L3 Lead Examiner Report 1906

Summer 2019

L3 Qualification in Computing

Unit 1: Principles of Computer Systems

Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications come from Pearson, the world's leading learning company. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications website at http://qualifications.pearson.com/en/home.html for our BTEC qualifications.

Alternatively, you can get in touch with us using the details on our contact us page at http://qualifications.pearson.com/en/contact-us.html

If you have any subject specific questions about this specification that require the help of a subject specialist, you can speak directly to the subject team at Pearson. Their contact details can be found on this link:
http://qualifications.pearson.com/en/support/support-for-you/teachers.html

You can also use our online Ask the Expert service at https://www.edexcelonline.com

You will need an Edexcel Online username and password to access this service.

Pearson: helping people progress, everywhere

Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your learners at: www.pearson.com/uk

Summer 2019
Publications Code 31768_1906_ER

Grade Boundaries

What is a grade boundary?

A grade boundary is where we set the level of achievement required to obtain a certain grade for the externally assessed unit. We set grade boundaries for each grade, at Distinction, Merit and Pass.

Setting grade boundaries

When we set grade boundaries, we look at the performance of every learner who took the external assessment. When we can see the full picture of performance, our experts are then able to decide where best to place the grade boundaries – this means that they decide what the lowest possible mark is for a particular grade.

When our experts set the grade boundaries, they make sure that learners receive grades which reflect their ability. Awarding grade boundaries is conducted to ensure learners achieve the grade they deserve to achieve, irrespective of variation in the external assessment.

Variations in external assessments

Each external assessment we set asks different questions and may assess different parts of the unit content outlined in the specification. It would be unfair to learners if we set the same grade boundaries for each assessment, because then it would not take accessibility into account.

Grade boundaries for this, and all other papers, are on the website via this link:

http://qualifications.pearson.com/en/support/support-topics/results-certification/grade-boundaries.html

Unit 1: Principles of Computer Science

Grade	Unclassified	N grade	Level 3		
			Pass	Merit	Distinction
Boundary Mark	0	11	22	36	51

Introduction

This was the fifth examination season for Level 3 BTEC Computing Unit 1 Principles of Computer Science 31768H.

This unit is assessed through a single written examination which is two hours in length and the number of marks available is 90.

This unit is a mandatory unit for all learners studying the extended certificate, foundation diploma, all diplomas and the extended diploma.

The examination for this unit will always contain four sections and each section will have a scenario that will be used throughout the whole of that section. The scenario will be clearly stated at the beginning of each section.

Each section is broken down into sub-questions which will then test learners on different areas of the specification and learners should be expected to apply their knowledge to the scenario.

Learners will be given an information booklet. They will be instructed to look at individual parts / sections of this during the examination in order to answer questions.
The information booklet **may** give learners:

1. Information about problems that they need to solve.
2. Programming code for them to interpret, analyse or evaluate.
3. Requirements or designs for a new program that is needed.
4. An algorithm for them to interpret, analyse or evaluate.

At no point during the examination will learners be expected to write code in a particular language. Learners will only be given small pieces code to interpret, analyse or evaluate.

All sections of the examination paper provide differentiation at all attainment levels and the paper is designed to be ramped in difficulty so that a larger percentage of higher grade marks are allocated to the later stages of the paper.

Introduction to the Overall Performance of the Unit

The overall performance of learners is similar to the last examination season for this unit. The average mark per candidate has risen slightly which shows that centres are better preparing learners for the rigor of this exam. However, there is still evidence to suggest that there are still a lot of learners who are not fully prepared to take this examination.

It is worth noting that the recommended Guided Learning Hours (GLH) for this unit is 120. It is recommended that centres ensure that this amount of time is used to ensure that learners are equipped with the knowledge to allow them to answer a range of different questions covering the whole specification.

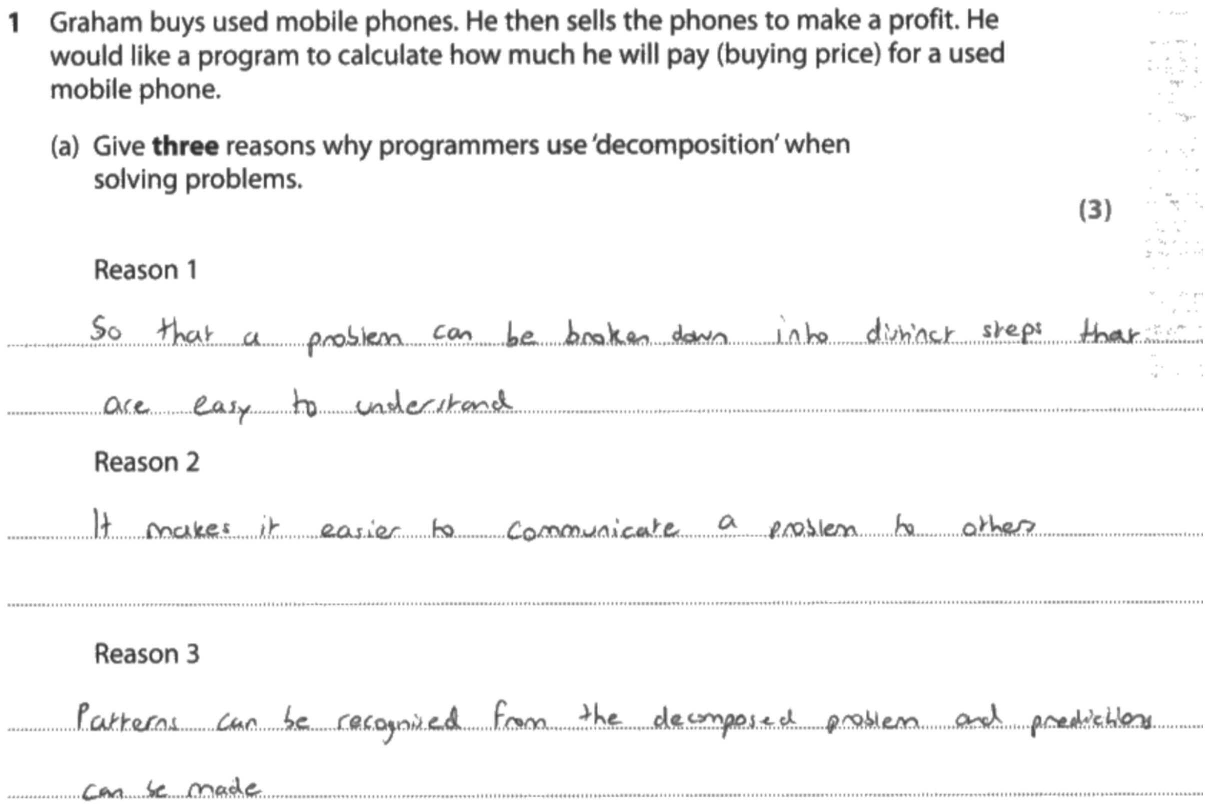
While learners did not perform well on some of the longer questions, on the whole the performance on the shorter response questions appears to have improved with many learners picking up some marks on each short answer question. The performance on the extended questions has remained in line with the previous exam season. Learners still do not fully understand the demands of the higher order command words such as discuss, analyse and evaluate. Learners were not able to meet the demands of these higher order command verbs which resulted in many learners achieving lower marks on the extended questions.

Individual Questions

The following section considers each question on the paper, providing examples of learner responses and a brief commentary of why the responses gained the marks they did. This section should be considered with the live external assessment and corresponding mark scheme.

Question 1a (3 Marks)

The majority of learners gained two marks. Most learners were able to identify two reasons why programmers use decomposition, some learners however struggled to find a third reason third mark was often missed by repeating a previous reason or giving vague answers such as “it is quicker”



1 Graham buys used mobile phones. He then sells the phones to make a profit. He would like a program to calculate how much he will pay (buying price) for a used mobile phone.

(a) Give **three** reasons why programmers use ‘decomposition’ when solving problems.

(3)

Reason 1

So that a problem can be broken down into distinct steps that are easy to understand

Reason 2

It makes it easier to communicate a problem to others

Reason 3

Patterns can be recognized from the decomposed problem and predictions can be made

3 Marks awarded for:

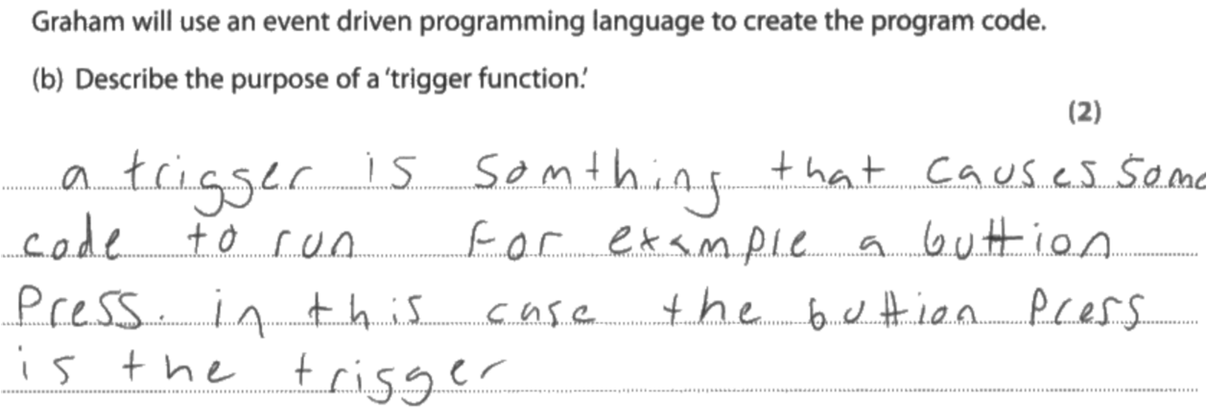
Problem can be broken down 1 Mark

Communicate a problem to other 1 Mark

Recognise patterns 1 Mark

Question 1b (2 Marks)

Most learners gained two marks. Learners were able to give a clear description of a trigger function, sometime by giving examples of user action or processes.



Graham will use an event driven programming language to create the program code.

(b) Describe the purpose of a 'trigger function.'

(2)

a trigger is somthing that causes some code to run For example a buttion Press. in this case the buttion Press is the trigger

2 Marks awarded for:

'Causes some code to run' (1 mark)

'For example a button press' (1 mark)

Question 1c (5 Marks)

Learners often scored 5 marks for completing the formula in the required boxes. Common reasons why marks were not awarded included errors in the formulae such as dividing by 100 and then multiplying by 0.25 rather than 25. Where a learner had repeated the same error in all boxes they able to gain one mark.

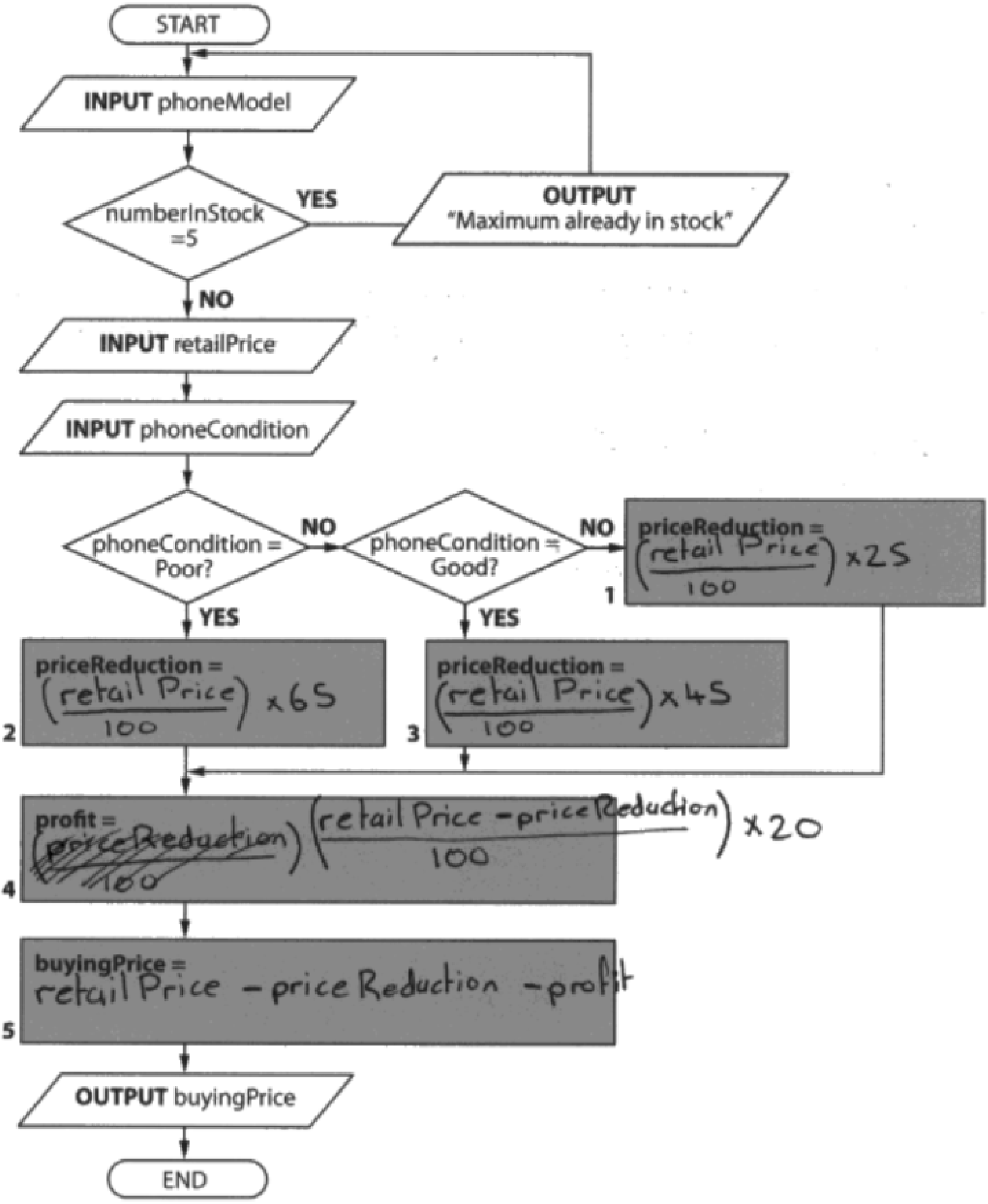
All acceptable variations of correct formulae were given full credit.

(c) Graham has started a flowchart to show the logic of the program.

The completed flowchart must meet the rules given in **Figure 1**.

Complete the statements in the shaded boxes numbered 1 to 5 to show the calculations the program will perform for each variable.

(5)


START
INPUT phoneModel
numberInStock =5
YES
OUTPUT "Maximum already in stock"
NO
INPUT retailPrice
INPUT phoneCondition
phoneCondition = Poor?
NO
phoneCondition = Good?
NO
1
priceReduction = (retail Price / 100) x25
YES
YES
2
priceReduction = (retail Price / 100) x65
3
priceReduction = (retail Price / 100) x45
4
profit = ~~(priceReduction / 100)~~ ((retail Price - priceReduction) / 100) x20
5
buyingPrice = retail Price - price Reduction - profit
OUTPUT buyingPrice
END

5 Marks awarded for:

Process Symbol 1 - retailPrice/100*25 (1 Mark)

Process Symbol 2 - retailPrice/100*65 (1 Mark)

Process Symbol 3 - retailPrice/100*45 (1 Mark)

Process Symbol 4 - (retailPrice – priceReduction)/100*20 (1 Mark)

Process Symbol 5 - retailPrice –priceReduction-profit (1 Mark)

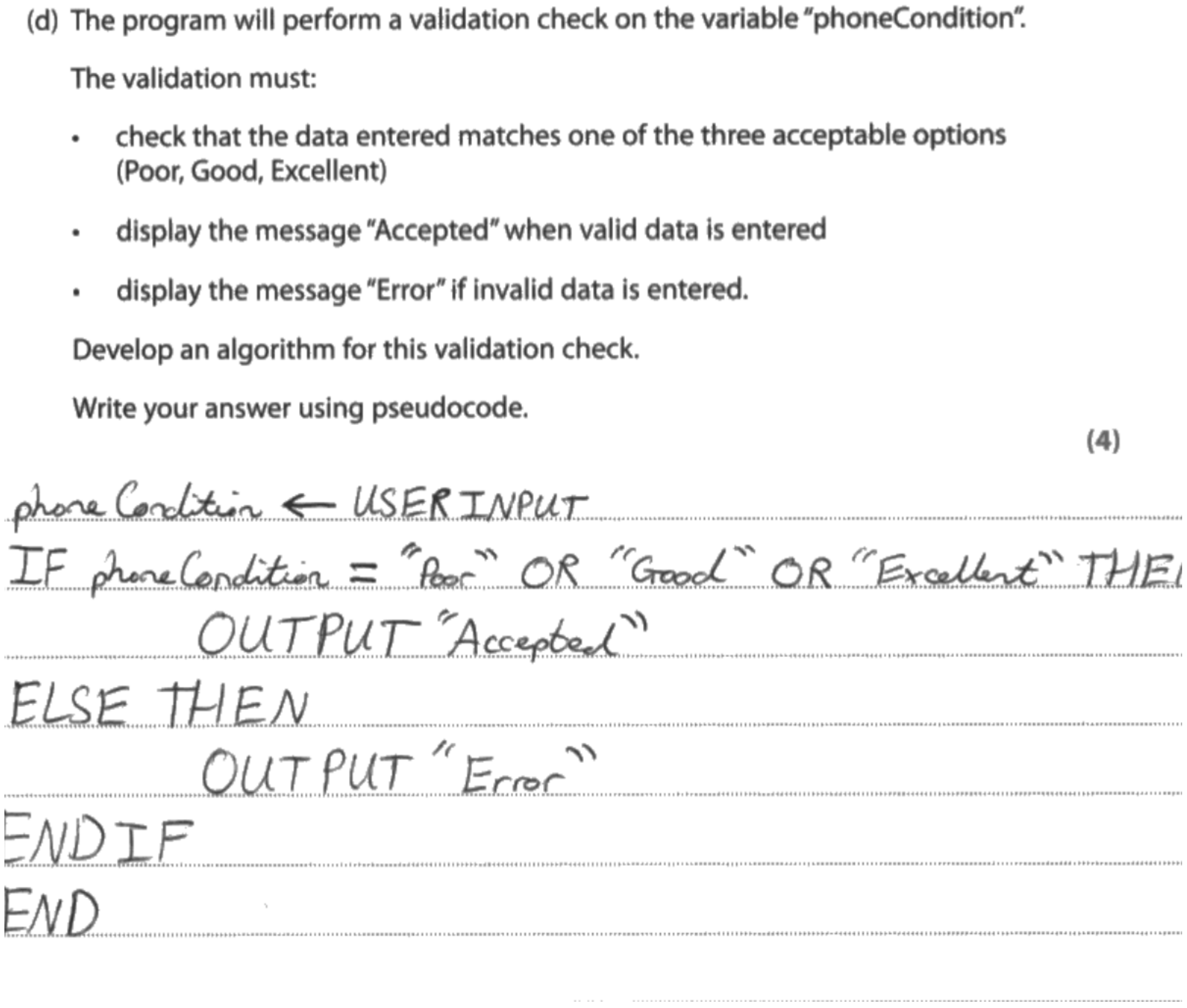
Question 1d (4 Marks)

A many of learners manage to score four marks on this question. Most learners were able to use the correct logic for a solution, some using the logic operator OR to combine all conditions on one line. Many used 3 separate conditions using ELIF or a combination/series of IF statements.

The main reason some learners only gained 3 marks was not using a variable name that was linked to variable name specified in the question "phoneCondition". However any recognised variation of the variable name was accepted.

The answer was written using pseudocode so any recognizable syntax was accepted.

Some learners included loops as part of their answer to ensure a valid value was entered however tis was not necessary to gain full marks.



(d) The program will perform a validation check on the variable "phoneCondition".

The validation must:

- check that the data entered matches one of the three acceptable options (Poor, Good, Excellent)
- display the message "Accepted" when valid data is entered
- display the message "Error" if invalid data is entered.

Develop an algorithm for this validation check.

Write your answer using pseudocode.

(4)

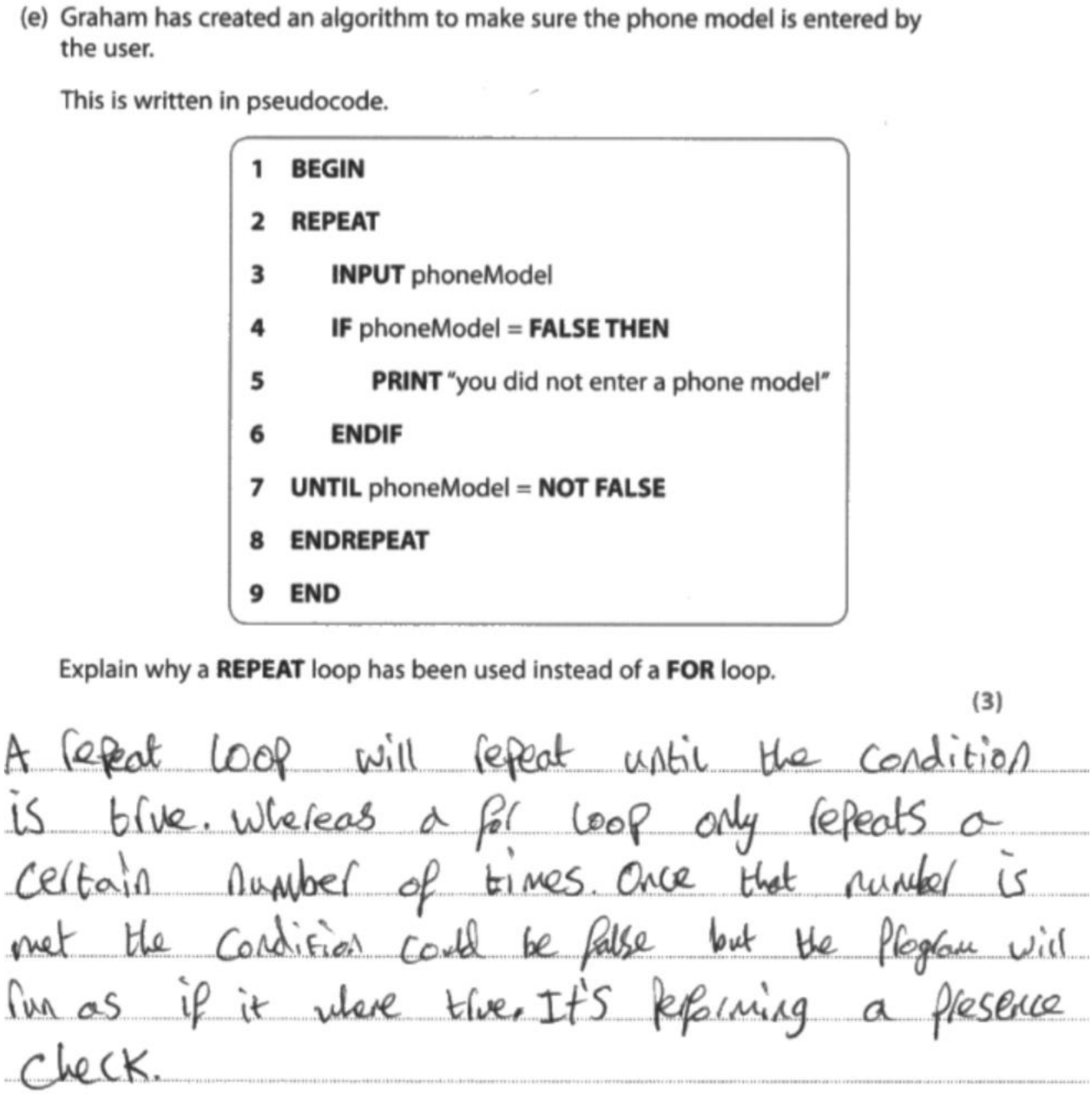
```
phoneCondition ← USERINPUT
IF phoneCondition = "Poor" OR "Good" OR "Excellent" THEN
        OUTPUT "Accepted"
ELSE THEN
        OUTPUT "Error"
ENDIF
END
```

4 Marks awarded for:

- testing the variable phoneCondition (1 mark)
- checks that data entered matches one of the three options (poor, good, excellent) (1 mark)
- correctly printing "Accepted" for valid data (1 mark)
- correctly printing "Error" for invalid data (1 mark)

Question 1e (3 Marks)

Learners were able to score 3 marks on this question. Most learners could explain that the FOR loop was a fixed number of times, and the REPEAT loop would continue as long as invalid data was input. However, many learners did not realise that the FOR loop could eventually pass invalid data on to the rest of the program.



(e) Graham has created an algorithm to make sure the phone model is entered by the user.

This is written in pseudocode.

```
BEGIN
REPEAT
    INPUT phoneModel
    IF phoneModel = FALSE THEN
        PRINT "you did not enter a phone model"
    ENDIF
UNTIL phoneModel = NOT FALSE
ENDREPEAT
END
```

Explain why a **REPEAT** loop has been used instead of a **FOR** loop.

(3)

A repeat loop will repeat until the condition is true. whereas a for loop only repeats a certain number of times. Once that number is met the condition could be false but the program will run as if it were true. It's performing a presence check.

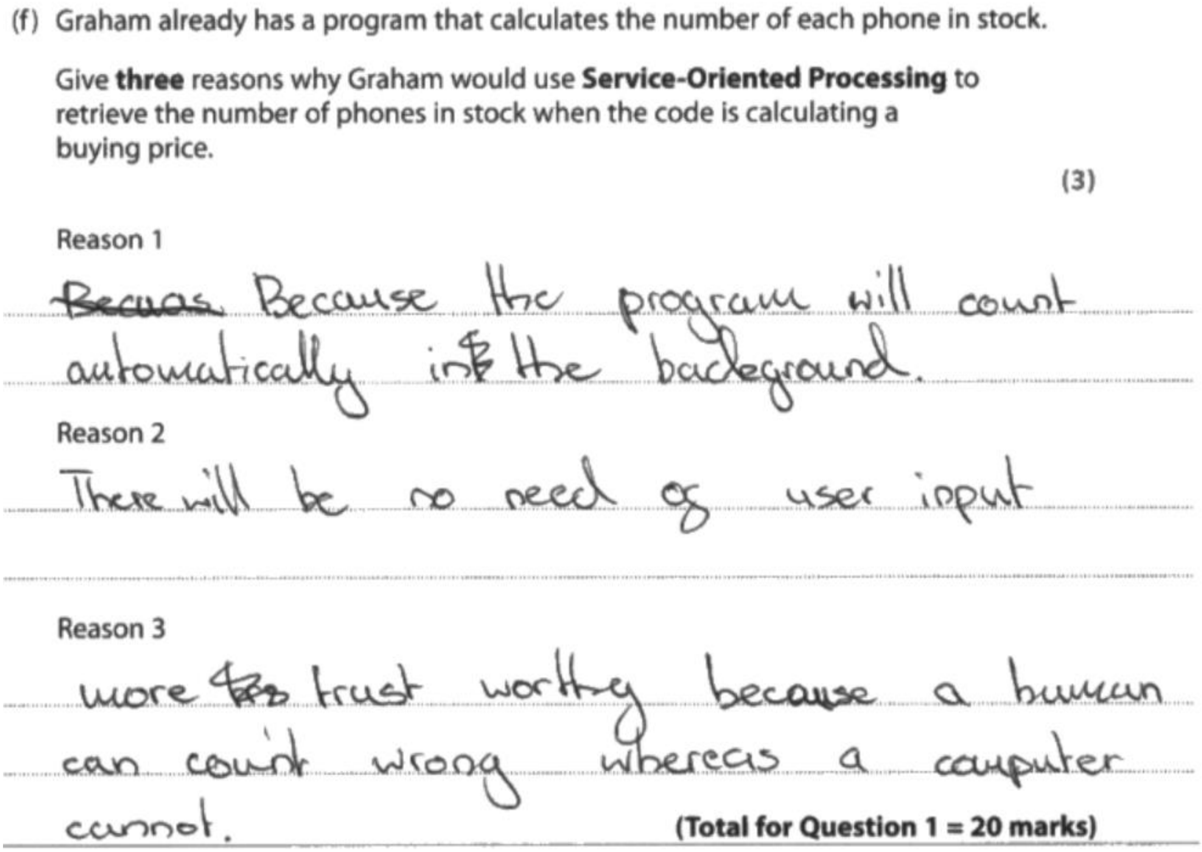
3 Marks awarded for:

- FOR loop fixed number of times (1 mark)
- A repeat loop will repeat until the condition is true (1 mark)
- avoid passing an invalid value to another part of the program (1 mark)

Question 1f (3 Marks)

Many learners found this question challenging. Only a few learners managed to score 3 marks. The most common answer gaining credit was for stating that the task can run in the background or that the system runs multiple tasks at the same time.

Answer often focused on why the numbers needed to be calculated, or how the price was calculated rather than the use of Service-Oriented Processing



(f) Graham already has a program that calculates the number of each phone in stock.

Give **three** reasons why Graham would use **Service-Oriented Processing** to retrieve the number of phones in stock when the code is calculating a buying price.

(3)

Reason 1

~~Becaus~~ Because the program will count automatically in~~t~~ the background.

Reason 2

There will be no need og user input

Reason 3

more ~~tr~~ trust worthy because a human can count wrong whereas a computer cannot.

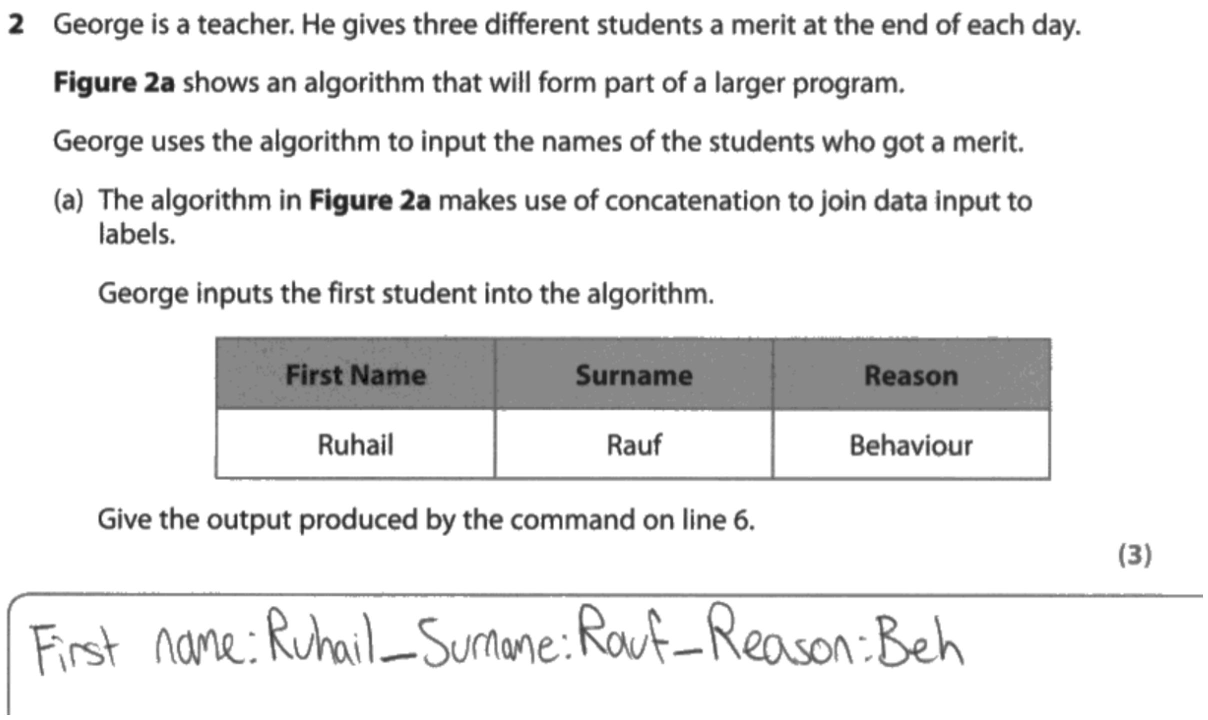
(Total for Question 1 = 20 marks)

2 Marks awarded for:

- Automatically in the background (1 mark)
- No need of user input (1 Mark)

Question 2a (3 Marks)

The majority of learners scored 3 marks on this question. The most common reason for not awarding the third mark was not truncating the reason. Many learners reproducing the full word “behaviour” at the end of the printout. Some learners placed each section on a new line and so only gained the first mark for concatenating first name.



2 George is a teacher. He gives three different students a merit at the end of each day.

Figure 2a shows an algorithm that will form part of a larger program.

George uses the algorithm to input the names of the students who got a merit.

(a) The algorithm in **Figure 2a** makes use of concatenation to join data input to labels.

George inputs the first student into the algorithm.

First Name	Surname	Reason
Ruhail	Rauf	Behaviour

Give the output produced by the command on line 6.

(3)

First name:Ruhail_Surname:Rauf_Reason:Beh

3 Marks awarded for:

Concatenating the First name label with the associated data (First name:Ruhail) (1 Mark)

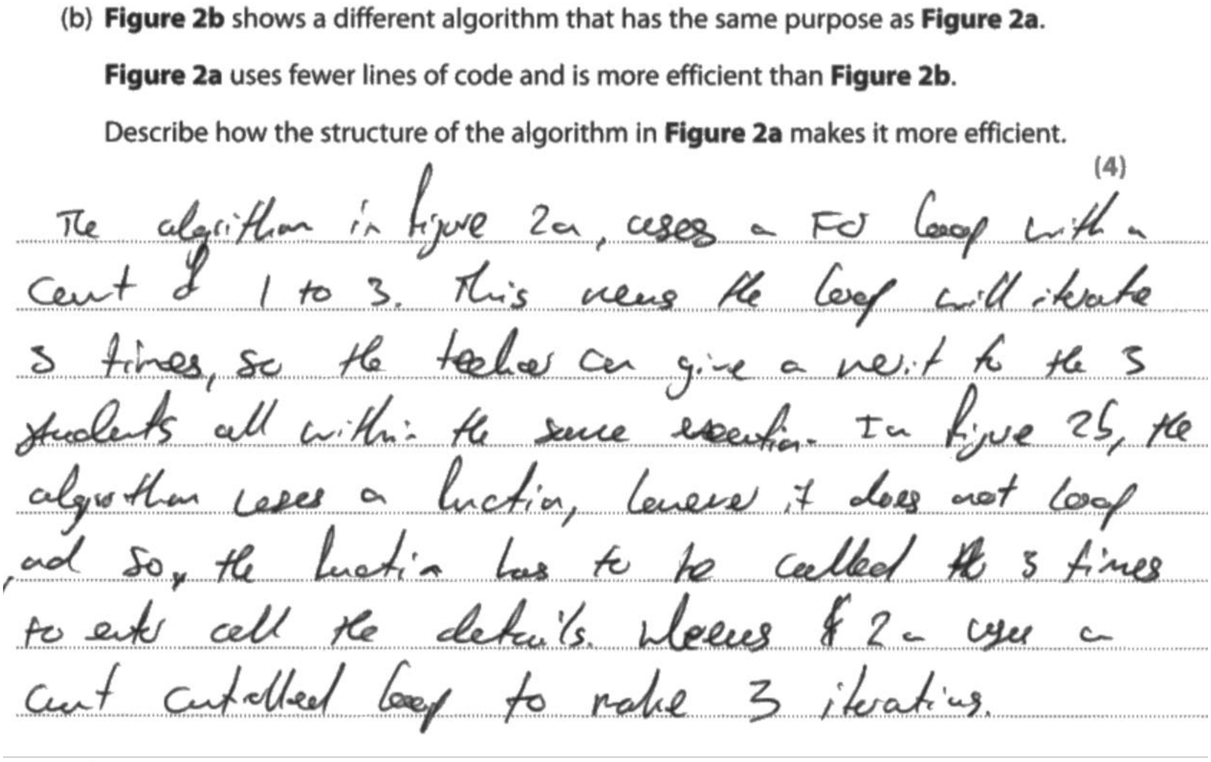
Concatenating the Surname label with the associated data (_Surname:Rauf) (1 Mark)

Concatenating the Score label with the associated **truncated** data (_Reason:Beh) (1 Mark)

Question 2b (4 Marks)

Some learners were able to score four marks on this question. However, many learners did not address the main part of the question which was describing how structure of the algorithm in 2b was more efficient.

Many learners simply described the two algorithms and did not refer to efficiency.



(b) **Figure 2b** shows a different algorithm that has the same purpose as **Figure 2a**.

Figure 2a uses fewer lines of code and is more efficient than **Figure 2b**.

Describe how the structure of the algorithm in **Figure 2a** makes it more efficient.

(4)

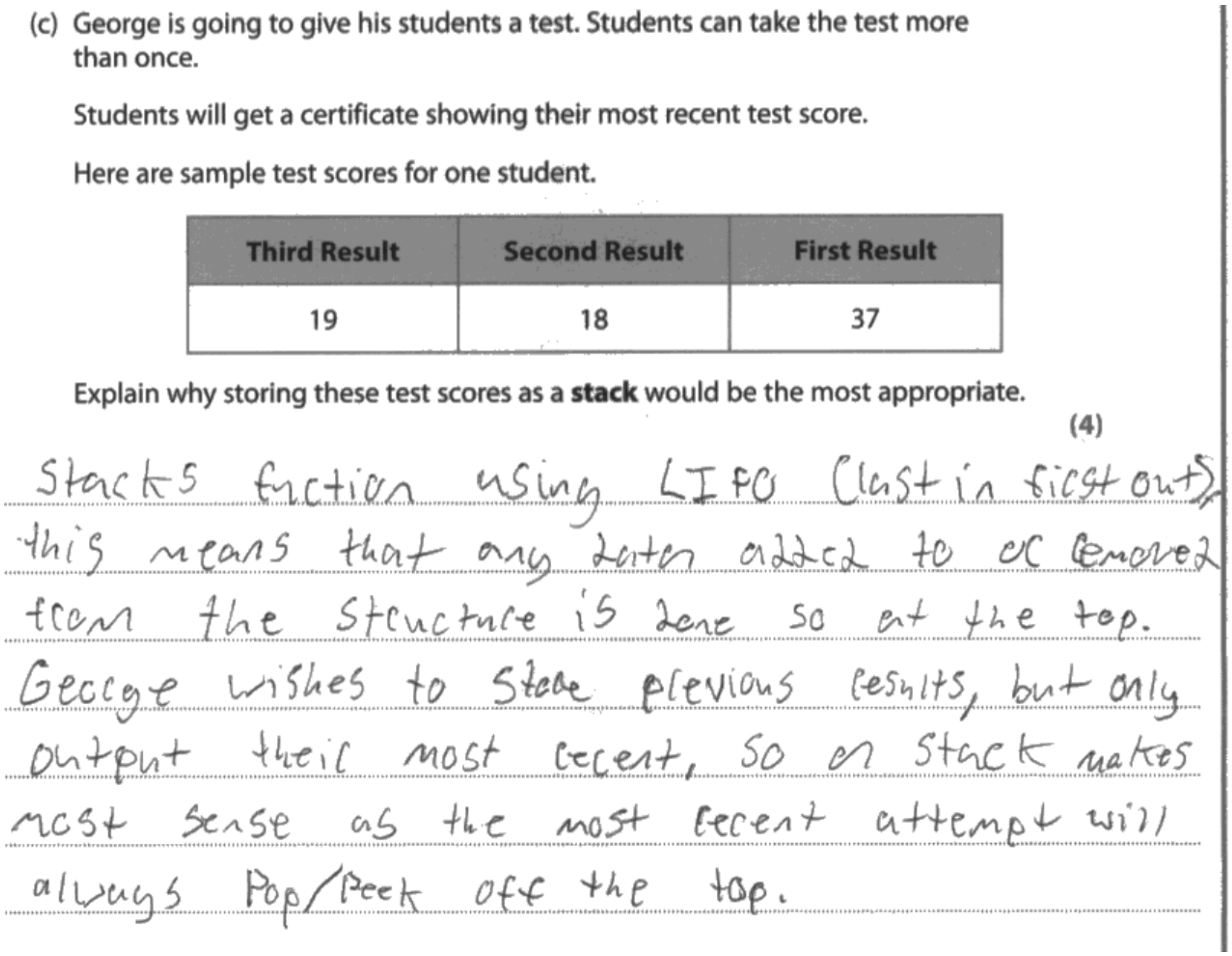
The algorithm in figure 2a, uses a FOR loop with a count of 1 to 3. This means the loop will iterate 3 times, so the teacher can give a result to the 3 students all within the same execution. In figure 2b, the algorithm uses a function, however it does not loop and so, the function has to be called the 3 times to enter all the details. Whereas f 2a uses a count controlled loop to make 3 iterations.

4 Marks awarded for:

- it uses a FOR loop (1 Mark)
- the loop iterates over the same small amount of code (1 Mark)
- figure 2b places the code in a function (1 Mark)
- function has to be called separately each time it's used (1 Mark)

Question 2c (4 Marks)

Most learners scored three or four marks for this question. Most were able to describe the stack structure and state that the most recent mark would be on top, however they did not then relate this to certificate displaying only the most recent score. Some learners compared the stack to a queue, explaining why the queue would not work, which was not required.



(c) George is going to give his students a test. Students can take the test more than once.

Students will get a certificate showing their most recent test score.

Here are sample test scores for one student.

Third Result	Second Result	First Result
19	18	37

Explain why storing these test scores as a **stack** would be the most appropriate.

(4)

Stacks fuction using LIFO (last in first out), this means that any data added to or removed from the structure is done so at the top. George wishes to store previous results, but only output their most recent, so a stack makes most sense as the most recent attempt will always Pop/Peek off the top.

4 Marks awarded for:

Last In First Out (LIFO) (1 Mark)

data added to the top of the stack (1 Mark)

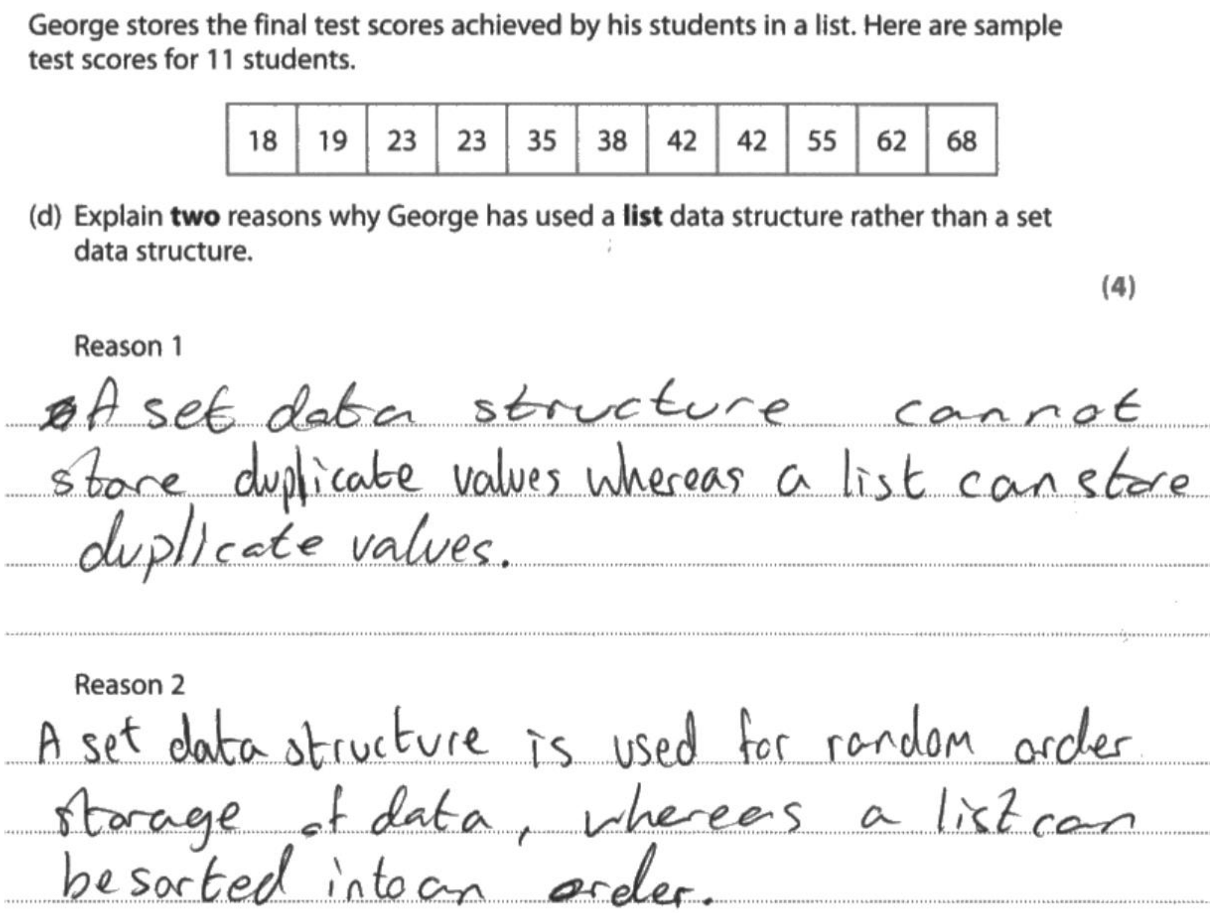
most recent test score will be at the top of the stack (1 Mark)

can output/pop/peek the value on top of the stack (1 Mark)

Question 2d (4 Marks)

Learners generally did not perform very well on this question; many learners were only able to explain one reason. Often, learners were able to give a difference between list and a set, or state what a list or set looked like. However they were not able to explain the impact of these on the test scores and therefore why George used a list.

The most frequent answers referred to the sorting of the list, or the repetition of data in the list.



George stores the final test scores achieved by his students in a list. Here are sample test scores for 11 students.

18	19	23	23	35	38	42	42	55	62	68

(d) Explain **two** reasons why George has used a **list** data structure rather than a set data structure.

(4)

Reason 1

A set data structure cannot store duplicate values whereas a list can store duplicate values.

Reason 2

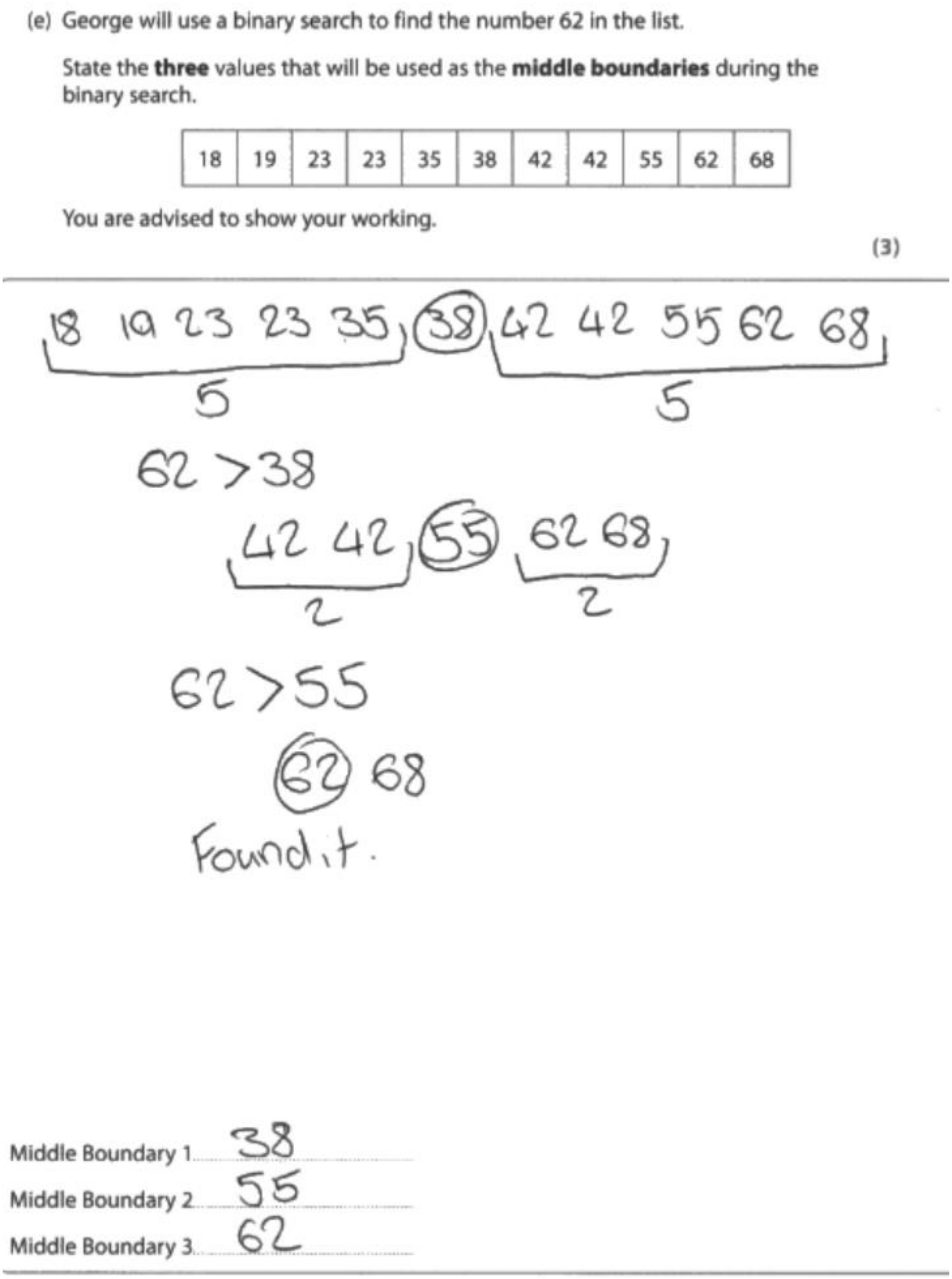
A set data structure is used for random order storage of data, whereas a list can be sorted into an order.

4 Marks awarded for:

- A list can be sorted (1) however a set may not maintain the order of values (1)
- The same test score may appear several times (1) however a set does not allow duplicate values to be stored (1)

Question 2e (3 Marks)

Most learners picked up 3 marks on this question. With some learners showing the working out. Some learners showing the working out were able to gain some marks for a partly correct answer as the method used enabled the process to be seen and credit given for this. For example choosing the wrong first value, but then correctly the next values based on that.



(e) George will use a binary search to find the number 62 in the list.

State the **three** values that will be used as the **middle boundaries** during the binary search.

18	19	23	23	35	38	42	42	55	62	68

You are advised to show your working.

(3)

$\underbrace{18\ \ 19\ \ 23\ \ 23\ \ 35}_{5},\boxed{38},\underbrace{42\ \ 42\ \ 55\ \ 62\ \ 68}_{5}$

$62 > 38$

$\underbrace{42\ \ 42}_{2},\boxed{55}\ \underbrace{62\ \ 68}_{2}$

$62 > 55$

$\boxed{62}\ 68$

Found it.

Middle Boundary 1 38

Middle Boundary 2 55

Middle Boundary 3 62

3 Marks awarded for:

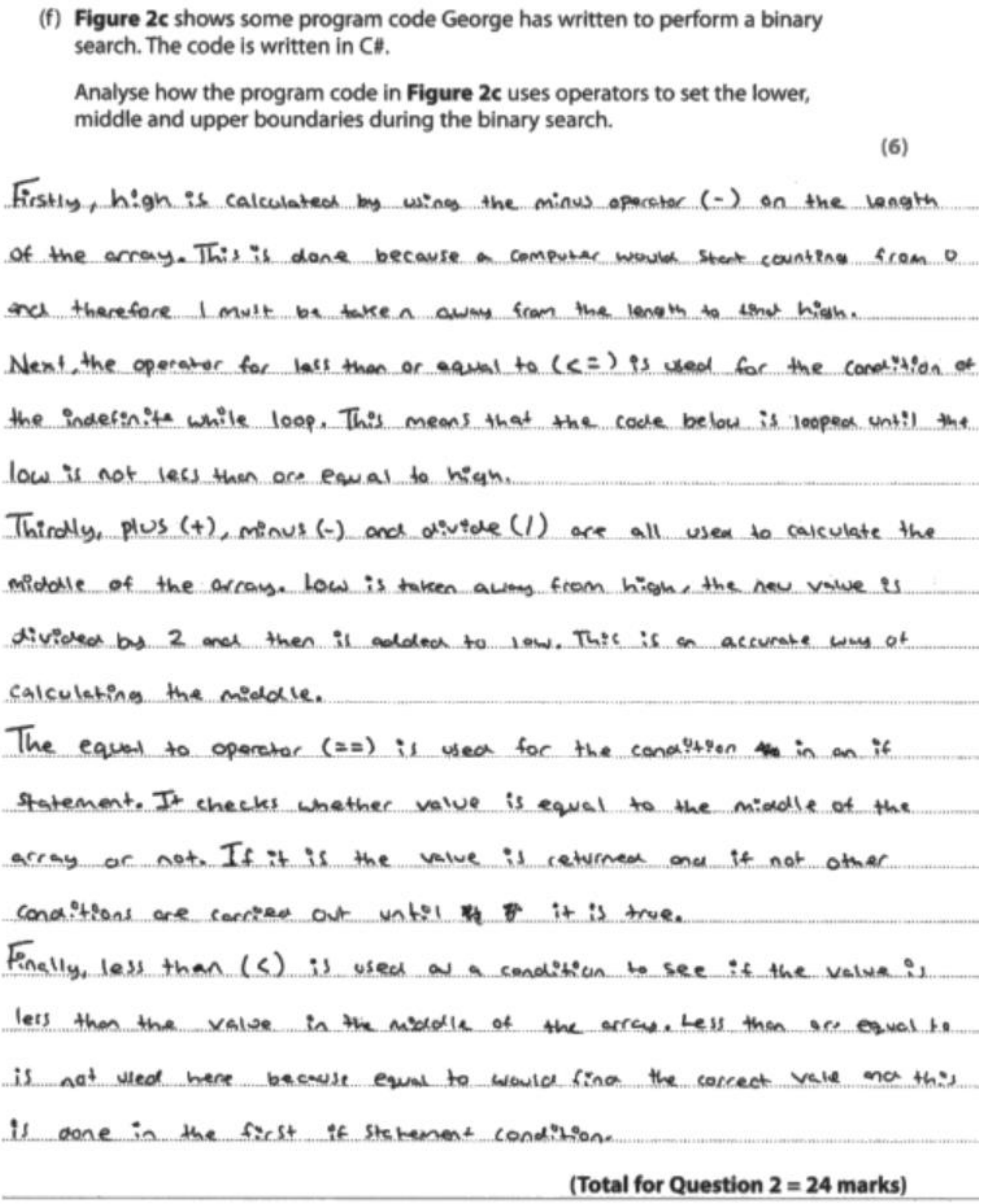
33 (1 Mark)

55 (1 Mark)

62 (1 Mark)

Question 2f (6 Marks)

This question was generally not well answered, the question required an analysis of how operators were used to set boundaries. The majority of learners simply described a binary search without reference to the operators, therefore they were placed in the lower mark range.



(f) **Figure 2c** shows some program code George has written to perform a binary search. The code is written in C#.

Analyse how the program code in **Figure 2c** uses operators to set the lower, middle and upper boundaries during the binary search.

(6)

Firstly, high is calculated by using the minus operator (-) on the length of the array. This is done because a computer would start counting from 0 and therefore 1 must be taken away from the length to that high.

Next, the operator for less than or equal to (<=) is used for the condition of the indefinite while loop. This means that the code below is looped until the low is not less than ore equal to high.

Thirdly, plus (+), minus (-) and divide (/) are all used to calculate the middle of the array. low is taken away from high, the new value is divided by 2 and then is added to low. This is an accurate way of calculating the middle.

The equal to operator (==) is used for the condition in an if statement. It checks whether value is equal to the middle of the array or not. If it is the value is returned and if not other conditions are carried out until it is true.

Finally, less than (<) is used as a condition to see if the value is less than the value in the middle of the array. Less than are equal to is not used here because equal to would find the correct vale and this is done in the first if statement condition.

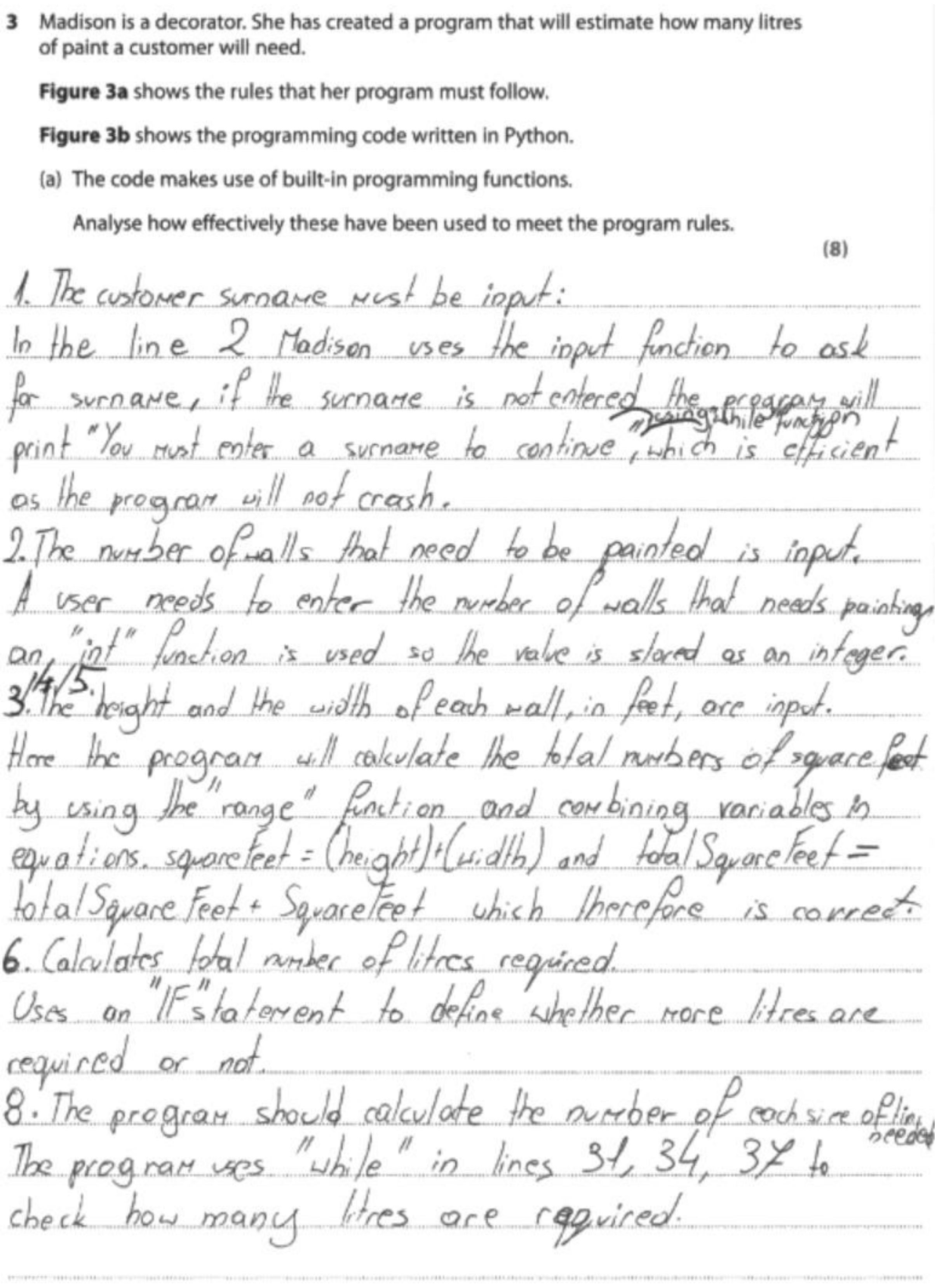
(Total for Question 2 = 24 marks)

6 Marks awarded for:

This answer was placed at the top of the range. The learner demonstrates accurate and detailed knowledge, the answer refers to the operators specifically and accurately describes how they used in the context of the question.

Question 3a (8 Marks)

Most learners achieved only the lower end of the mark range for this question. The majority of answers showed a very limited understanding of built-in-functions and the use of them. Frequently learners would attempt a description of code in general or described some features of the program. Such answers gained little credit.



3 Madison is a decorator. She has created a program that will estimate how many litres of paint a customer will need.

Figure 3a shows the rules that her program must follow.

Figure 3b shows the programming code written in Python.

(a) The code makes use of built-in programming functions.

Analyse how effectively these have been used to meet the program rules.

(8)

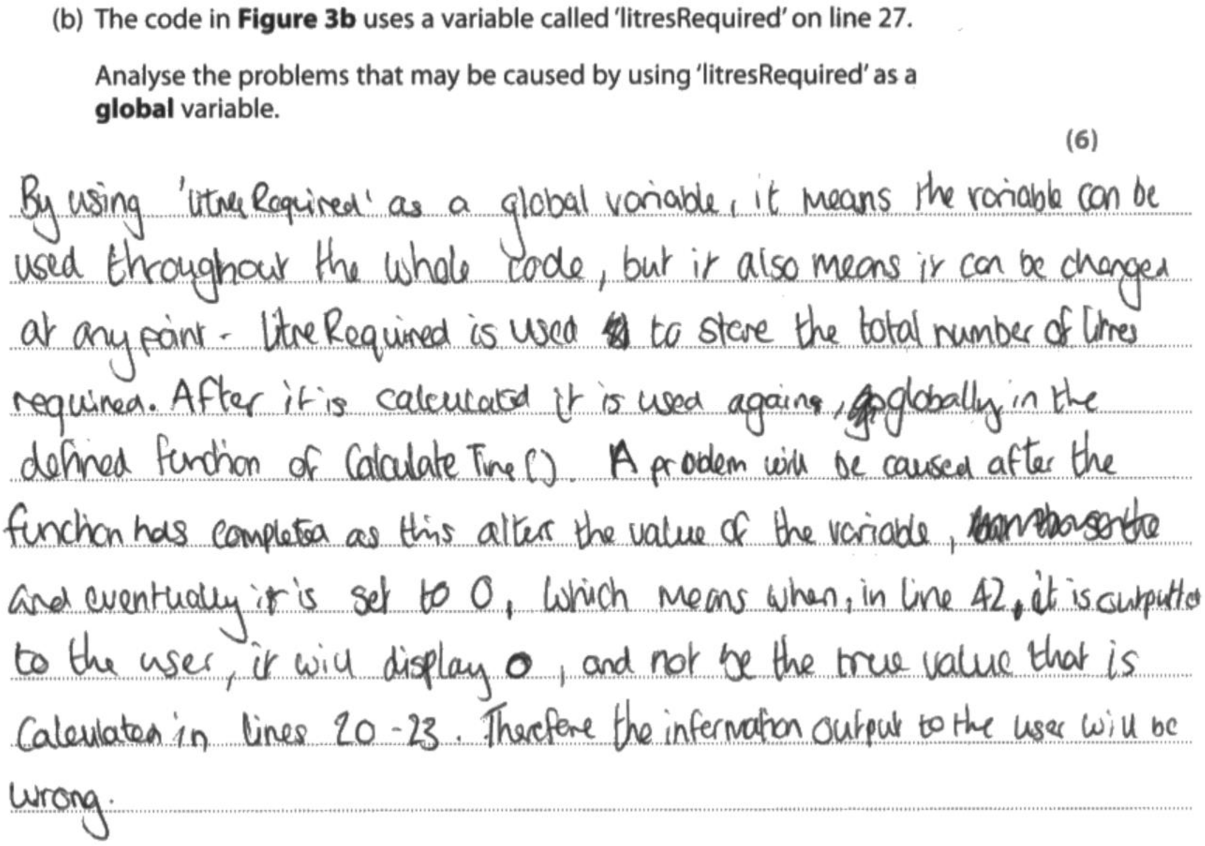
1. The customer surname must be input:
In the line 2 Madison uses the input function to ask for surname, if the surname is not entered "using a while function the program will print "You must enter a surname to continue", which is efficient as the program will not crash.
2. The number of walls that need to be painted is input.
A user needs to enter the number of walls that needs painting an "int" function is used so the value is stored as an integer.
3/4/5. The height and the width of each wall, in feet, are input.
Here the program will calculate the total numbers of square feet by using the "range" function and combining variables to equations. squareFeet = (height)+(width) and totalSquareFeet = totalSquareFeet + SquareFeet which therefore is correct.
6. Calculates total number of litres required.
Uses an "IF"statement to define whether more litres are required or not.
8. The program should calculate the number of each size of lin needed
The program uses "while" in lines 31, 34, 37 to check how many litres are reqquired.

4 Marks awarded for:

- Lower mark band 2 answer.
- Demonstrates some accurate knowledge and understanding, with few minor omissions/any gaps or omissions are minor
- Breaks the situation down into component parts and some of the points made will be relevant to the context in the question
- Displays a partially developed analysis which considers some interrelationships or linkages but not always sustained
- The learner has written about some basic functions such as the input, integer and range functions these are linked to the scenario and a description of how they are used is given however an analysis of how these meet the program rules is not well developed. rules

Question 3b (6 Marks)

Most learners picked up some marks on this question with a lot of learners scoring marks in bands 1 and 2. However leaners often simply described the difference between global and local variables and did not analyse the resulting problems in much detail.



(b) The code in **Figure 3b** uses a variable called 'litresRequired' on line 27.

Analyse the problems that may be caused by using 'litresRequired' as a **global** variable.

(6)

By using 'litreRequired' as a global variable, it means the variable can be used throughout the whole code, but it also means it can be changed at any point. litreRequired is used to store the total number of litres required. After it is calculated it is used again, globally in the defined function of CalculateTime(). A problem will be caused after the function has completed as this alters the value of the variable, and eventually it is set to 0, which means when, in line 42, it is outputted to the user, it will display 0, and not be the true value that is calculated in lines 20-23. Therefore the information output to the user will be wrong.

4 Marks awarded for:

The answer is placed at the top of MB 2, the answer is related to the issues cause in the program, the learner has described the effect of using the variable globally and has realised that the final output would be wrong.

Demonstrates some accurate knowledge and understanding, with few minor omissions/any gaps or omissions are minor

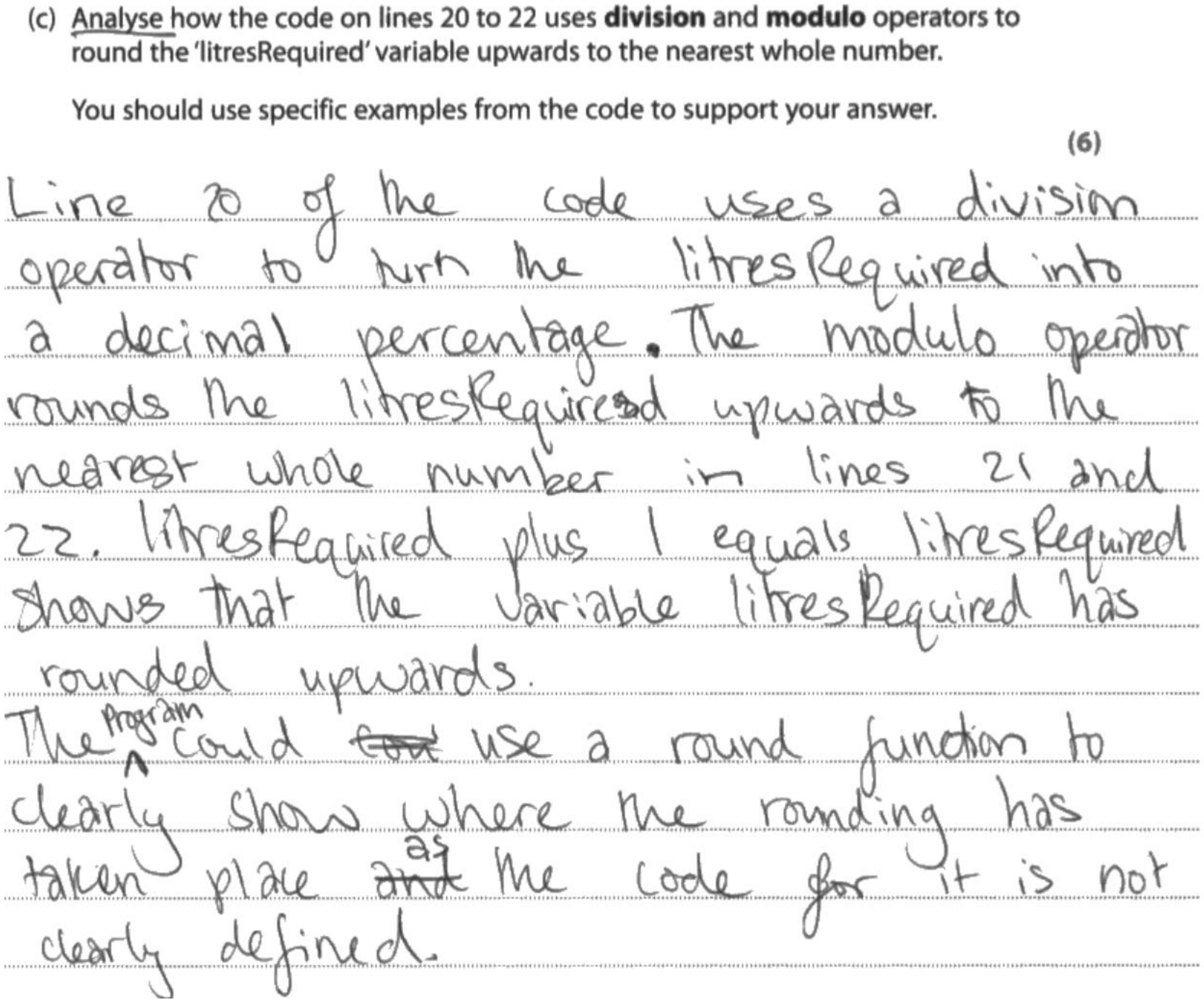
Breaks the situation down into component parts and some of the points made will be relevant to the context in the question

Displays a partially developed analysis which considers some interrelationships or linkages but not always sustained.

Question 3c (6 Marks)

Most learners were placed in mark band 1 or mark band 2 for this question. The majority of learners simply described the program very few students managed to link this to the scenario and how the two types of division are used which is needed to achieve full marks.

Learners showed an awareness of what was happening and were able to relate this to the need to work in whole tins of paint, however the way this was achieved using the two types of division was not linked to this in many answers.



(c) Analyse how the code on lines 20 to 22 uses **division** and **modulo** operators to round the 'litresRequired' variable upwards to the nearest whole number.

You should use specific examples from the code to support your answer.

(6)

Line 20 of the code uses a division operator to turn the litresRequired into a decimal percentage. The modulo operator rounds the litresRequired upwards to the nearest whole number in lines 21 and 22. litresRequired plus 1 equals litresRequired shows that the variable litresRequired has rounded upwards.

The Program could ~~tool~~ use a round function to clearly show where the rounding has taken place ~~and~~ as the code for it is not clearly defined.

3 Marks awarded for:

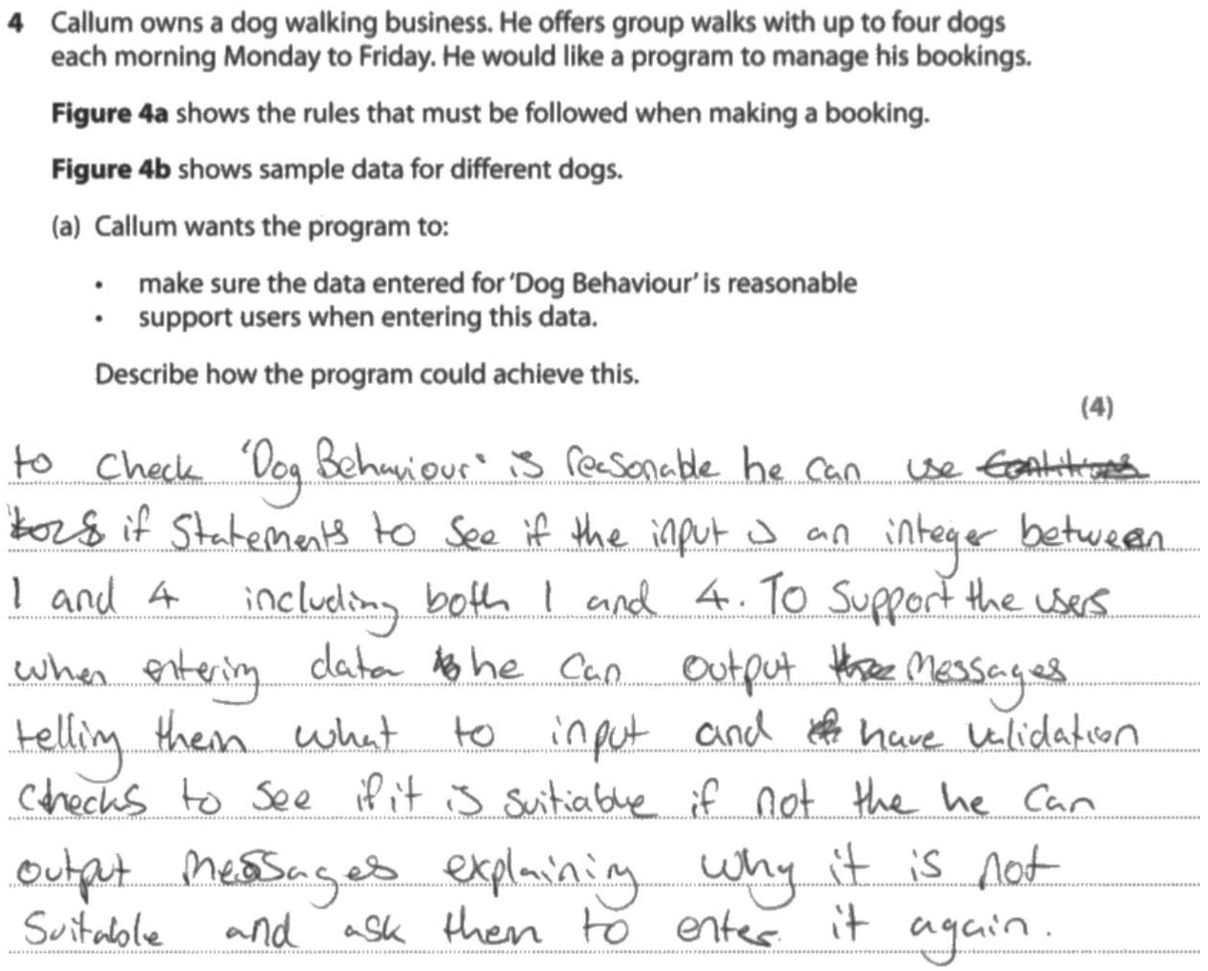
The learner is aware that the number of litres needs to be made into a whole number and rounded upwards. However, the answer does not demonstrate full understanding of how the two types of division are used to achieve this. The answer does however use named lines of code to illustrate the point made.

- Demonstrates some accurate knowledge and understanding, with few minor omissions/any gaps or omissions are minor
- Breaks the situation down into component parts and some of the points made will be relevant to the context in the question
- Displays a partially developed analysis which considers some interrelationships or linkages but not always sustained.

Question 4a (4 Marks)

Most learners were able achieve 3 marks on this question, most recognised that validation checks were needed and therefore a range check could be used. Error messages were frequently mentioned. However the use of meaningful prompts or other support when entering the data was often missing.

Some candidates misunderstood the question and wrote about ways of assessing dogs behaviour and comparing the types of behaviours shown at each level.



4 Callum owns a dog walking business. He offers group walks with up to four dogs each morning Monday to Friday. He would like a program to manage his bookings.

Figure 4a shows the rules that must be followed when making a booking.

Figure 4b shows sample data for different dogs.

(a) Callum wants the program to:

- make sure the data entered for 'Dog Behaviour' is reasonable
- support users when entering this data.

Describe how the program could achieve this.

(4)

to Check 'Dog Behaviour' is reasonable he can use ~~Conditions~~ ~~tors~~ & if Statements to See if the input is an integer between 1 and 4 including both 1 and 4. To Support the users when entering data ~~to~~ he Can output ~~these~~ messages telling them what to input and ~~it~~ have validation checks to see if it is suitiable if not the he can output messages explaining why it is not Suitable and ask them to enter it again.

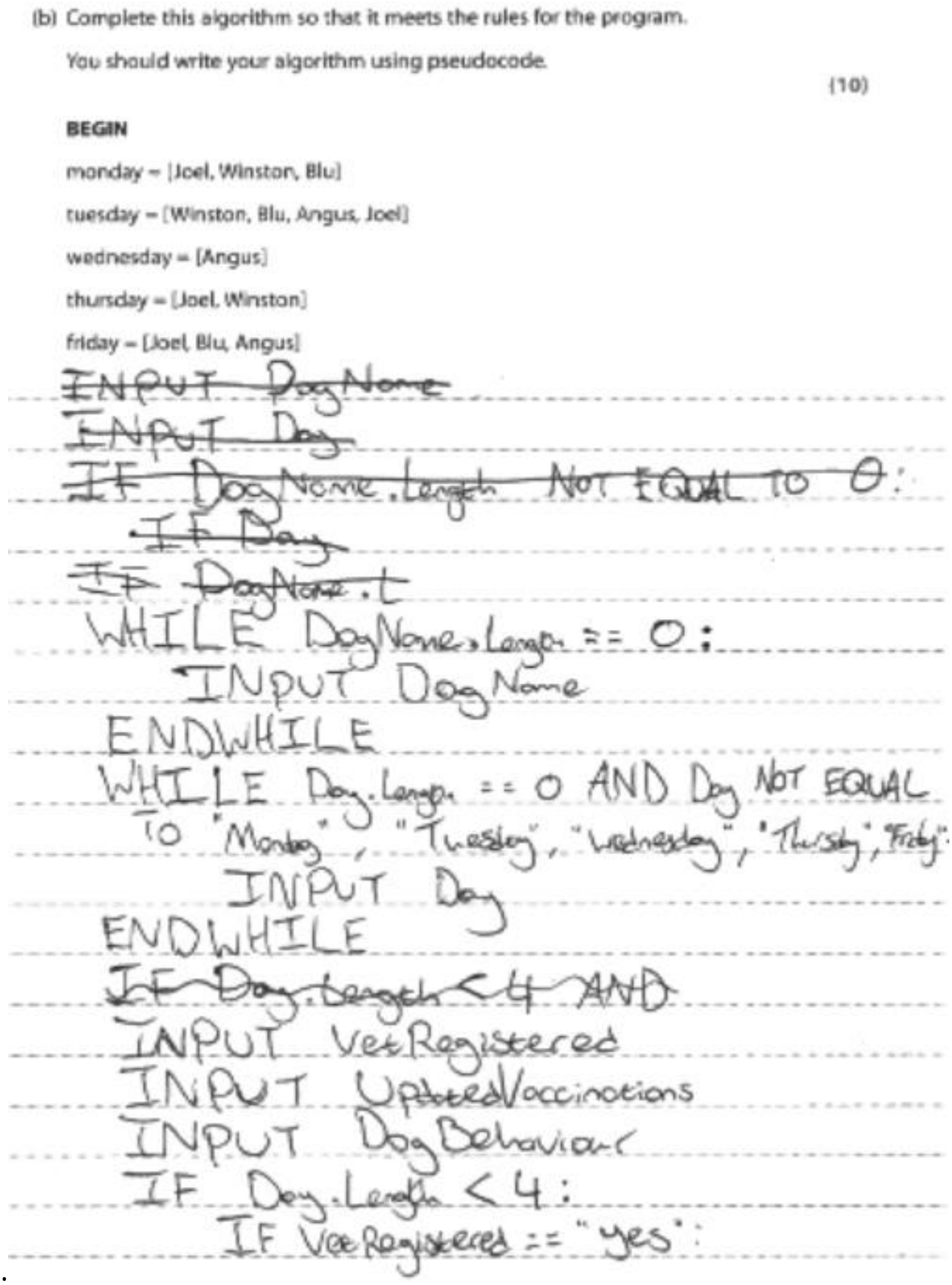
4 Marks awarded for:

- uses validation (1 Mark)
- check data entered meets set criteria (1 Mark)
- provide meaningful error messages (1 Mark)
- provide meaningful prompts (1 Mark)

Question 4b (10 Marks)

On the whole, this question was not very well answered. Many learners started by repeating the code given in the question. Most answers checked the number of dogs already booked on any one day, however a common weakness was not asking the user for the day they required so that this could be checked to see if it was full.

Many answers consisted of a series of IF statements, however the flow between them and the outputs were often not correct logically. Inefficient algorithms were commonly seen.

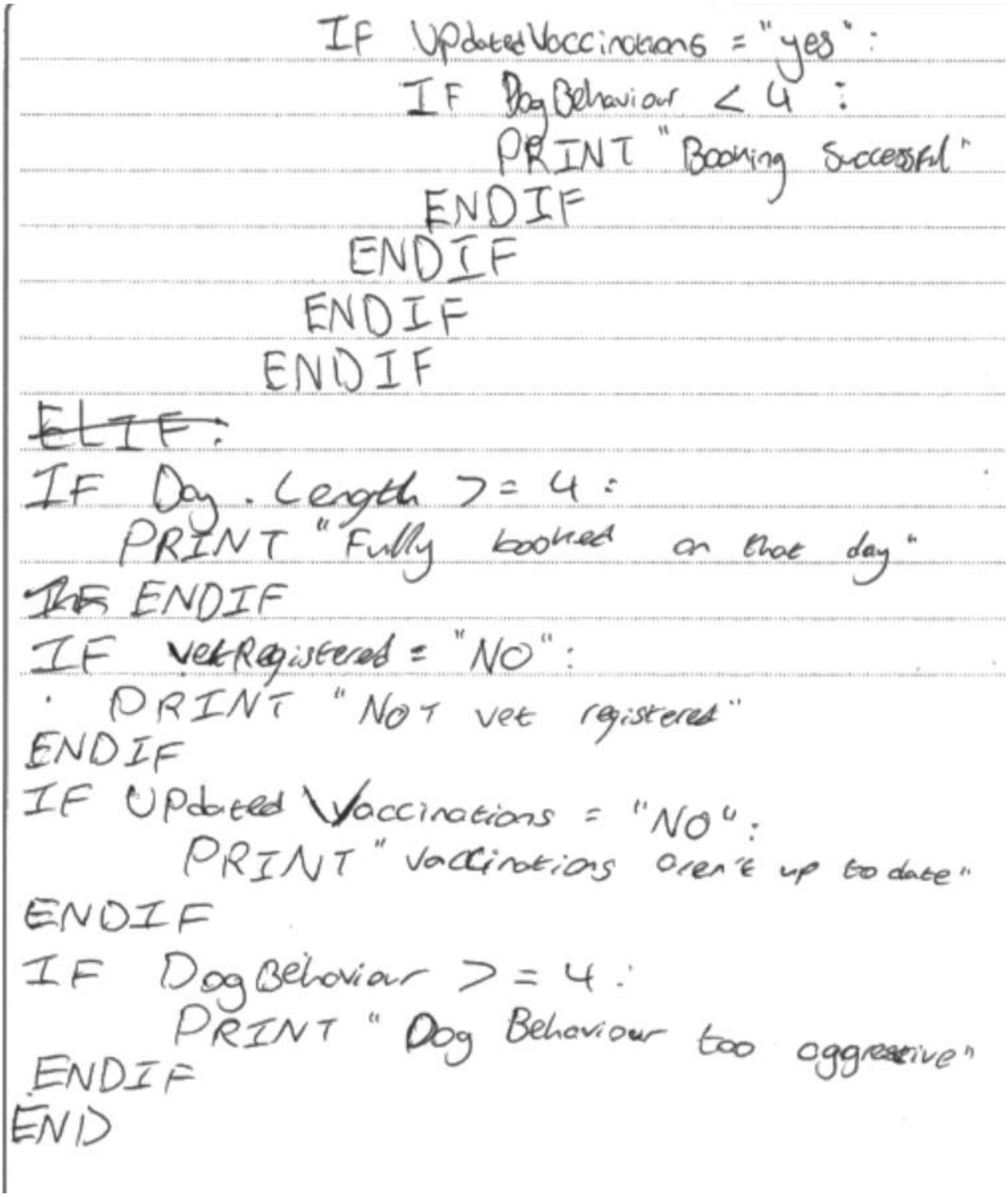


(b) Complete this algorithm so that it meets the rules for the program.

You should write your algorithm using pseudocode.

(10)

```
BEGIN
monday = [Joel, Winston, Blu]
tuesday = [Winston, Blu, Angus, Joel]
wednesday = [Angus]
thursday = [Joel, Winston]
friday = [Joel, Blu, Angus]
~~INPUT DogName~~
~~INPUT Day~~
~~IF DogName.Length NOT EQUAL TO 0:~~
    ~~IF Day~~
~~IF DogName.L~~
 WHILE DogName.Length == 0:
      INPUT DogName
  ENDWHILE
  WHILE Day.Length == 0 AND Day NOT EQUAL
    TO "Monday", "Tuesday", "Wednesday", "Thursday", "Friday":
        INPUT Day
  ENDWHILE
  ~~IF Day.Length < 4 AND~~
   INPUT VetRegistered
   INPUT UpdatedVaccinations
   INPUT DogBehaviour
   IF Day.Length < 4:
       IF VetRegistered == "yes":
```



```
                IF UpdatedVaccinations = "yes" :
                    IF DogBehaviour < 4 :
                        PRINT "Booking Successful"
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
~~ELIF:~~
IF Day.Length >= 4 :
    PRINT "Fully booked on that day"
~~Th~~ ENDIF
IF vetRegistered = "NO" :
    PRINT "Not vet registered"
ENDIF
IF Updated Vaccinations = "NO" :
        PRINT "Vaccinations aren't up to date"
ENDIF
IF DogBehaviour >= 4 :
        PRINT "Dog Behaviour too aggressive"
ENDIF
END
```

7 Marks awarded for:

The response is an almost complete but inefficient solution. The use of logical operators is mostly accurate. The program would achieve the correct outcome. However, the solution is inefficient in that error messages are produced in a separate set of IF statements.

The response is placed in mark band 2.

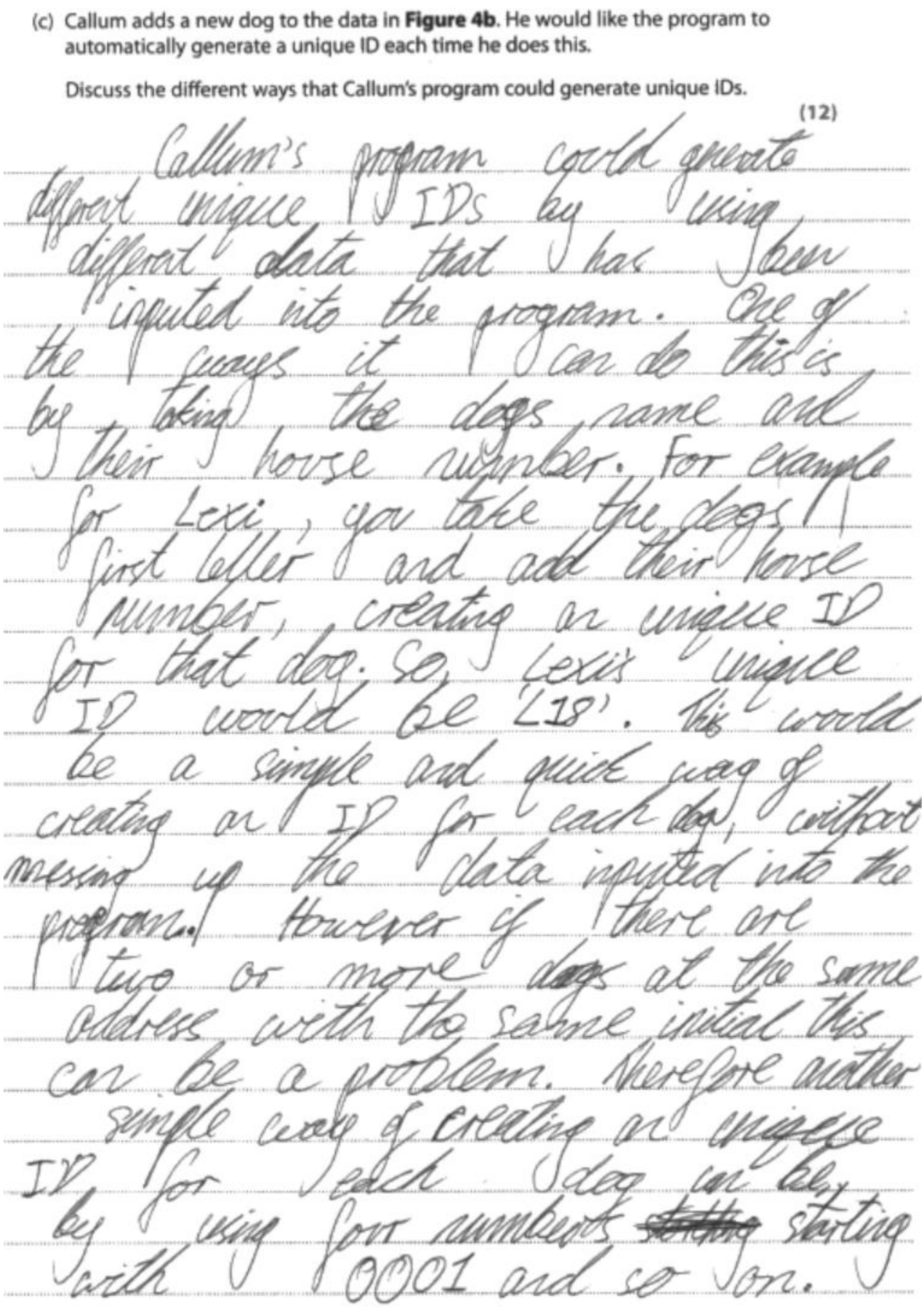
- Structure of the algorithm uses mostly appropriate hierarchies/subdivision to provide some clarity and readability.
- Variable/object/process names are mostly appropriate but there is some inconsistency
- Use of logical operations and sequences/structure are mostly accurate with only minor errors.
- Accepted conventions have been applied but there are some inconsistencies.
- An almost complete/inefficient solution has been achieved.

Question 4c (12 Marks)

Most learners were placed in mark band one or two for this question. Most learners were able to suggest one or more ways of producing a unique ID, the data provided had two dogs with the same name to provide a basis for an example. Many learners spotted this and were able to use it as an example.

Common answers included, using a sequential number, using a random number, using data from other fields concatenated to produce a unique code of some type.

Learners often describe these methods; however they did not then go on to consider the viability of these methods as the database increased in size.

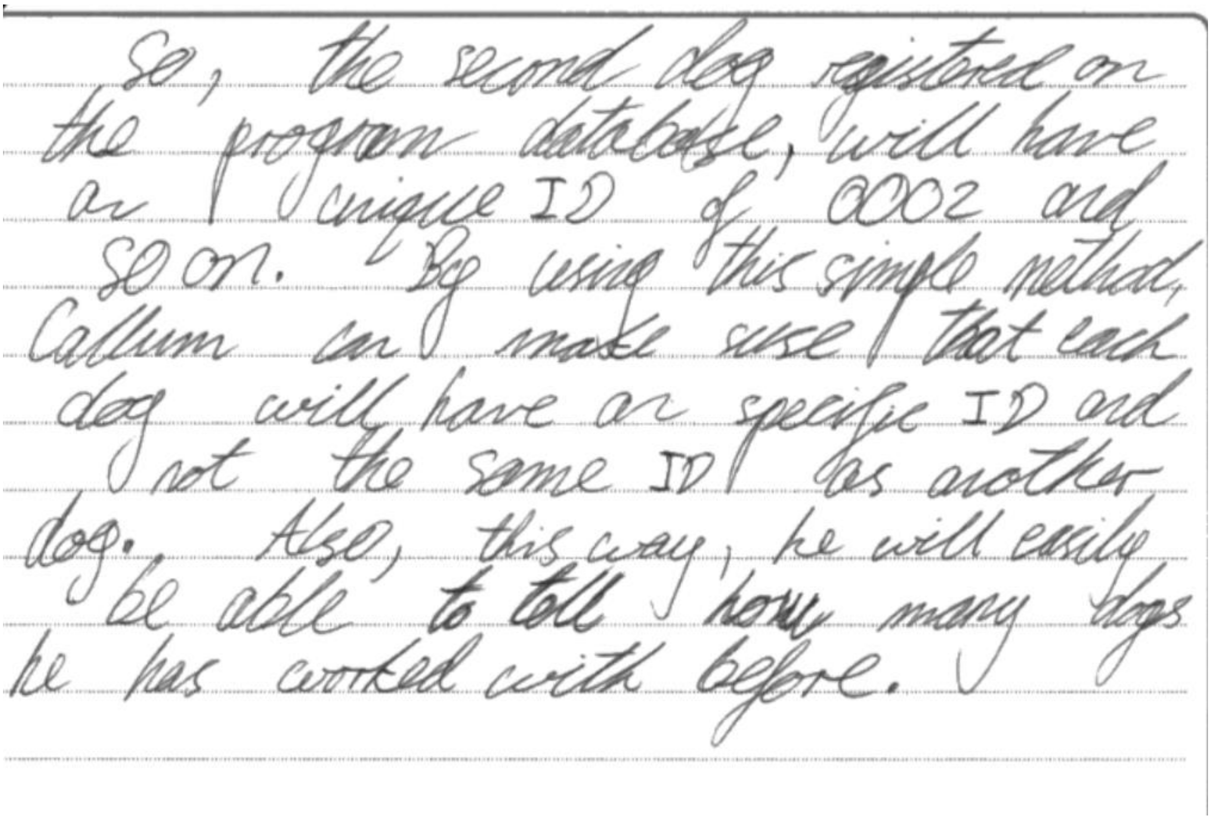


(c) Callum adds a new dog to the data in **Figure 4b**. He would like the program to automatically generate a unique ID each time he does this.

Discuss the different ways that Callum's program could generate unique IDs.

(12)

Callum's program could generate different unique IDs by using different data that has been inputed into the program. One of the ways it can do this is by taking the dogs name and their house number. For example for Lexi, you take the dogs first letter and add their house number, creating an unique ID for that dog. So Lexi's unique ID would be 'L18'. This would be a simple and quick way of creating an ID for each dog, without messing up the data inputted into the program. However if there are two or more dogs at the same address with the same initial this can be a problem. Therefore another simple way of creating an unique ID for each dog can be, by using four numbers ~~starting~~ starting with 0001 and so on.



So, the second dog registered on the program database, will have an unique ID of 0002 and so on. By using this simple method, Callum can make sure that each dog will have an specific ID and not the same ID as another dog. Also, this way, he will easily be able to tell how many dogs he has worked with before.

8 Marks awarded for:

- The response shows an understanding of different ways in which unique IDs can be created. The discussion is linked to the problem and includes some discussion of how a larger data set would cope with the unique IDs used.
- Demonstrates some accurate knowledge and understanding, with only minor gaps or omissions
- Some of the points made will be relevant to the context in the question, but the link will not always be clear
- Displays a partially developed discussion which considers some different aspects and some consideration of how they interrelate, but not always in a sustained way.

The response is placed in mark band 2.

Summary

Based on performance in this examination series, learners are offered the following advice:

- Ensure that learners make full use of the information booklet when answering the exam questions. When candidates are referred to the information booklet, they should make sure that their answer is specific to the information / program code / rules or other stimulus given.
- For shorter response questions (5 marks or less), learners should be encouraged to note the number of marks available as this will help them identify the number of points they need to make. For example, in a 4 mark 'Explain one…' style question, learners would need to make at least four linked points that expand/exemplify understating of a single point
- When producing extended writing responses (6 marks or more) learners should ensure that they consider a range of points, each of which should be expanded or supported with examples and applied to the given context.
- Develop a better understanding of logic and have more practice at being able to come up with solutions to complex problems. Being able to create effective algorithms will help in all learning aims in this unit. Learners should be familiar with developing algorithms using both flowcharts and Pseudocode.

Llywodraeth Cynulliad Cymru
Welsh Assembly Government

Rewarding Learning

For more information on Pearson qualifications, please visit
http://qualifications.pearson.com/en/home.html

Pearson Education Limited. Registered company number 872828
with its registered office at Edinburgh Gate, Harlow, Essex CM20 2JE