable as a boundary description.
- **(a)(i)** To be awarded marking point 4, the candidate needed to use the correct expression for generating the required random number. The statement `RAND(201)` would generate a real value in the range 0 to 200.999 (recurring) rather than an integer value in the range 1 to 200 inclusive, as required.
- The statement `IF RAND(201) = INTEGER` was not a valid way of testing whether a number was an integer. The candidate needed to use the `INT()` function to convert from a `REAL` to an `INTEGER`.
- **(a)(ii)** The candidate's description needed to explain how the algorithm would ensure that each array element was unique. This would involve the use of a loop containing a test to compare each new random number with all those values already stored in the array.
- **(b)(ii)** The candidate needed to use the comparison operators >= and <= respectively to be awarded the second mark.

Example Candidate Response – low

5 (a) A student is learning about arrays.

She wants to write a program to:

- declare a 1D array `RNum` of 100 elements of type `INTEGER`
- assign each element a random value in the range 1 to 200 inclusive
- count and output how many numbers generated were between 66 and 173 inclusive.

(i) Write pseudocode to represent the algorithm.

```
DECLARE RNum Array [0:99] : INTEGER   (1)
REPDECLARE Count
DECLARE ~~Input~~ Num : INTEGER
~~Count ← 0~~ Count ← 0
Num ← 0
REPEAT
        MID (RNum, 1, 100)
        FOR ~~Count~~ RNum ~~IN RANGE 1-200~~
(2)             INPUT ← Num IN RANGE 1 to 200
        IF ~~X Num~~ ((66 <= Num) ~~OR~~ AND (Num < 173))
            THEN                                   (3)
            Count ← Count +1
        ← NEXT Count
(4) UNTILL RNum ← 100
OUTPUT Count   (5)
```

[6]

(ii) The student decides to modify the algorithm so that each element of the array will contain a unique value.

Describe the changes that the student needs to make to the algorithm.

The student will need to add a IF statement that look for repetition and if it found a repeated value changed until it's unique. (6)

[3]

Examiner comments

1 The candidate is awarded marking point 1 for the array declaration.

2 The candidate is not awarded marking points 3 or 4 because they do not attempt to generate a random number, or to assign a value to the `RNum` array.

3 The candidate's range check is incorrect so they are not awarded marking point 5.

4 The candidate's loop termination condition is incorrect so they are not awarded marking point 2.

5 The candidate is not awarded marking point 6. Although the variable 'Count' is incremented in the loop, the candidate also uses it as the loop counter in the statement: `NEXT Count`.

Mark for (a)(i) = 1 out of 6

6 The candidate's description is too vague and repeats the question. The phrase 'a repeated value changed' is not sufficient to imply the generation of a new random number.

Mark for (a)(ii) = 0 out of 3

Example Candidate Response – low, continued

(b) The following is a pseudocode function.

Line numbers are given for reference only.

```
01  FUNCTION StringClean(InString : STRING) RETURNS STRING
02
03      DECLARE NextChar : CHAR
04      DECLARE OutString : STRING
05      DECLARE Counter : INTEGER
06
07      OutString ← ""
08
09      FOR Counter ← 1 TO LENGTH(InString)
10          NextChar ← MID(InString, Counter, 1)
11          NextChar ← LCASE(NextChar)
12          IF NOT((NextChar < 'a') OR (NextChar > 'z')) THEN
13              OutString ← OutString & NextChar
14          ENDIF
15      NEXT Counter
16
17      RETURN OutString
18
19  ENDFUNCTION
```

(i) Examine the pseudocode and complete the following table.

	Answer
Give a line number containing an example of an initialisation statement.	03
Give a line number containing the start of a repeating block of code.	09
Give a line number containing a logic operation.	10
Give the number of parameters to the function `MID()`.	3

7

[4]

(ii) Write a simplified version of the statement in line 12.

IF NOT ("a" > NextChar < "z") THEN 8

.....

[2]

Examiner comments

7 The candidate completes rows 2 and 4 correctly, but rows 1 and 3 are incorrect.

Mark for (b)(i) = 2 out of 4

8 The candidate is awarded one mark for using `NOT`, recognising the attempt at simplification.

Mark for (b)(ii) = 1 out of 2

Total mark awarded = 4 out of 15

How the candidate could have improved their answer

- **(a)(i)** To be awarded marking point 2, the candidate's solution needed to contain a loop that repeated 100 times. The statement `UNTIL RNum ← 100` contained two errors: `RNum` was an array and not an integer and the assignment arrow needed to be an equals sign.
- **(a)(i)** To be awarded marking point 3, the candidate needed to attempt to store a value in the array.
- **(a)(i)** To be awarded marking point 4, the candidate needed to use the correct expression to generate the required random number.
- **(a)(i)** To be awarded marking point 5, the candidate needed to make sure the range comparison had been correct. The comparison statement used contained `Num < 173` which should have been `Num <= 173`.
- **(a)(i)** To be awarded marking point 6, the candidate needed to make sure that the variable `Count` had not also been referenced in the loop statement `NEXT Count`.
- **(a)(ii)** The candidate's description needed to explain how the algorithm would ensure that each array element was unique. They needed to use a loop containing a test to compare each new random number with all those values already stored in the array.
- **(b)(ii)** The candidate needed to make sure that the conditional expression was correct. In this case, the required expression would have been `NextChar < 'a' OR NextChar > 'z'`.

Common mistakes candidates made in this question

- **(a)(i)** Some candidates made little or no attempt to answer this question.
- **(a)(i)** Few candidates used the `RAND()` function. Candidates needed to refer to the index provided for a description of pseudocode functions and operators and these definitions needed to be followed.
- **(a)(i)** Many candidates shortened the conditional expression `IF RNum >= 66 AND RNum <= 173` to the unacceptable form `IF RNum >= 66 AND <= 173`.
- **(a)(ii)** Many candidates struggled with questions which asked for a description of an algorithm. Their descriptions tended to lack detail and in many cases just repeated phrases from the question.
- **(a)(ii)** It might be helpful for candidates to imagine they were describing the algorithm to a fellow student, who would then use the description to produce a solution in pseudocode.
- **(a)(ii)** Many candidates referred to features that were not required by the question.

Question 6

Example Candidate Response – high	Examiner comments

6 A procedure `CountVowels()` will:

- be called with a string containing alphanumeric characters as its parameter
- count and output the number of occurrences of each vowel (a, e, i, o, u) in the string
- count and output the number of occurrences of the other alphabetic characters (as a single total).

The string may contain both upper and lower case characters.

Each count value will be stored in a unique element of a global 1D array `CharCount` of type `INTEGER`. The array will contain six elements.

Write pseudocode for the procedure `CountVowels()`.

```
PROCEDURE CountVowels (Stringpassed: STRING)
    DECLARE i : INTEGER                                   (1)
    DECLARE tempchar : STRING                             (2)
    FOR i ← 1 TO LENGTH(Stringpassed)                     (3)
        tempchar ← MID(stringpassed, i, 1)
        CASE OF tempchar
            "a" : CharCount[1] ← CharCount[1]+1;
            "e" : CharCount[2] ← CharCount[2]+1;
            "i" : CharCount[3] ← CharCount[3]+1;
            "o" : CharCount[4] ← CharCount[4]+1;          (4)
            "u" : CharCount[5] ← CharCount[5]+1;
        OTHERWISE                                         (5)
            CharCount[6] ← CharCount[6]+1
        ENDCASE
(6)
    NEXT i
```

Examiner comments

(1) The candidate is awarded marking point 2 for their declaration of the variable used as a loop counter.

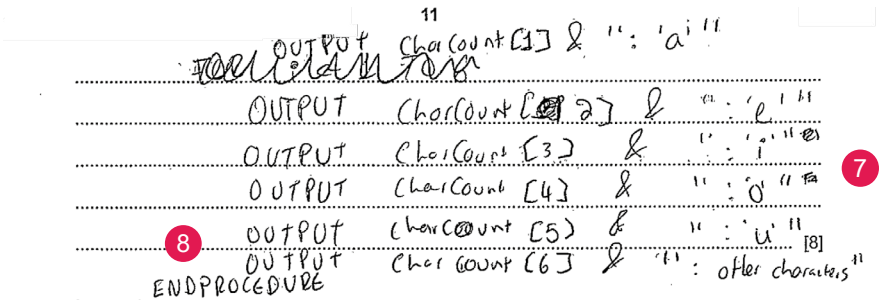
(2) The candidate is not awarded marking point 3.

(3) The candidate is awarded marking point 5 for the extraction of a character.

(4) The candidate is awarded marking point 6 for the vowel test and increment operations.

(5) The candidate is not awarded marking point 7. Their solution should count all non-vowel characters, not just alphabetic.

(6) The candidate is awarded marking point 4 for the loop.

Example Candidate Response – high, continued	Examiner comments
	7 The candidate is awarded marking point 8 for the output of the results. 8 The candidate is awarded marking point 1 for the procedure declaration. **Total mark awarded = 6 out of 8**

How the candidate could have improved their answer

- To be awarded marking point 3, the candidate needed to initialise the array elements to zero.
- To be awarded marking point 7, the candidate could have included a conditional statement to ignore non-alphabetic characters in the `OTHERWISE` clause in their solution.

Example Candidate Response – middle

6 A procedure `CountVowels()` will:

- be called with a string containing alphanumeric characters as its parameter
- count and output the number of occurrences of each vowel (a, e, i, o, u) in the string
- count and output the number of occurrences of the other alphabetic characters (as a single total).

The string may contain both upper and lower case characters.

Each count value will be stored in a unique element of a global 1D array `CharCount` of type `INTEGER`. The array will contain six elements.

Write pseudocode for the procedure `CountVowels()`.

```
PROCEDURE CountVowels(Message : STRING)                                  (1)
    DECLARE Count : INTEGER                                               (2)
    DECLARE Item : STRING                                                 (3)
    FOR Counter ← 0 TO LENGTH(CharCount)-1
        Item ← MID(Message, Counter, 1)                                   (4)
        CASE OF Item
            "a" : CharCount[0] ← CharCount[0]+1
            "e" : CharCount[1] ← CharCount[1]+1
            "i" : CharCount[2] ← CharCount[2]+1
            "o" : CharCount[3] ← CharCount[3]+1
            "u" : CharCount[4] ← CharCount[4]+1                           (5)
            OTHERWISE : CharCount[5] ← CharCount[5]+1                     (6)
        ENDCASE
(7) NEXT Counter
    OUTPUT "Letter a : " & NUM_TO_STR(CharCount[0])
    OUTPUT "Letter e : " & NUM_TO_STR(CharCount[1])
    OUTPUT "Letter i : " & NUM_TO_STR(CharCount[2])
    OUTPUT "Letter o : " & NUM_TO_STR(CharCount[3])
    OUTPUT "Letter u : " & NUM_TO_STR(CharCount[4])
    OUTPUT "Other Letters : " & NUM_TO_STR(CharCount[5])
ENDPROCEDURE                                                              (8)
```

Examiner comments

1. The candidate is awarded marking point 1 for the procedure declaration.
2. The candidate is not awarded marking point 2. They declare the identifier as `Count` but later use `Counter`
3. The candidate is not awarded marking point 3.
4. The candidate is not awarded marking point 5. They use the `MID()` function correctly, but the identifier `Counter` starts from zero, which would be an invalid value.
5. The candidate is awarded marking point 6 for the vowel test and increment operations.
6. The candidate is not awarded marking point 7 not awarded. The solution would count **all** non-vowel characters, not just alphabetic.
7. The candidate is awarded marking point 4 for the loop structure.
8. The candidate is awarded marking point 8 for the output statements.

Total mark awarded = 4 out of 8

How the candidate could have improved their answer

- To be awarded marking point 2, the candidate needed to ensure that their use of the variable used as the loop counter was consistent. The variable they used was declared as `Count` but they later referred to it as `Counter`.
- To be awarded marking point 3, the candidate needed to initialise the array elements to zero.
- To be awarded marking point 5, the candidate needed to make sure the range of values of the parameter `Count` to the `MID()` function were correct.
- To improve their answer and be awarded marking point 7, the candidate needed to make sure the solution had included a conditional statement to ignore non-alphabetic characters in the `OTHERWISE` clause.

Example Candidate Response – low

6 A procedure `CountVowels()` will:

- be called with a string containing alphanumeric characters as its parameter
- count and output the number of occurrences of each vowel (a, e, i, o, u) in the string
- count and output the number of occurrences of the other alphabetic characters (as a single total).

The string may contain both upper and lower case characters.

Each count value will be stored in a unique element of a global 1D array `CharCount` of type `INTEGER`. The array will contain six elements.

Write pseudocode for the procedure `CountVowels()`.

```
CharCount[1] ← 0
CharCount[2] ← 0
CharCount[3] ← 0
CharCount[4] ← 0
CharCount[5] ← 0
CharCount[6] ← 0
PROCEDURE CountVowels(V:STRING)
FOR w ← TO length(V)
N ← MID(V,w,4)
IF N = 'a' or N = 'A'
CharCount[1] ← CharCount[1] + 1
ELSE
IF N = 'a' or N = 'E'
THEN
CharCount[2] ← CharCount[2] + 1
ELSE IF N = 'i' OR N = 'I' THEN
CharCount[3] ← CharCount[3] + 1
ELSE IF N = 'o' OR N = 'O'
THEN CharCount[4] ← CharCount[4] + 1
ELSE IF N = 'u' or N = 'U'
THEN CharCount[5] ← CharCount[5] + 1
```

Examiner comments

1 The candidate is awarded marking point 3 for initialising the array.

2 The candidate is not awarded marking point 1 here. Their procedure heading is correct but the end statement is missing.

3 The candidate does not declare the local variables so they are not awarded marking point 2.

4 The candidate is not awarded marking point 4 because the start of the range is unspecified.

5 The candidate is awarded marking point 5 for extracting a character.

6 The candidate's comparison is incorrect so they are not awarded marking point 6.

Example Candidate Response – low, continued	Examiner comments
7 ELSE CharCount[6] ← CharCount[6] + 1 8 ENDIF [8]	7 The candidate is not awarded marking point 7. The solution would count **all** non-vowel characters, not just alphabetic. 8 The candidate does not give an output of the results so they are not awarded marking point 8. **Total mark awarded = 2 out of 8**

How the candidate could have improved their answer

- To be awarded marking point 1, the candidate needed to include the `ENDPROCEDURE` statement in their solution.
- To be awarded marking point 2, the candidate needed to declare the variable used as the loop counter.
- To be awarded marking point 4, the candidate could have included the initial value in their loop statement.
- To be awarded marking point 6, the candidate needed to make sure that their comparisons for each of the vowel characteristics were correct.
- To be awarded marking point 7, the candidate needed to include a conditional statement to ignore non-alphabetic characters within the final `ELSE` clause in their solution.
- To be awarded marking point 8, the candidate needed to output their results.

Common mistakes candidates made in this question

- Many candidates did not initialise the `CharCount` array elements. They needed to initialise the variables before they were used.
- The question stated that the string contained alphanumeric characters. Many candidates correctly included a `CASE` statement to match each of the vowel characters and to increment the corresponding element in the `CharCount` array. Some candidates used `OTHERWISE`, which would count the number of occurrences of all non-vowel characters, and not just the alphabetic ones. They could have included a conditional test within the `OTHERWISE` clause to fix this.

Question 7

Example Candidate Response – high

7 A procedure, `FormatName()`:

- is called with a string containing words and spaces as its parameter. In this context, a word is any sequence of characters that does not contain a space character.
- creates a new formatted string from this string with the following requirements:
  1. Any leading spaces removed (spaces before the first word).
  2. Any trailing spaces removed (spaces after the last word).
  3. Any multiple spaces between words converted to a single space.
  4. All characters converted to lower case.

The `FormatName()` procedure has been written in a programming language and is to be tested using the black-box method.

(a) Give a test string that could be used to show that all **four** formatting requirements have been applied correctly.

Use the symbol '∇' to represent a space character.

"∇HEllo∇my∇∇NAME∇isA" 1 [3]

(b) The `FormatName()` procedure should assign a value to the global variable `FString`.

There is a fault in the program, which means that the assignment does not always take place.

Explain **two** ways of exposing the fault.

comparing the trace table to the values given by a variable watch. 2

Implementing white box testing which tests the data with values a normal person wouldn't use. [2] 3

Examiner comments

1 The string the candidate gives correctly addresses all the requirements.

Mark for (a) = 3 out of 3

2 The candidate is awarded one mark for referring to the use of a trace table.

3 The candidate needs to give a more detailed explanation about how this would be used to expose the fault.

Mark for (b) = 1 out of 2

Total mark awarded = 4 out of 5

How the candidate could have improved their answer

(b) The candidate's second explanation needed to address the question and explain how the fault may be exposed. Their reference to the term 'white box testing' was not sufficient and the phrase 'with values a normal person would use' did not suggest the investigative use of test data as suggested by the phrase 'try different test values to see which ones fail'.

Example Candidate Response – middle	Examiner comments
7 A procedure, `FormatName()`: • is called with a string containing words and spaces as its parameter. In this context, a word is any sequence of characters that does not contain a space character. • creates a new formatted string from this string with the following requirements: 1. Any leading spaces removed (spaces before the first word). 2. Any trailing spaces removed (spaces after the last word). 3. Any multiple spaces between words converted to a single space. 4. All characters converted to lower case. The `FormatName()` procedure has been written in a programming language and is to be tested using the black-box method. **(a)** Give a test string that could be used to show that all **four** formatting requirements have been applied correctly. Use the symbol '∇' to represent a space character. ∇∇words∇∇words∇∇ 1 ..... [3] **(b)** The `FormatName()` procedure should assign a value to the global variable `FString`. There is a fault in the program, which means that the assignment does not always take place. Explain **two** ways of exposing the fault. By testing all lines 2 ..... [2]	1 The candidate's string addresses requirements 1, 2 and 3, but not requirement 4. Mark for (a) = 2 out of 3 2 The candidate's answer is not detailed enough to be awarded the marks. Mark for (b) = 0 out of 2 **Total mark awarded = 2 out of 5**

How the candidate could have improved their answer

- **(a)** The candidate needed to make sure their string contained at least one upper case character to address requirement 4.
- **(b)** The candidate needed to give a more detailed explanation. The number of answer lines, the marks available and the command word 'explain' indicated the level of detail required.

Example Candidate Response – low	Examiner comments
7 A procedure, `FormatName()`: • is called with a string containing words and spaces as its parameter. In this context, a word is any sequence of characters that does not contain a space character. • creates a new formatted string from this string with the following requirements: 1. Any leading spaces removed (spaces before the first word). 2. Any trailing spaces removed (spaces after the last word). 3. Any multiple spaces between words converted to a single space. 4. All characters converted to lower case. The `FormatName()` procedure has been written in a programming language and is to be tested using the black-box method. **(a)** Give a test string that could be used to show that all **four** formatting requirements have been applied correctly. Use the symbol '∇' to represent a space character. LCASE("∇") returns ∇ 1 [3] **(b)** The `FormatName()` procedure should assign a value to the global variable `FString`. There is a fault in the program, which means that the assignment does not always take place. Explain **two** ways of exposing the fault. -Use of white box testing : Testing every statement one by one- 2 -Use of ~~Beta~~ testers : Dummies to check for any errors 3 [2]	1 The candidate's answer does not address the question. Mark for (a) = 0 out of 3 2 The candidate is awarded a mark for their reference to white box testing and testing every statement. 3 The candidate's answer is not meaningful. Mark for (b) = 1 out of 2 **Total mark awarded = 1 out of 5**

How the candidate could have improved their answer

- **(a)** The candidate's answer did not address the question, it appeared to be a pseudocode statement. The question asked for a test string with characteristics that addressed the four requirements.
- **(b)** It was not clear what the candidate meant by 'Dummies'. The candidate needed to give a more detailed answer.

Common mistakes candidates made in this question

Many candidates gave answers which consisted of lines of pseudocode rather than a test data string.

Question 8

Example Candidate Response – high

8 A program is needed to take a string containing a full name and to produce a new string of initials.

Some words in the full name will be ignored. For example, "the", "and", "of", "for" and "to" may all be ignored.

Each letter of the new string must be upper case.

For example:

Full name	Initials
Integrated Development Environment	IDE
The American Standard Code for Information Interchange	ASCII

The programmer has decided to use the following global variables:

- a ten element 1D array `IgnoreList` of type `STRING` to store the ignored words
- a string `FNString` to store the full name string.

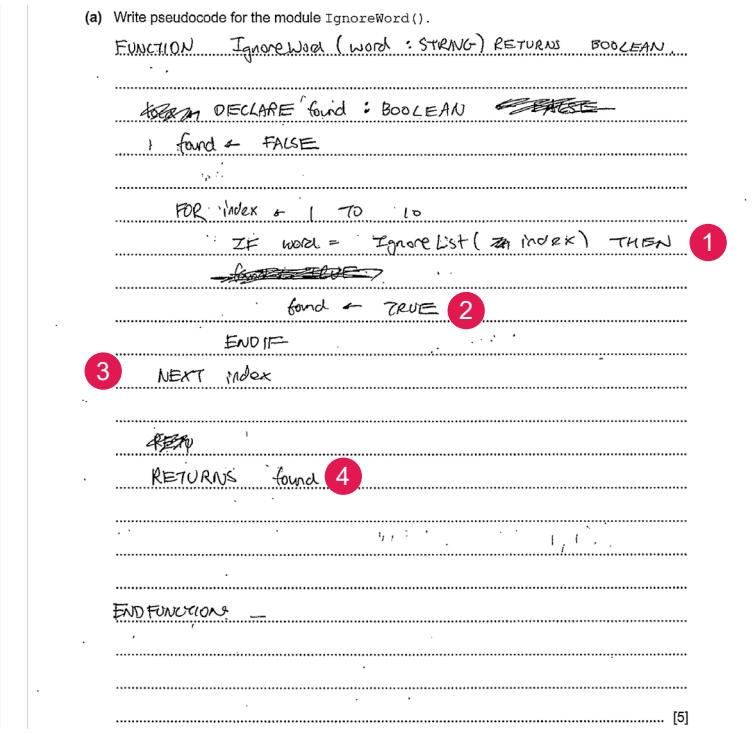
Assume that:

- each alphabetic character in the full name string may be either upper or lower case
- the full name string contains at least one word.

The programmer has started to define program modules as follows:

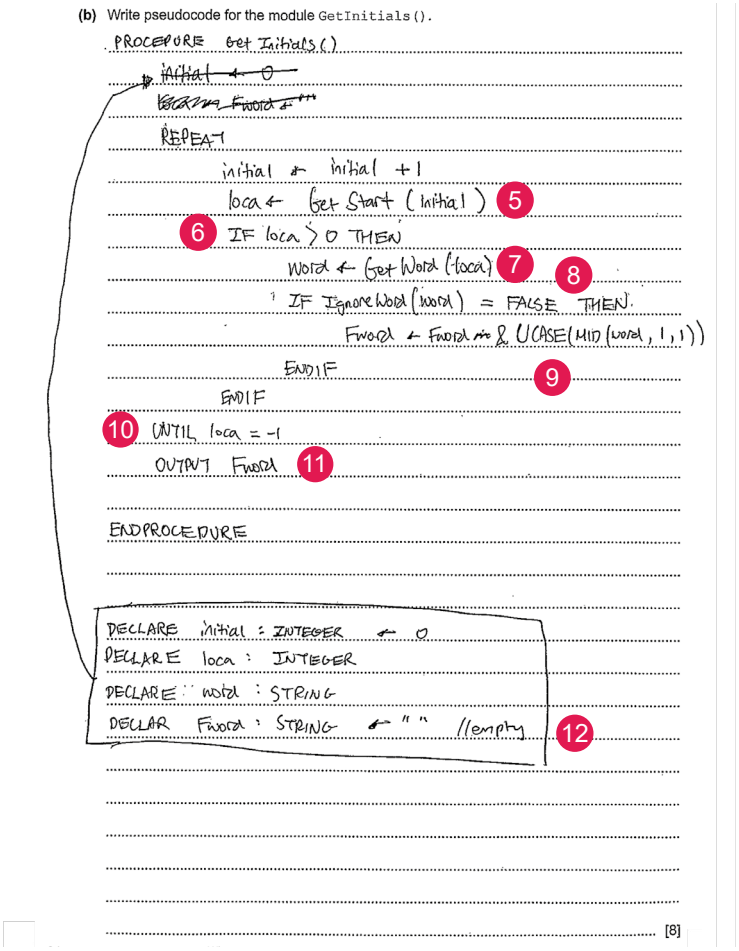
Module	Description
`GetStart()`	• Called with an `INTEGER` as its parameter, representing the number of a word in `FNString` • Returns the character start position of that word in `FNString` or returns `-1` if that word does not exist • For example: `GetStart (3)` applied to `"hot and cold"` returns 9
`GetWord()`	• Called with the position of the first character of a word in `FNString` as its parameter • Returns the word from `FNString` • For example: if `FNString` contains the string `"hot and cold"`, `GetWord(9)` returns `"cold"`
`IgnoreWord()`	• Called with a `STRING` parameter representing a word • Searches for the word in the `IgnoreList` array • Returns `TRUE` if the word is found, otherwise returns `FALSE`
`GetInitials()`	• Processes the sequence of words in the full name one word at a time • Calls `GetStart()`, `GetWord()` and `IgnoreWord()` to process each word to form the new string • Outputs the new string

Examiner comments

Example Candidate Response – high, continued	Examiner comments
	1 The candidate is awarded marking point 3 for their comparison statement. 2 The candidate is awarded marking point 4 for setting a flag when a match is detected. 3 The candidate is awarded marking point 1 for the loop. 4 The candidate is awarded marking point 5 for the return of a Boolean value. Mark for (a) = 4 out of 5

Example Candidate Response – high, continued

(b) Write pseudocode for the module `GetInitials()`.



Examiner comments

5 The candidate uses `GetStart()` so is awarded marking point 3.

6 The candidate is awarded marking point 4 for testing the return value.

7 The candidate uses `GetWord()` so is awarded marking point 5.

8 The candidate is awarded marking point 6 for testing the return value from `IgnoreWord()`.

9 The candidate is awarded marking point 7 for building the new string.

10 The candidate is awarded marking point 2 for the loop.

11 The candidate is awarded marking point 9 for the output of the new string.

12 The candidate is awarded marking point 1 for the declaration and initialisation of the new string.

Mark for (b) = 8 out of 8

Total mark awarded = 12 out of 13

How the candidate could have improved their answer

- **(a)** To be awarded marking point 2, the candidate needed to convert both strings (the parameter value and the value read from the array) to the same case prior to the comparison.
- **(b)** the candidate's solution also addressed marking point 8, but their answer had already been awarded full marks. (The mark scheme allowed for 9 possible mark points but the question was for 8 marks in total)

Example Candidate Response – middle

Examiner comments

8 A program is needed to take a string containing a full name and to produce a new string of initials.

Some words in the full name will be ignored. For example, "the", "and", "of", "for" and "to" may all be ignored.

Each letter of the new string must be upper case.

For example:

Full name	Initials
Integrated Development Environment	IDE
The American Standard Code for Information Interchange	ASCII

The programmer has decided to use the following global variables:

- a ten element 1D array `IgnoreList` of type `STRING` to store the ignored words
- a string `FNString` to store the full name string.

Assume that:

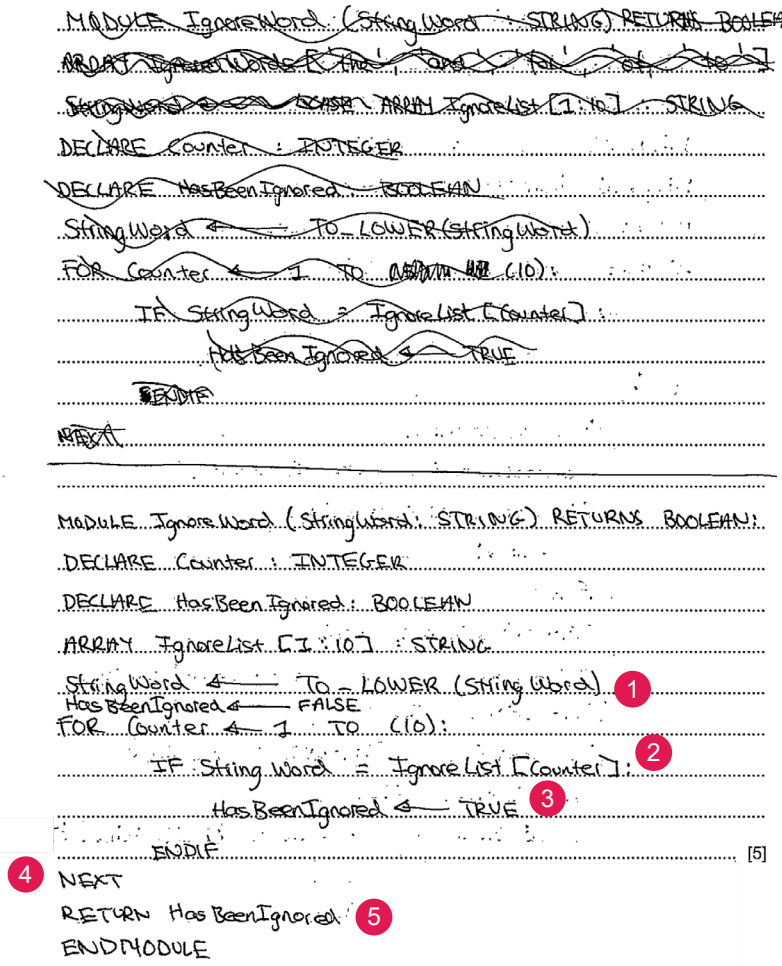
- each alphabetic character in the full name string may be either upper or lower case
- the full name string contains at least one word.

The programmer has started to define program modules as follows:

Module	Description
`GetStart()`	• Called with an `INTEGER` as its parameter, representing the number of a word in `FNString` • Returns the character start position of that word in `FNString` or returns `-1` if that word does not exist • For example: `GetStart(3)` applied to `"hot and cold"` returns `9`
`GetWord()`	• Called with the position of the first character of a word in `FNString` as its parameter • Returns the word from `FNString` • For example: if `FNString` contains the string `"hot and cold"`, `GetWord(9)` returns `"cold"`
`IgnoreWord()`	• Called with a `STRING` parameter representing a word • Searches for the word in the `IgnoreList` array • Returns `TRUE` if the word is found, otherwise returns `FALSE`
`GetInitials()`	• Processes the sequence of words in the full name one word at a time • Calls `GetStart()`, `GetWord()` and `IgnoreWord()` to process each word to form the new string • Outputs the new string

Example Candidate Response – middle, continued

(a) Write pseudocode for the module `IgnoreWord()`.



```
MODULE IgnoreWord (StringWord : STRING) RETURNS BOOLEAN
DECLARE Counter : INTEGER
DECLARE HasBeenIgnored : BOOLEAN
StringWord ← TO_LOWER(StringWord)
FOR Counter ← 1 TO (10):
    IF StringWord = IgnoreList[Counter]:
        HasBeenIgnored ← TRUE
    ENDIF
NEXT
```

```
MODULE IgnoreWord (StringWord : STRING) RETURNS BOOLEAN:
DECLARE Counter : INTEGER
DECLARE HasBeenIgnored : BOOLEAN
ARRAY IgnoreList[1:10] : STRING
StringWord ← TO_LOWER(StringWord)        1
HasBeenIgnored ← FALSE
FOR Counter ← 1 TO (10):
    IF StringWord = IgnoreList[Counter]:        2
        HasBeenIgnored ← TRUE        3
    ENDIF
NEXT        4
RETURN HasBeenIgnored        5
ENDMODULE
```

[5]

Examiner comments

1 The candidate is not awarded marking point 2 because they only convert one of the two strings to a known case.

2 The candidate is awarded marking point 3 for the comparison.

3 The candidate sets a flag value when a match has been detected so is awarded marking point 4.

4 The candidate is awarded marking point 1 for the loop structure.

5 The candidate is awarded marking point 5 for the return of a Boolean value.

Mark for (a) = 4 out of 5

Example Candidate Response – middle, continued

(b) Write pseudocode for the module `GetInitials()`.

```
MODULE GetInitials():
DECLARE FNString : STRING
DECLARE Ignored : BOOLEAN
DECLARE Counter : INTEGER
DECLARE LetterPosition : INTEGER
ARRAY IgnoredList[1:10] : STRING
DECLARE Initials : STRING 6

DO
    GetStart(Counter)
    LetterPosition ← GetStart(Counter) 7
    IF LetterPosition = (-1) : 8
        BREAK
    ENDIF
    GetWord(LetterPosition)
    StringWord ← GetWord(LetterPosition) 9
    IgnoreWord(StringWord)
    Ignored ← IgnoreWord(StringWord)
    IF Ignored = FALSE: 10
        Initials ← Initials & UCASE(MID(StringWord,1,1)) 11
12  ENDIF
13 UNTIL EOF(FNString) = TRUE
OUTPUT(Initials) 14
ENDMODULE
```

[8]

Examiner comments

6 The candidate is not awarded marking point 1. Although they declare the variable `Initials`, they do not initialise it.

7 The candidate uses `GetWord()` so is awarded marking point 5.

8 The candidate is awarded marking point 4 for testing the return value.

9 The candidate uses `GetWord()` so is awarded marking point 5.

10 The candidate is awarded marking point 6 for testing the return value.

11 The candidate is awarded marking point 7 for building the new string.

12 The candidate does not attempt marking point 8.

13 The candidate is not awarded marking point 2 because their loop termination condition is invalid.

14 The candidate is awarded marking point 9 for the output of the new string.

Mark for (b) = 5 out of 8

Total mark awarded = 9 out of 13

How the candidate could have improved their answer

- **(a)** To be awarded marking point 2, the candidate needed to convert the parameter value to lower case prior to the comparison.
- **(b)** To be awarded marking point 1, the candidate needed to initialise the variable `Initials` to an empty string, following the declaration.
- **(b)** To be awarded marking point 2, the candidate needed to repeat the loop until all of the words had been extracted from `FNString`. This would have been indicated by function `GetStart()` returning the value -1.
- **(b)** To be awarded marking point 3, the candidate needed to set the parameter value of function `GetStart()` to the initial value of 1 and then subsequently increment this each time around the loop.
- **(b)** To be awarded marking point 8, the candidate needed to increment the variable 'Counter' within the loop.

Example Candidate Response – low

Examiner comments

8 A program is needed to take a string containing a full name and to produce a new string of initials.

Some words in the full name will be ignored. For example, "the", "and", "of", "for" and "to" may all be ignored.

Each letter of the new string must be upper case.

For example:

Full name	Initials
Integrated Development Environment	IDE
The American Standard Code for Information Interchange	ASCII

The programmer has decided to use the following global variables:

- a ten element 1D array `IgnoreList` of type `STRING` to store the ignored words
- a string `FNString` to store the full name string.

Assume that:

- each alphabetic character in the full name string may be either upper or lower case
- the full name string contains at least one word.

The programmer has started to define program modules as follows:

Module	Description
`GetStart()`	• Called with an `INTEGER` as its parameter, representing the number of a word in `FNString` • Returns the character start position of that word in `FNString` or returns `-1` if that word does not exist • For example: `GetStart(3)` applied to `"hot and cold"` returns `9`
`GetWord()`	• Called with the position of the first character of a word in `FNString` as its parameter • Returns the word from `FNString` • For example: if `FNString` contains the string `"hot and cold"`, `GetWord(9)` returns `"cold"`
`IgnoreWord()`	• Called with a `STRING` parameter representing a word • Searches for the word in the `IgnoreList` array • Returns `TRUE` if the word is found, otherwise returns `FALSE`
`GetInitials()`	• Processes the sequence of words in the full name one word at a time • Calls `GetStart()`, `GetWord()` and `IgnoreWord()` to process each word to form the new string • Outputs the new string

Example Candidate Response – low, continued

(a) Write pseudocode for the module IgnoreWord().

```
FUNCTION IgnoreWord(String: STRING) RETURN
BOOLEAN
DECLARE Index: INTEGER
DECLARE Found: BOOLEAN
Found ← FALSE
Index ← 1
WHILE NOT Found OR Index < 11
    IF LCASE(String) = IgnoreList[Index]
        Found = TRUE
    ENDIF
    Index ← Index + 1
ENDWHILE

RETURN Found
ENDFUNCTION
```

[5]

Examiner comments

1 The candidate incorrectly uses the keyword STRING as an identifier name. In this case it is not relevant to any of the available mark points.

2 The condition is incorrect so the candidate is not awarded marking point 1.

3 The candidate is not awarded marking point 2.

4 The candidate is not awarded marking point 3 for their comparison.

5 The candidate sets a glad when a match is detected so is awarded marking point 4.

6 The candidate is awarded marking point 5 for the return of a Boolean value.

Mark for (a) = 3 out of 5

Example Candidate Response – low, continued

(b) Write pseudocode for the module `GetInitials()`.

```
PROCEDURE GetInitials(FNString: STRING)
    DECLARE WordNum: INTEGER
    DECLARE 1stChar: INTEGER
    DECLARE String: STRING
    DECLARE Initials: STRING
    String ← ""
    Initials ← "" 7
    WordNum ← 1
    1stChar ← 0
    WHILE 1stChar <> -1
        1stChar ← GetStart(WordNum) 8
        String ← GetWord(1stChar) 10
9       IF IgnoreWord(String) ← TRUE 11
        THEN
            WordNum ← WordNum + 1 12
        ENDIF
        IF IgnoreWord(String) ← FALSE
        THEN
            Initials ← Initials & String[1] 13
        ENDIF
        WordNum ← WordNum + 1
14  ENDWHILE

    OUTPUT Initials 15
END PROCEDURE
```

[8]

Examiner comments

7 The candidate is awarded marking point 1 for their declaration and initialisation of the variable `Initials`.

8 The candidate uses of `GetStart()` so is awarded marking point 3.

9 The candidate is not awarded marking point 4 because they make no attempt to test the return value from `GetStart()`.

10 The candidate is awarded marking point 5 for their assignment of the value returned from `GetWord()`.

11 The candidate uses the backarrow assignment symbol incorrectly in the comparison statement so they are not awarded marking point 6.

12 The candidate's preceding logic (and hence the context) is incorrect, so they are not awarded marking point 8.

13 The candidate is not awarded marking point 7 because `String[1]` is not a valid sub-string operation.

14 The candidate is not awarded marking point 2 because the loop statements would be executed even after `GetStart()` returned -1.

15 The candidate does not convert the characters in variable `Initials` to upper case so they are not awarded marking point 9.

Mark for (b) = 3 out of 8

Total mark awarded = 6 out of 13

How the candidate could have improved their answer

- The candidate used the identifier `String` to indicate a variable. `STRING` is a keyword and as such should not have been used as an identifier name. In this case of this question, no mark points were directly affected.
- **(a)** To be awarded marking point 1, the candidate needed to use correct the loop termination condition `AND` rather than `OR`.
- **(a)** To be awarded marking point 2, the candidate needed to convert the value read from the array to lower case prior to the comparison.

- **(b)** To be awarded marking point 2, the candidate needed to terminate the loop as soon as the function `GetStart()` had returned the value -1, to indicate that there were no more words in `FNString`.
- **(b)** To be awarded marking point 4, the candidate needed to test the value returned by function `GetStart()`.
- **(b)** To be awarded marking point 6, the candidate needed to use the correct syntax for the comparison statement. It should have been: `IF IgnoreWord(String) = TRUE`.
- **(b)** To be awarded marking point 7, the candidate needed to use the correct sub-string function to extract the first character from the identifier `String`.
- **(b)** To be awarded marking point 8, the candidate needed to make sure the logic preceding the increment was correct. The variable was incremented under the wrong circumstances.
- **(b)** To be awarded marking point 9, the candidate needed to convert the characters used to construct the output string to upper case.

Common mistakes candidates made in this question

- **(a)** Many candidates did not attempt to convert the strings being compared to a known case (upper or lower).
- **(b)** Many candidates struggled with user-defined functions. For example, the function `GetStart()` is described as taking an integer as a parameter and returning an integer value. An example of a valid pseudocode statement which uses this function is:

  Index ← `GetStart(ThisWordNum)`

  Many candidates treated the function as a procedure and the return value was ignored, for example:

  `CALL GetStart(ThisWordNum)`

  This immediately introduced a new problem as there was no return value to test.
- **(b)** Many candidates used the arithmetic operator '`+`' to concatenate strings rather than the correct string operator '`&`'.

Cambridge Assessment International Education
The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA, United Kingdom
t: +44 1223 553554
e: info@cambridgeinternational.org www.cambridgeinternational.org

© Cambridge University Press & Assessment 2022 v1