Examiners' Report
June 2018

GCSE Computer Science 1CP1 02

Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications come from Pearson, the UK's largest awarding body. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.edexcel.com or www.btec.co.uk.

Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus.

ResultsPlus

Giving you insight to inform next steps

ResultsPlus is Pearson's free online service giving instant and detailed analysis of your students' exam results.

- See students' scores for every exam question.
- Understand how your students' performance compares with class and national averages.
- Identify potential topics, skills and types of question where students may need to develop their learning further.

For more information on ResultsPlus, or to log in, visit www.edexcel.com/resultsplus.
Your exams officer will be able to set up your ResultsPlus account in minutes via Edexcel Online.

Pearson: helping people progress, everywhere

Pearson aspires to be the world's leading learning company. 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 students at: www.pearson.com/uk.

June 2018
Publications Code 1CP1_02_1806_ER

Introduction

This is the first time that candidates have sat for examinations in this component, which requires them to demonstrate application of computational thinking.

This untiered paper has been specifically designed so that easier questions are presented earlier in the paper, with gradually more challenging questions later on. However, candidates across the ability range will find questions that are both challenging and interesting throughout.

Candidates are introduced to a context for the entire paper. This context is expanded specifically for each question. Unless instructed otherwise, candidates should apply this context when developing responses. Additionally, candidates will find that 'command words' are used consistently in the paper to indicate the type of response expected.

This reformed GCSE has introduced a two-mark question type using the command word 'explain'. These questions are designed to allow candidates to demonstrate the depth of their understanding of a single topic. This requires candidates to give responses in two joined parts. The first part should be a fact that addresses the question. The second part of the link should be an expansion. It is helpful to consider using words such as because of / result of / associated with / reason for / connected to which explicitly show an expansion. Question 6c is an example of this type question.

Examiners saw many detailed responses. However, candidates often provided little more than simple statements and did not include examples and reasons where expansions or explanations were required. It is imperative that in a technical subject, the correct terminology is used and that sufficient detail is given to demonstrate understanding. are reminded to learn subject specific terms and their meanings.

A misunderstanding of the rules of precedence (BIDMAS) resulted in some marks not being awarded. are advised to revise the subject related Mathematics. Fully bracketed expressions ensure that the meaning is very clear.

Some responses indicated confusion between facilities of particular programming languages and the more fundamental concepts associated with programming. One example of this, as indicated in responses, showed confusion between arrays and Python lists. Arrays have static length and single data types; Python lists are dynamic and allow mixed data types.

When a candidate does not want a response to be marked because it has been replaced, the original should be crossed out. When there are two different responses to a question, it is unclear which the candidate intends to be marked. In these cases, the examiner must mark the first response given. This is especially relevant in the case where a response has been written on the question paper and the same question, with a different response, has been answered on an additional sheet.

Question 1 (a)

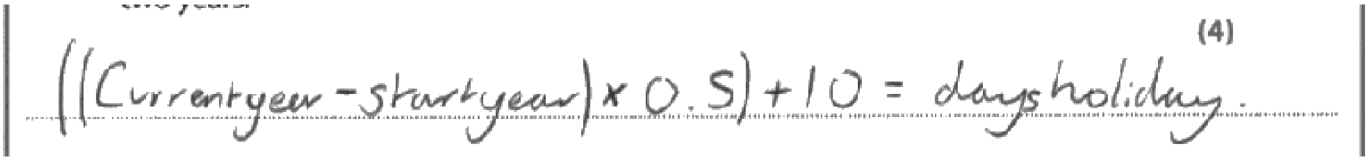
Age was the most frequent response that received no marks. If the age of an employee was to be stored, it would become out of date within a year. A better response would be to store the date of birth, which does not change. In general, a value that can be calculated from stored data is not itself stored.

Question 1 (b)

Full marks were easily awarded to candidates who either followed the rules of BIDMAS or used brackets to indicate order of evaluation. In the cases where this was not possible, candidates usually earned two or three marks.

The most common mistake was leaving out the subtraction of the two years where the extra holiday was not earned.

This response scored 3 marks.



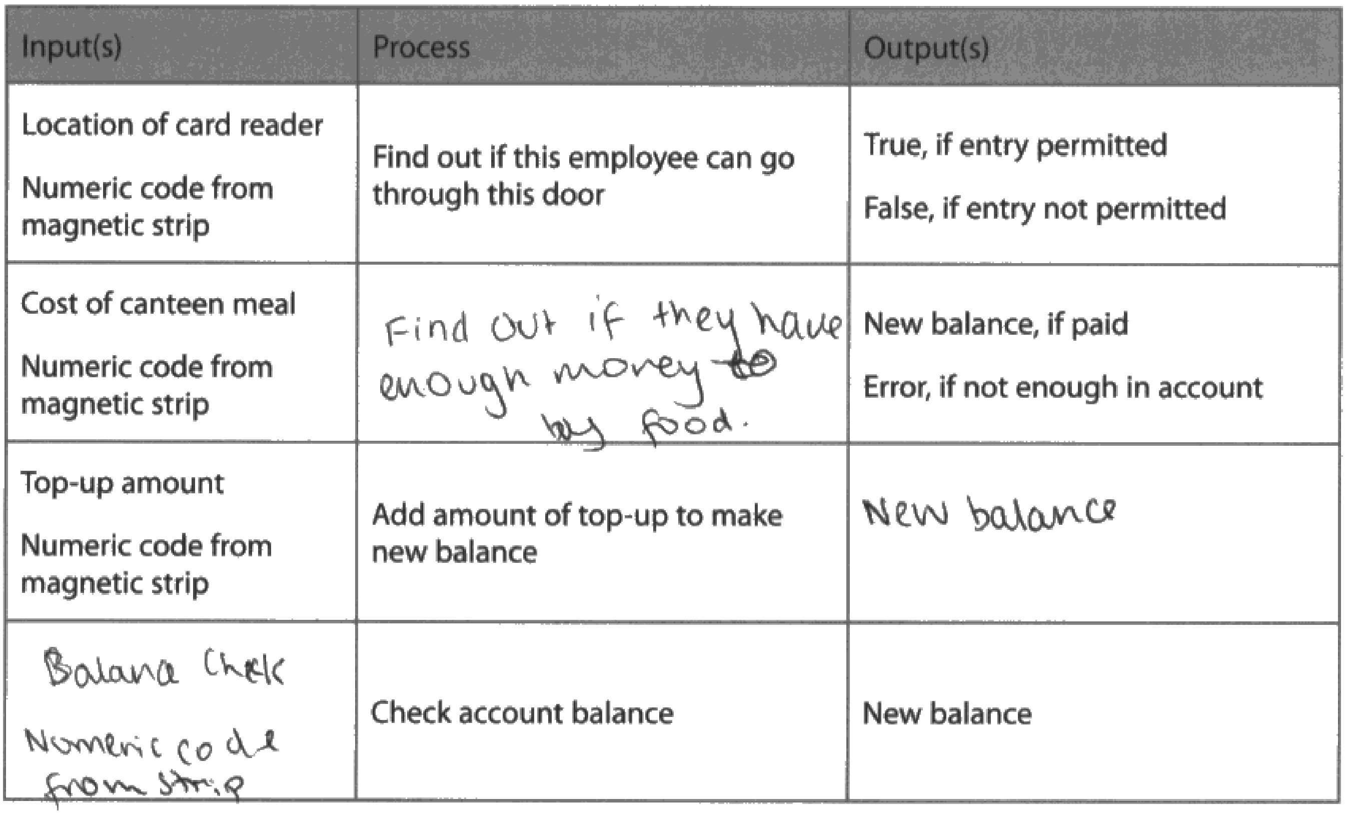
((Currentyear - startyear) x 0.5) + 10 = daysholiday.

(4)

Question 1 (c)

There were many excellent responses to this question.

This response scored 3 marks.



Input(s)	Process	Output(s)
Location of card reader Numeric code from magnetic strip	Find out if this employee can go through this door	True, if entry permitted False, if entry not permitted
Cost of canteen meal Numeric code from magnetic strip	Find out if they have enough money ~~to~~ by food.	New balance, if paid Error, if not enough in account
Top-up amount Numeric code from magnetic strip	Add amount of top-up to make new balance	New balance
Balance Chek Numeric code from strip	Check account balance	New balance

Question 2 (a)

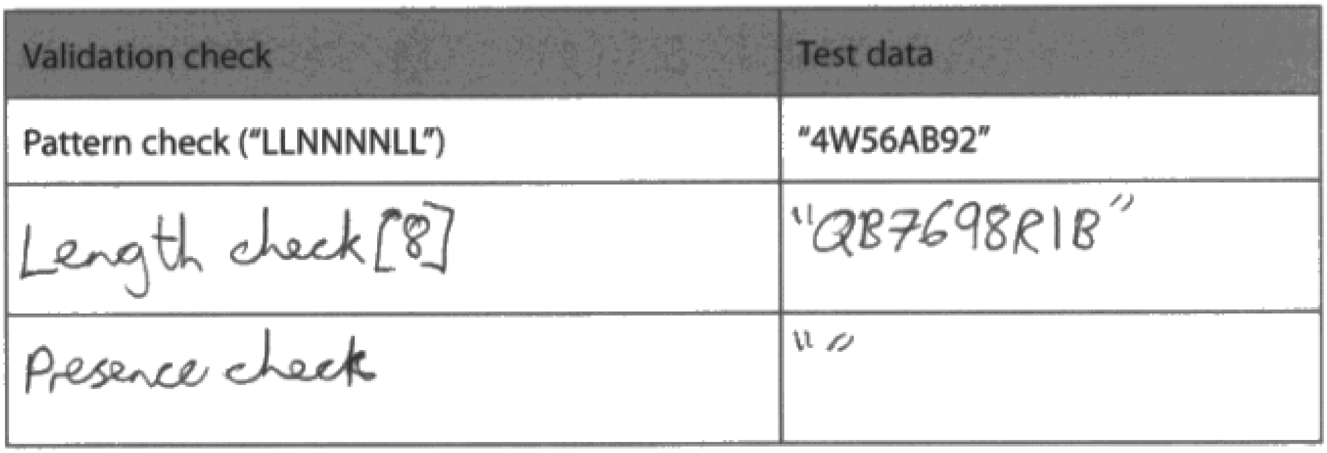
This question was very well answered by candidates. It was good to see that candidates could trace selection.

Question 2 (b)

Length check was the most commonly seen response. However, sometimes candidates supplied test data that had a length of eight characters. The test data should be caught by the validation check, not pass the validation check.

Presence check, which is correct, was often seen as a type of validation. However, the test data provided did not always match. The test data for ‘presence check’ should be an empty string, such as “”, null, or the word ‘nothing’. Many candidates supplied “ “, a string of spaces, which has length and content. This was awarded, but responses should be accurate in the context of input validation.

This response scored 4 marks.



Validation check	Test data
Pattern check (“LLNNNNLL”)	“4W56AB92”
Length check [8]	"QB7698R1B"
Presence check	" "

Question 3 (a) (i)

Candidates responded well to this question. However, they are reminded that to show full understanding, especially of blocked constructs, identification of the full block is required, not just the first keyword.

Question 3 (a) (ii)

Candidates responded well to this question.

Question 3 (a) (iii)

Candidates responded well to this question.

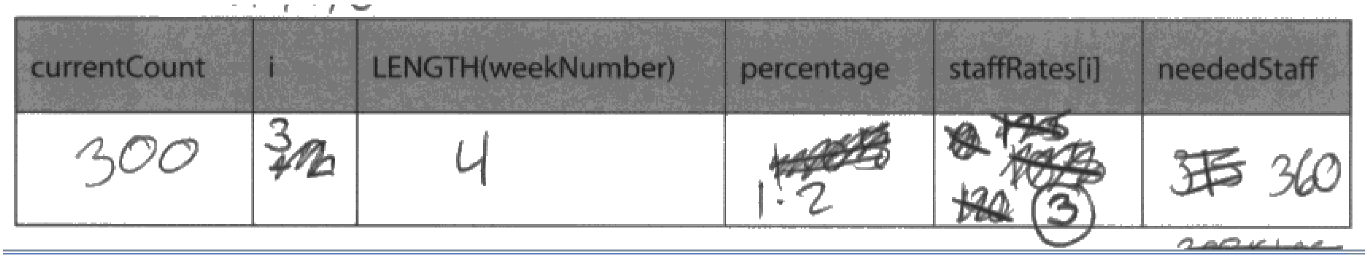
Question 3 (b)

A full range of marks was awarded for this question.

Rounding percentage to one decimal place (i.e. 1.2) could not be awarded. However, a follow through was allowed in the calculation of needed Staff.

The calculation for percentage, done in the machine, would not be rounded up before it was used in a following calculation. A programmer would have to instruct the machine to round up before carrying out further arithmetic.

This response scored 2 marks.



currentCount	i	LENGTH(weekNumber)	percentage	staffRates[i]	neededStaff
300	3 ~~[illegible]~~	4	~~[illegible]~~ 1.2	~~[illegible]~~ ③	~~375~~ 360

Question 3 (c)

Candidates responded well to this question.

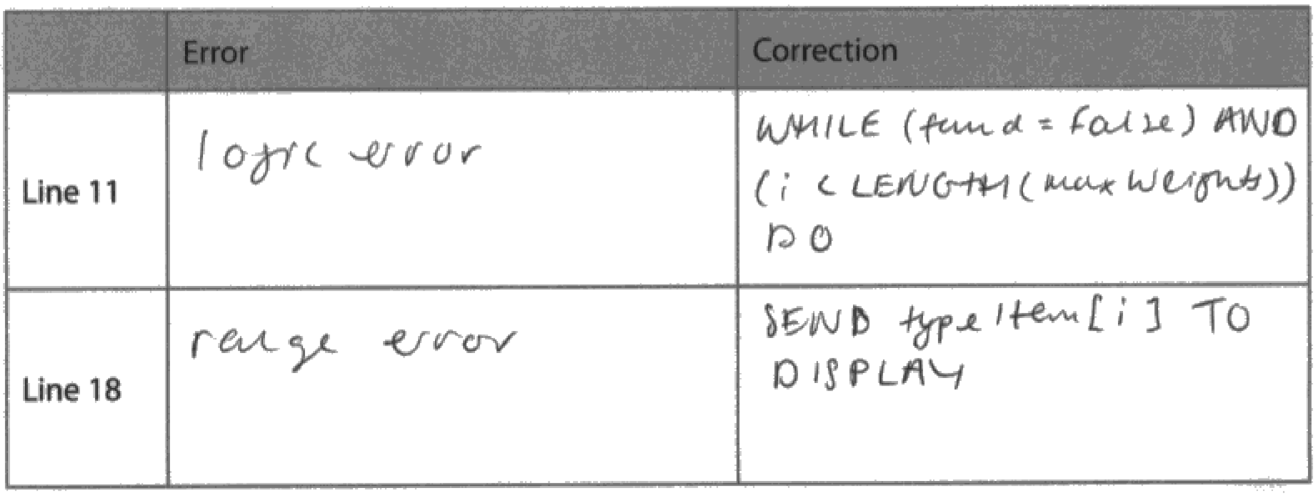
Question 4 (a)

Candidates responded well to this question.

Question 4 (b)

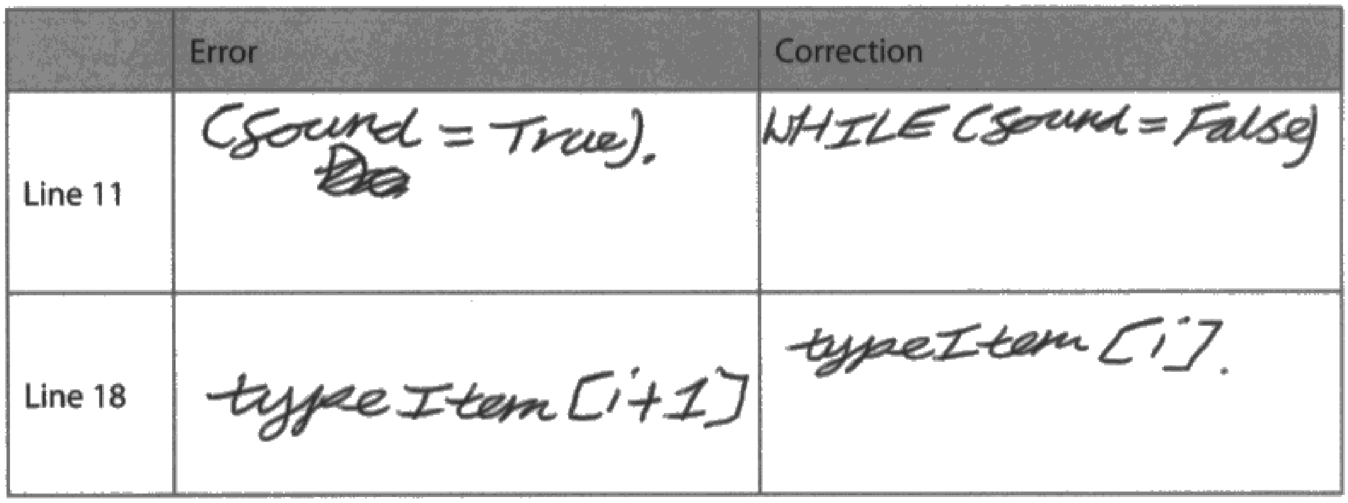
A full range of marks was awarded for this question. Some candidates chose to give the type of error for column one, rather than describe the actual error.

This response scored 2 marks.



	Error	Correction
Line 11	logic error	WHILE (found = false) AND (i < LENGTH(maxWeights)) DO
Line 18	range error	SEND typeItem[i] TO DISPLAY

It was only necessary to indicate the actual error and a correction.



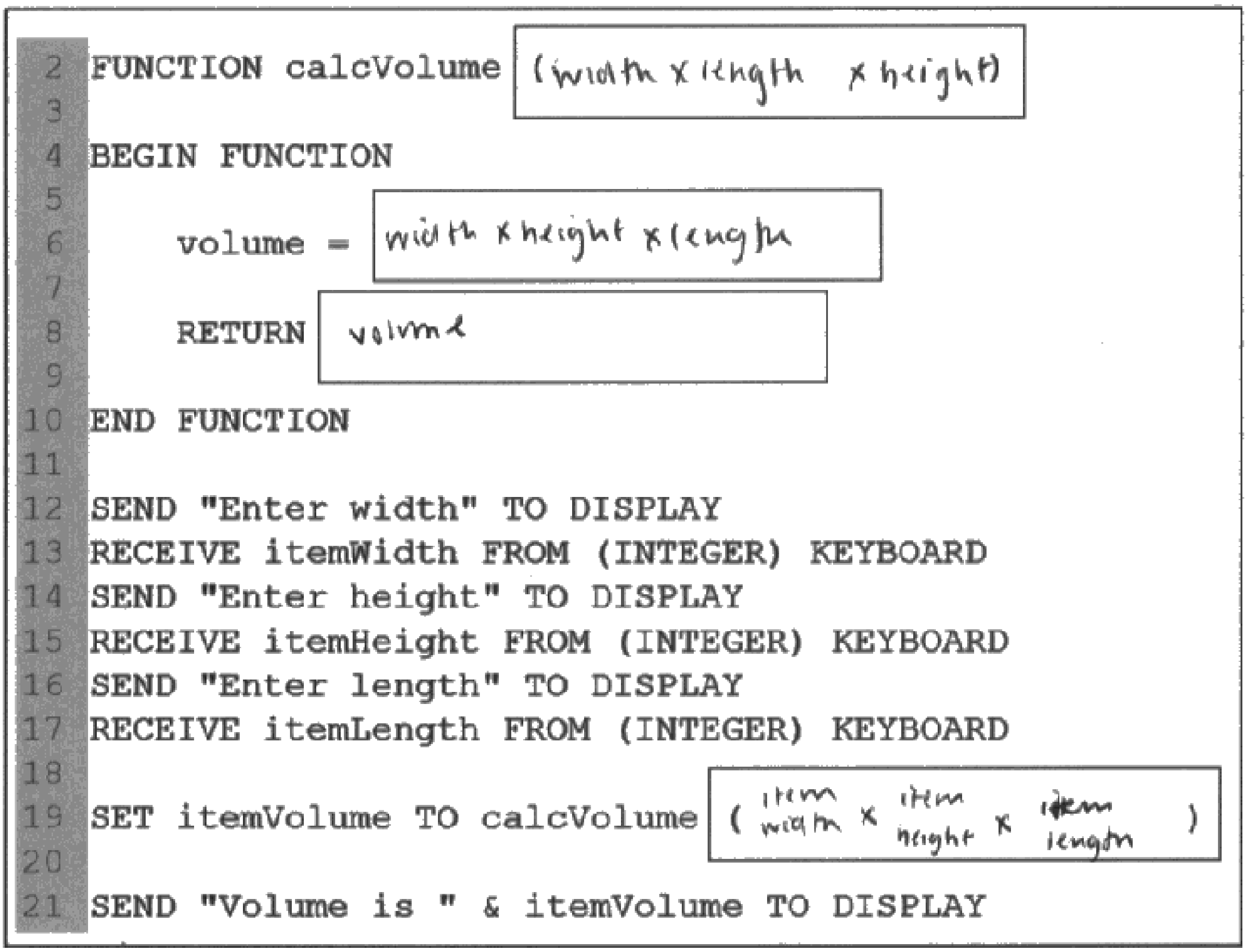
	Error	Correction
Line 11	(found = True). ~~Do~~	WHILE (found = False)
Line 18	typeItem[i+1]	typeItem[i].

Question 5 (a)

A wide range of marks was awarded for this question. A question about subprograms and parameter passing has appeared in almost all papers since the initial series.

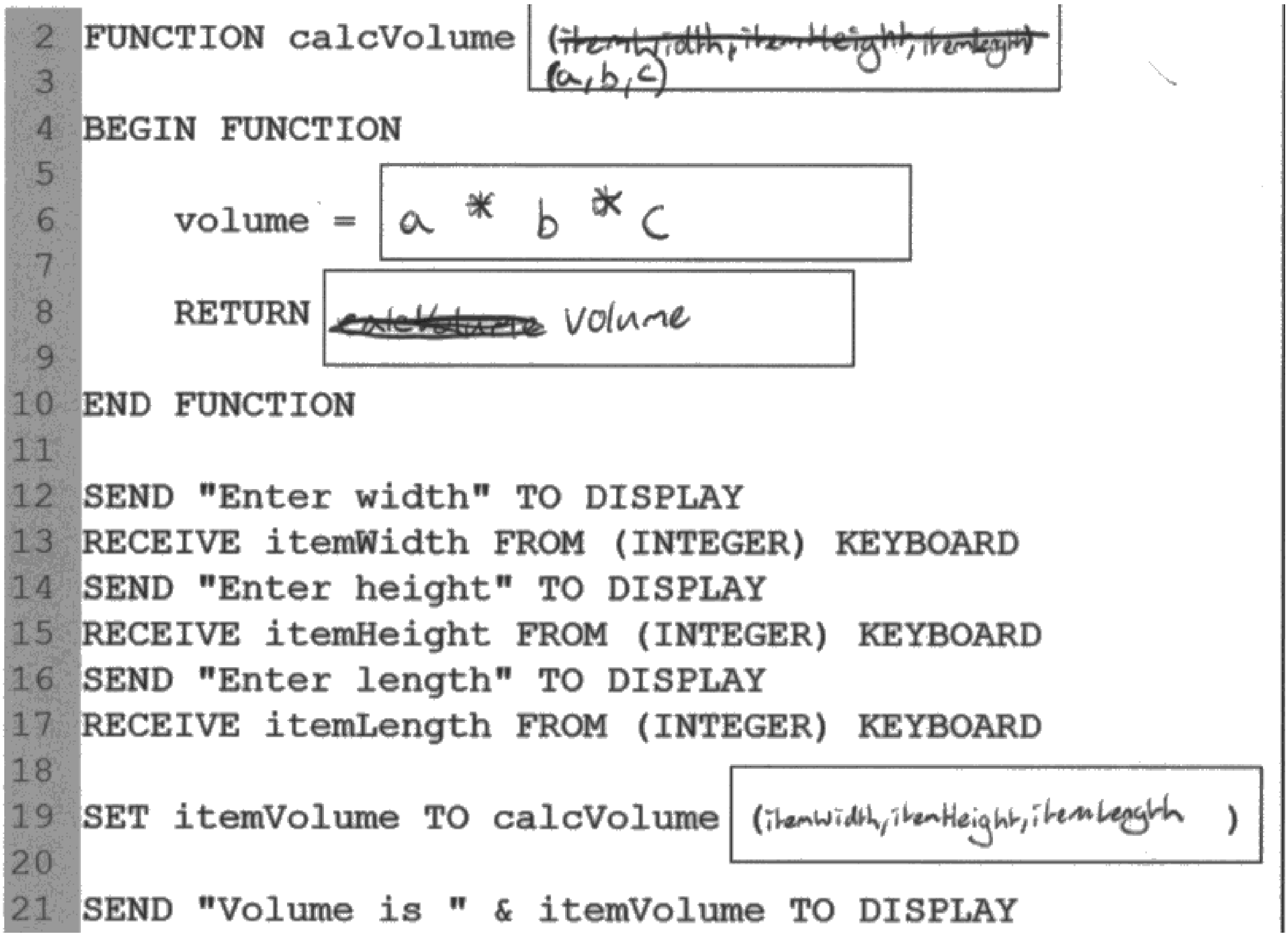
The most common mistake with parameter passing involved the use of multiplication. This example was awarded 1 mark for line 6, as the names of the variables matched the input parameters, and 1 mark for line 8, as the return value is correct. However, neither line 2 nor line 19 could be awarded because of the use of ‘x’ in the parameter list.

This response scored 2 marks.



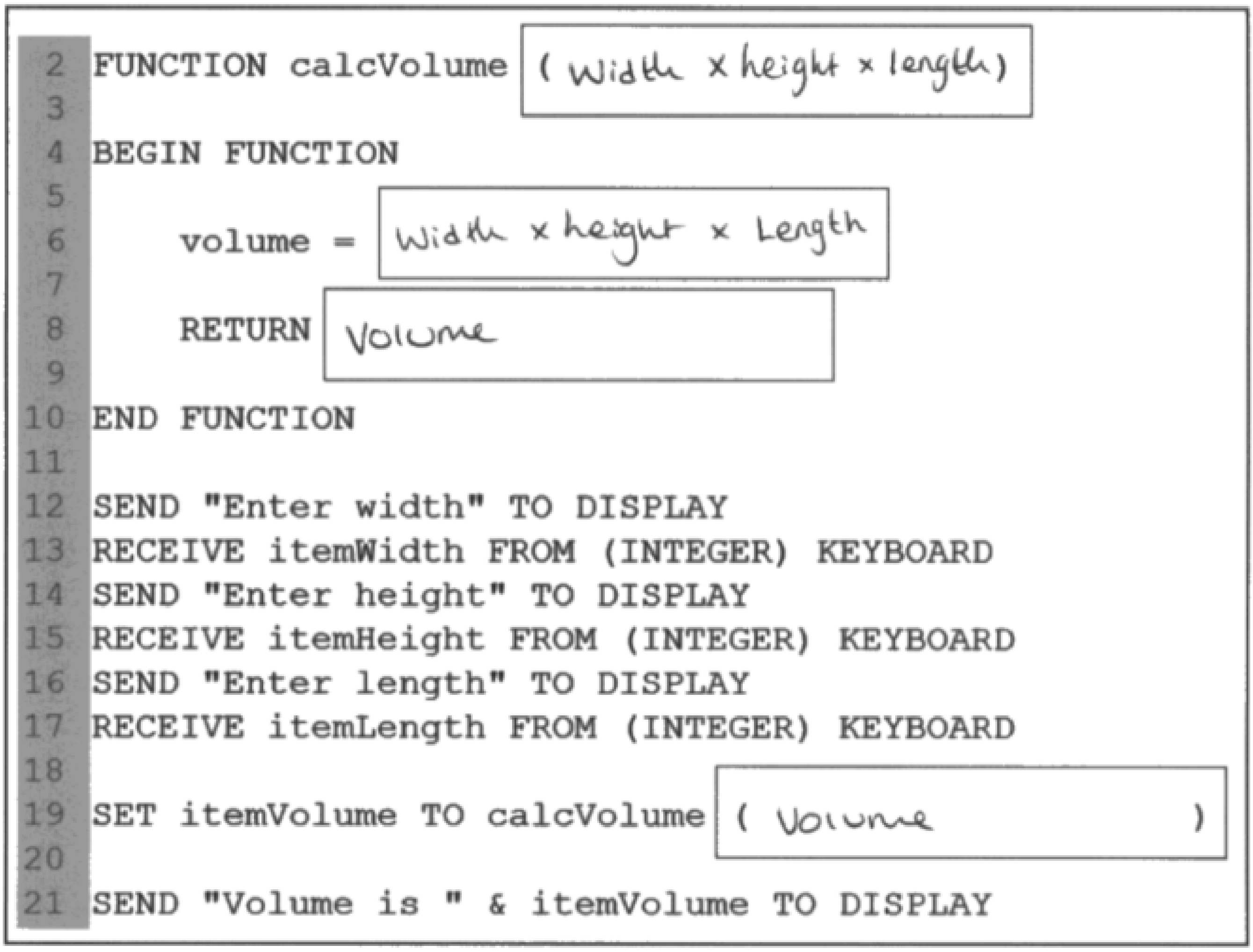
```
 2 FUNCTION calcVolume (width x length x height)
 3
 4 BEGIN FUNCTION
 5
 6     volume = width x height x length
 7
 8     RETURN volume
 9
10 END FUNCTION
11
12 SEND "Enter width" TO DISPLAY
13 RECEIVE itemWidth FROM (INTEGER) KEYBOARD
14 SEND "Enter height" TO DISPLAY
15 RECEIVE itemHeight FROM (INTEGER) KEYBOARD
16 SEND "Enter length" TO DISPLAY
17 RECEIVE itemLength FROM (INTEGER) KEYBOARD
18
19 SET itemVolume TO calcVolume ( item width x item height x item length )
20
21 SEND "Volume is " & itemVolume TO DISPLAY
```

In this example, the transition from main line variables to local parameters is very clear. The correct notation has been followed. The response scored 4 marks.



```
FUNCTION calcVolume (~~itemWidth, itemHeight, itemLength~~
                    (a,b,c)
BEGIN FUNCTION

    volume = a * b * c

    RETURN ~~calcVolume~~ volume

END FUNCTION

SEND "Enter width" TO DISPLAY
RECEIVE itemWidth FROM (INTEGER) KEYBOARD
SEND "Enter height" TO DISPLAY
RECEIVE itemHeight FROM (INTEGER) KEYBOARD
SEND "Enter length" TO DISPLAY
RECEIVE itemLength FROM (INTEGER) KEYBOARD

SET itemVolume TO calcVolume (itemWidth, itemHeight, itemLength )

SEND "Volume is " & itemVolume TO DISPLAY
```

Two marks were awarded for this example because the variable names (line 6) match those on line 2 and line 8 has the correct return value. No marks were awarded for the parameters themselves on line 2 due to the multiplication sign, which was a very commonly seen error. The call on line 19 is not correct. This response scored 2 marks.



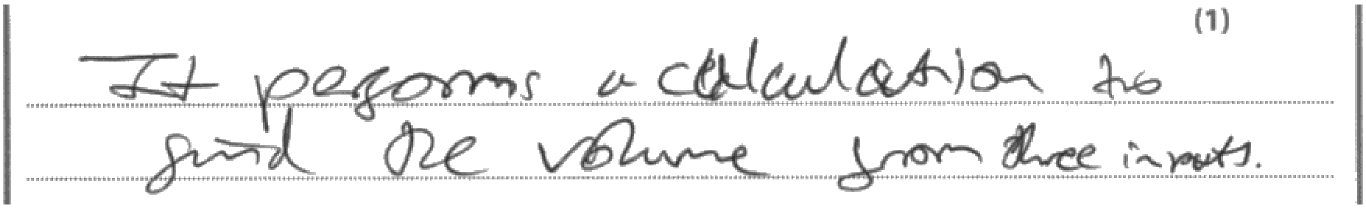
```
 2 FUNCTION calcVolume ( Width x height x length)
 3
 4 BEGIN FUNCTION
 5
 6     volume = Width x height x Length
 7
 8     RETURN Volume
 9
10 END FUNCTION
11
12 SEND "Enter width" TO DISPLAY
13 RECEIVE itemWidth FROM (INTEGER) KEYBOARD
14 SEND "Enter height" TO DISPLAY
15 RECEIVE itemHeight FROM (INTEGER) KEYBOARD
16 SEND "Enter length" TO DISPLAY
17 RECEIVE itemLength FROM (INTEGER) KEYBOARD
18
19 SET itemVolume TO calcVolume ( Volume )
20
21 SEND "Volume is " & itemVolume TO DISPLAY
```

Question 5 (b)

Candidates responded well to this question.

The most common errors described a function as involving calculations. This may be associated with the use of the word ‘function’ in Mathematics.

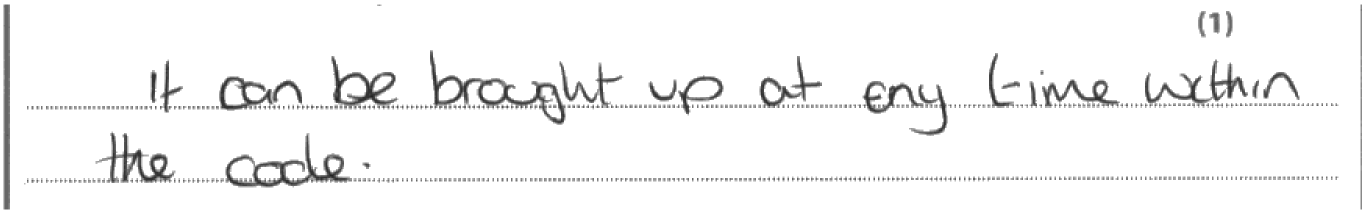
This response scored 0 marks.



(1)

It performs a calculation to find the volume from three inputs.

Other responses attempted to describe the characteristics of a function. This may be associated with the use of the word ‘function’ in Python to mean any subprogram, regardless if it returns a result or not. The following response scored 0 marks.



(1)

It can be brought up at any time within the code.

Question 5 (c)

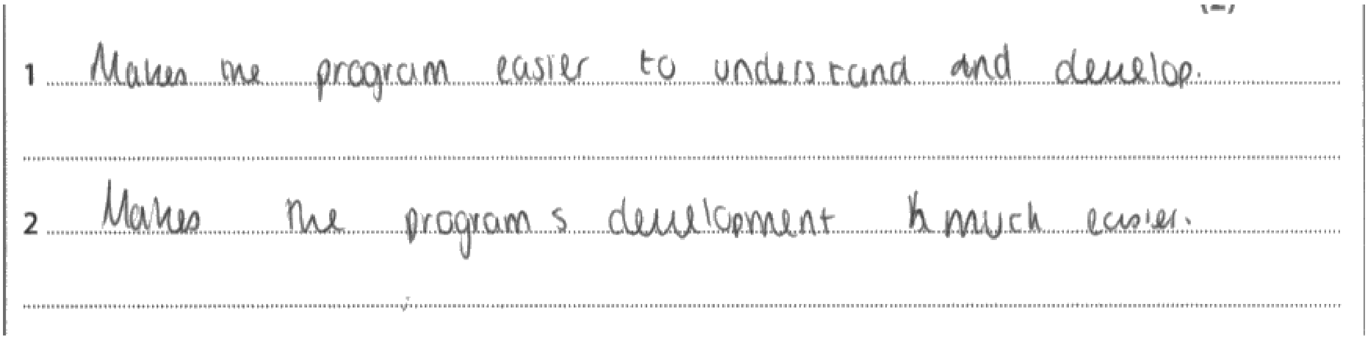
Candidates responded well to this question.

Question 5 (d)

Candidates found this question challenging. The focus of the question is evaluating how libraries aid the programmer, not describing characteristics of libraries.

Many responses just supplied descriptions of libraries. Other responses were vague and provided little detail.

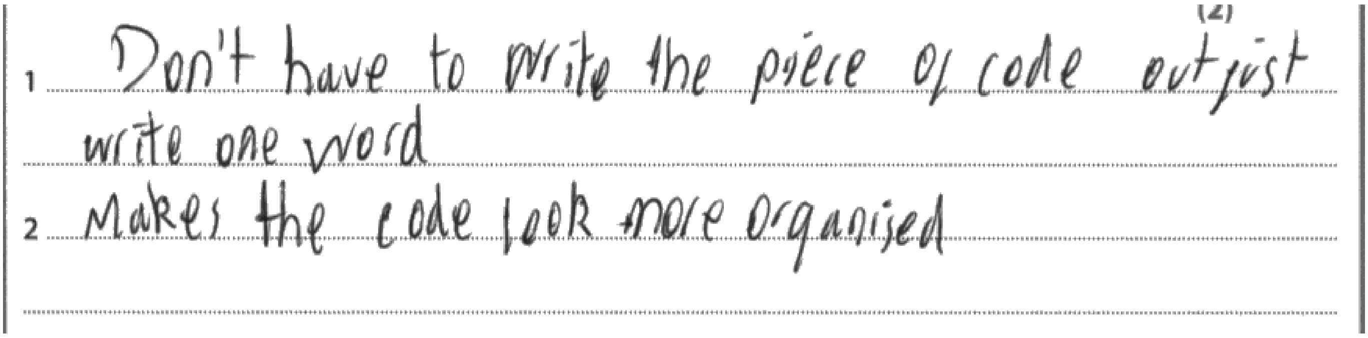
This response scored 0 marks.



1 Makes the program easier to understand and develop.

2 Makes the programs development h much easier.

This response scored 1 mark for the first part, which is equivalent to saving time.

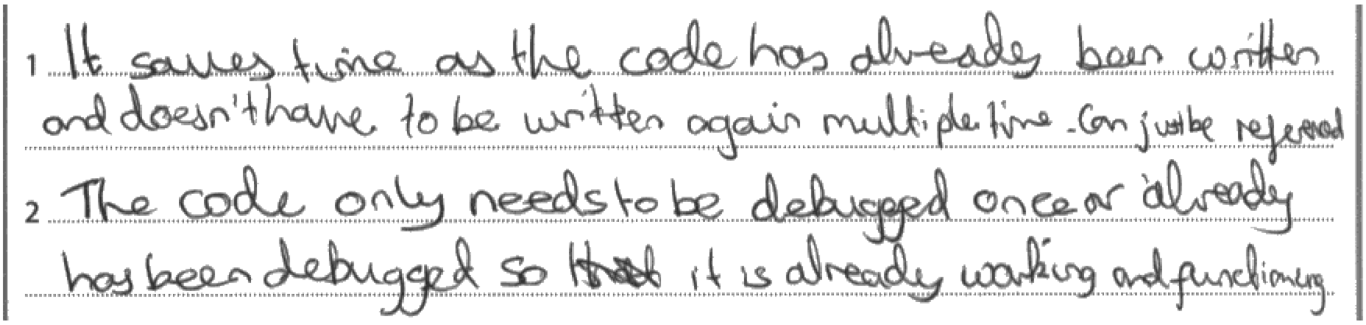


(2)

1 Don't have to write the piece of code out just write one word

2 Makes the code look more organised

The following response scored 2 marks.



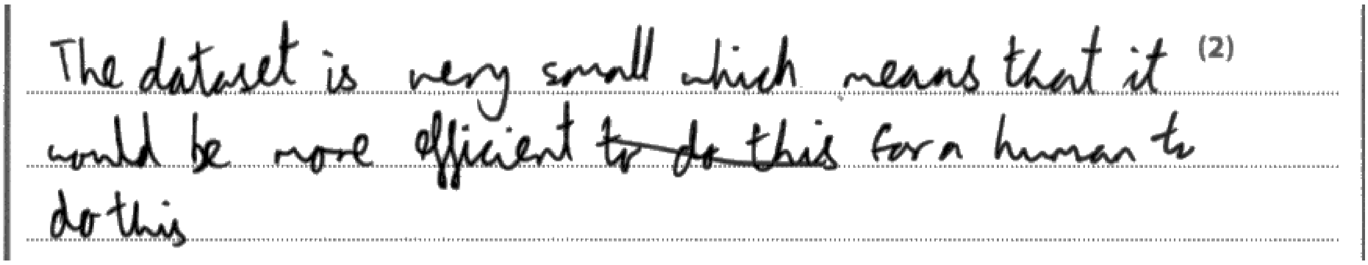
1 It saves time as the code has already been written and doesn't have to be written again multiple time. Can just be referred

2 The code only needs to be debugged once or already has been debugged so ~~that~~ it is already working and functioning

Question 6 (a) (i)

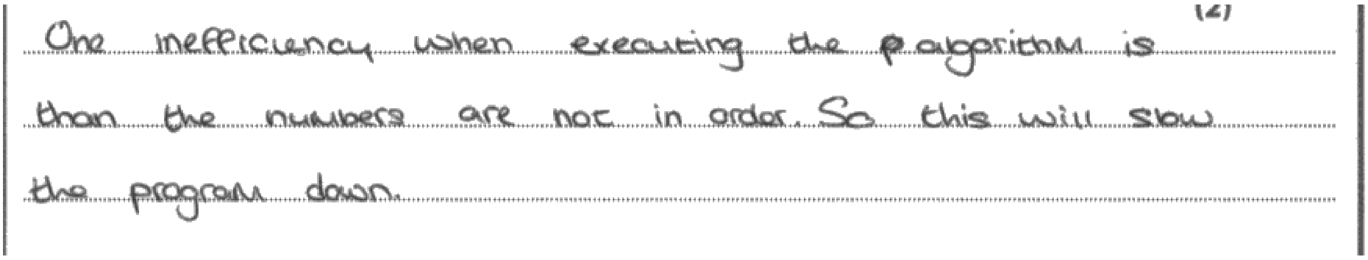
Candidates responded well to this question, demonstrating that they understand the inefficiencies of using a linear search on an unsorted list.

The following response scored 0 marks.



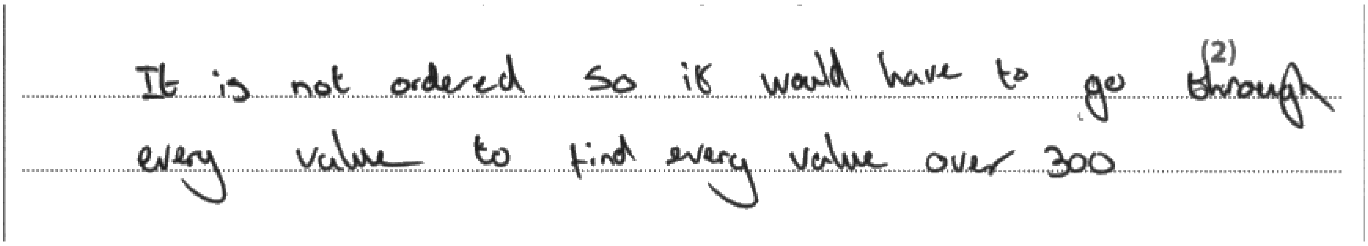
The dataset is very small which means that it (2)
would be more efficient ~~to do~~ this for a human to
do this

The following response scored 1 mark.



One inefficiency when executing the ~~p~~ algorithm is (2)
than the numbers are not in order. So this will slow
the program down.

The following response scored 2 marks.

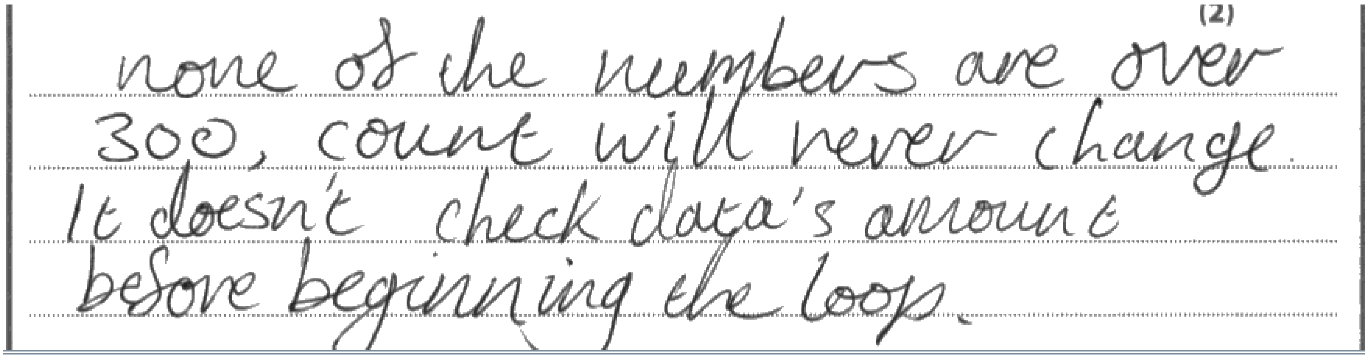


It is not ordered so it would have to go (2) through
every value to find every value over 300

Question 6 (a) (ii)

Candidates found this question challenging. However, there were responses indicating the need for looping through every item in the dataset, to report nothing in the end.

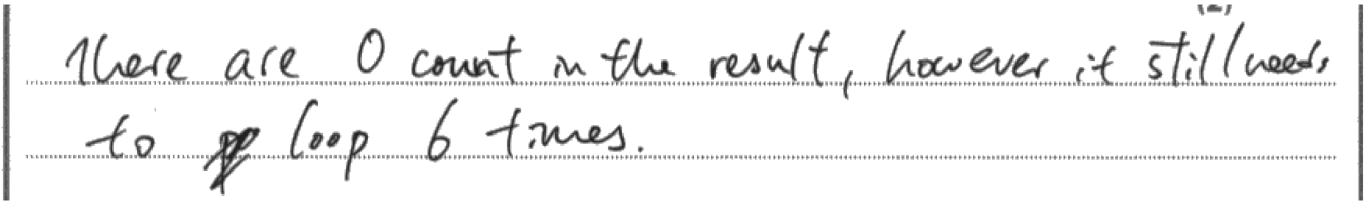
This response was awarded one mark for expressing the idea that there are no numbers over 300.



(2)

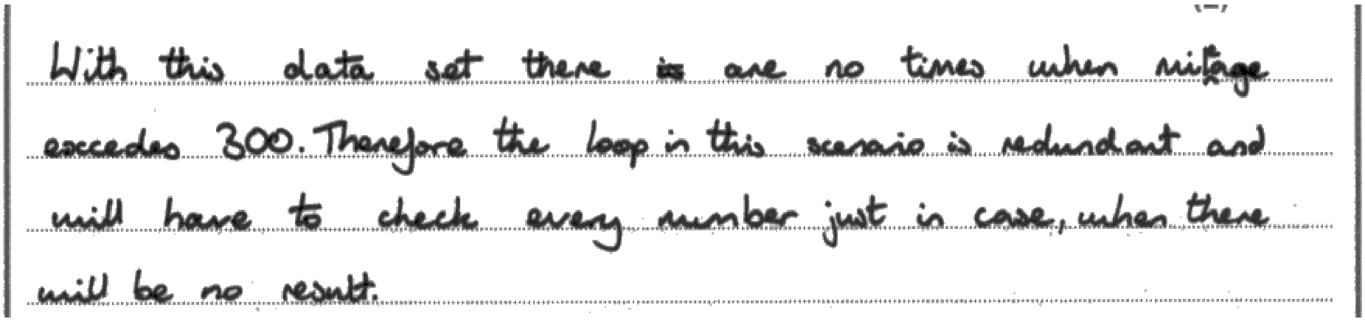
none of the numbers are over 300, count will never change. It doesn't check data's amount before beginning the loop.

The following response scored 2 marks.



There are 0 count in the result, however it still needs to ~~[illegible]~~ loop 6 times.

The following response scored 2 marks.



With this data set there ~~is~~ are no times when mileage exceedes 300. Therefore the loop in this scenario is redundant and will have to check every number just in case, when there will be no result.

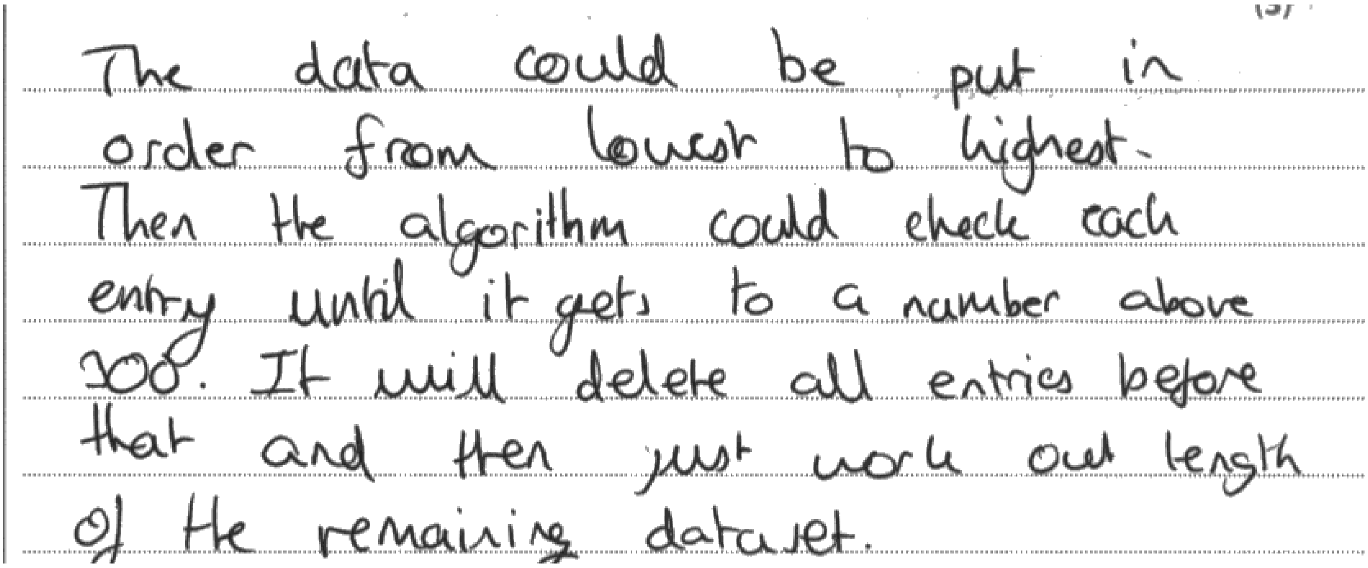
Question 6 (a) (iii)

Candidates found this question challenging.

Many responses identified that the dataset should be sorted. Some suggested that a binary search could then be used to find all the numbers over 300. A binary search would not be appropriate for solving this problem, as a binary search finds a single target. There is no guarantee that the number 300 is in the dataset at all.

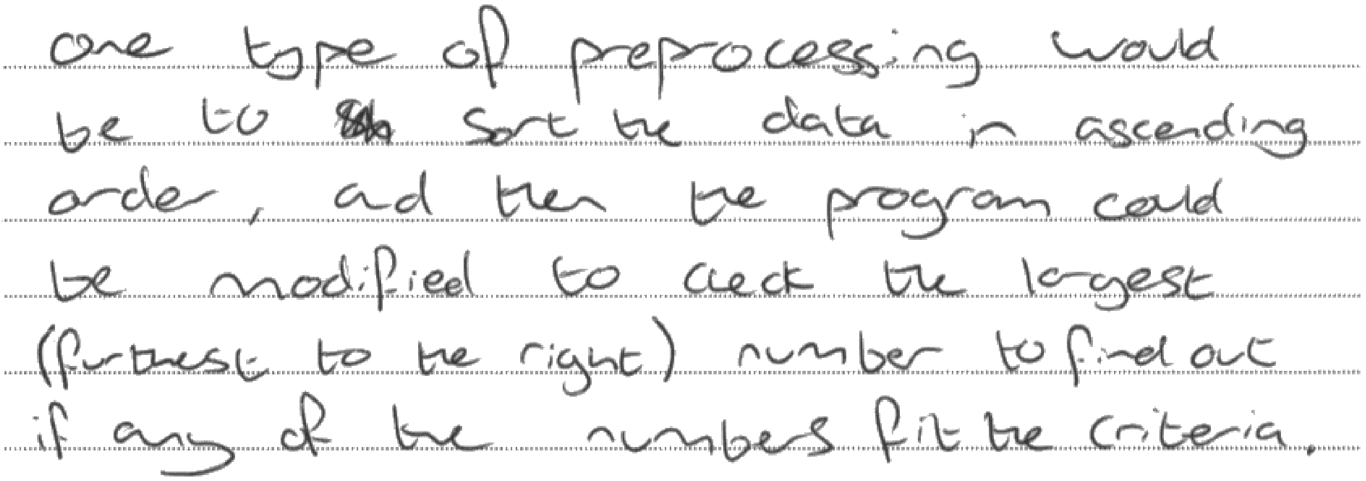
Other responses suggested sorting in ascending order, counting all those items less than 300, and then performing arithmetic based on the length of the dataset. A more appropriate solution would be only to look at the items in the dataset of interest, i.e. those over 300.

This response has been awarded one mark for indicating a sort.



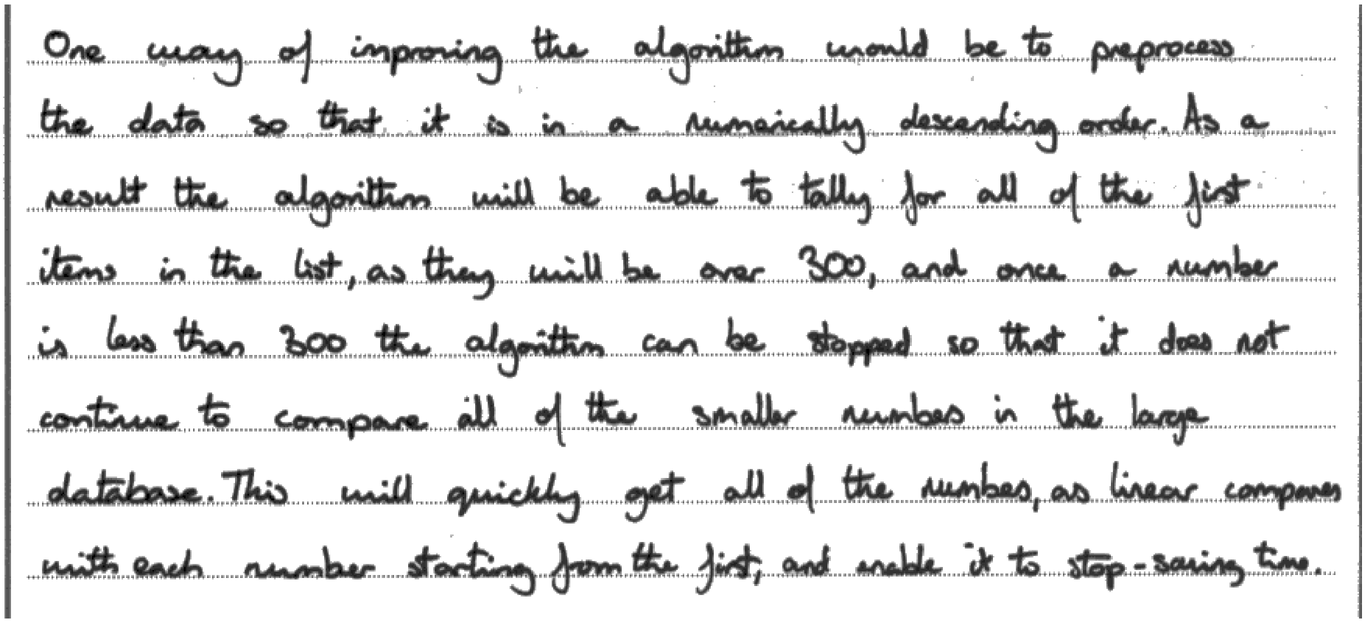
The data could be put in order from lowest to highest. Then the algorithm could check each entry until it gets to a number above 300. It will delete all entries before that and then just work out length of the remaining dataset.

The following response has been awarded two marks for indicating a sort and for checking the largest numbers first. It is missing identification of the test for less than 300.

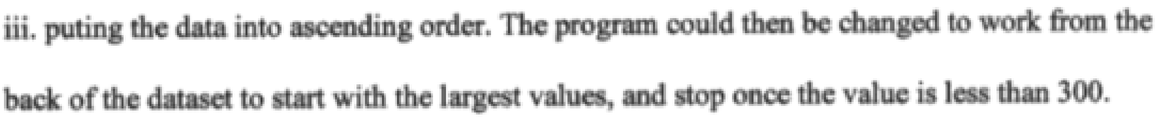


One type of preprocessing would be to ~~[illegible]~~ sort the data in ascending order, and then the program could be modified to check the largest (furthest to the right) number to find out if any of the numbers fit the criteria.

The following two responses each scored 3 marks.



One way of improving the algorithm would be to preprocess the data so that it is in a numerically descending order. As a result the algorithm will be able to tally for all of the first items in the list, as they will be over 300, and once a number is less than 300 the algorithm can be stopped so that it does not continue to compare all of the smaller numbers in the large database. This will quickly get all of the numbers, as linear compares with each number starting from the first, and enable it to stop - saving time.



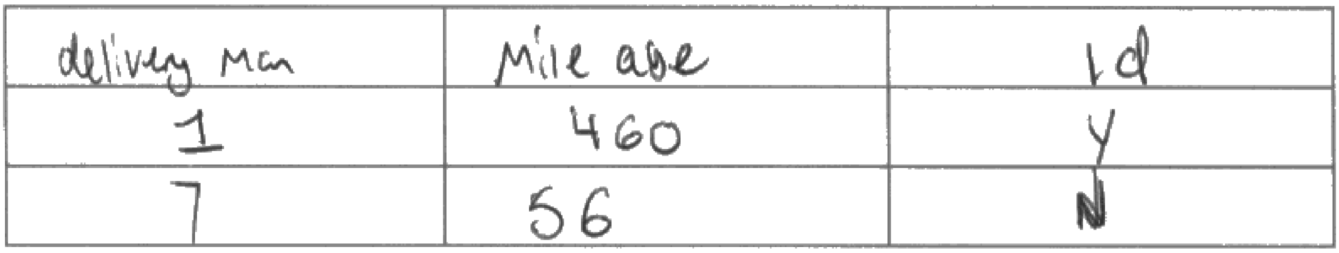
iii. puting the data into ascending order. The program could then be changed to work from the back of the dataset to start with the largest values, and stop once the value is less than 300.

Question 6 (b)

Candidates found this question challenging. Many responses indicated a confusion between 2-dimensional arrays and Python lists. This was indicated by the mixing of data types across the cells.

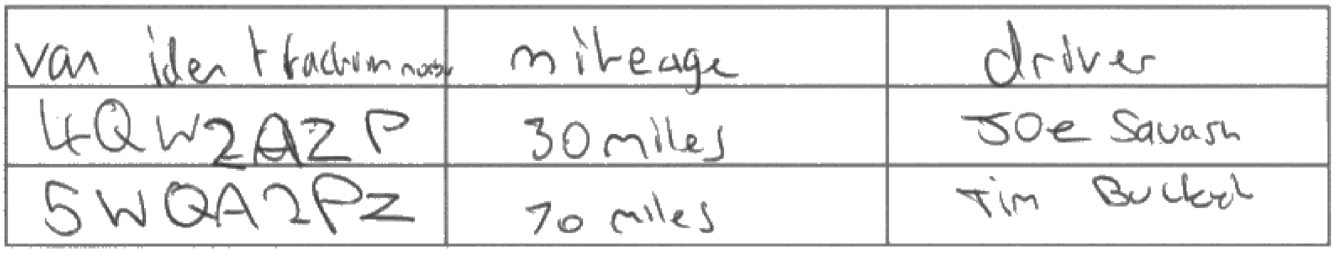
This response was awarded one mark for meaningful column headings. Four of the cells have what are obviously INTEGERS. The last column however, has CHARACTERS. That means that the lower 6 cells are not the same data type. Therefore, no mark was awarded for the cell contents.

The following response scored 1 mark.



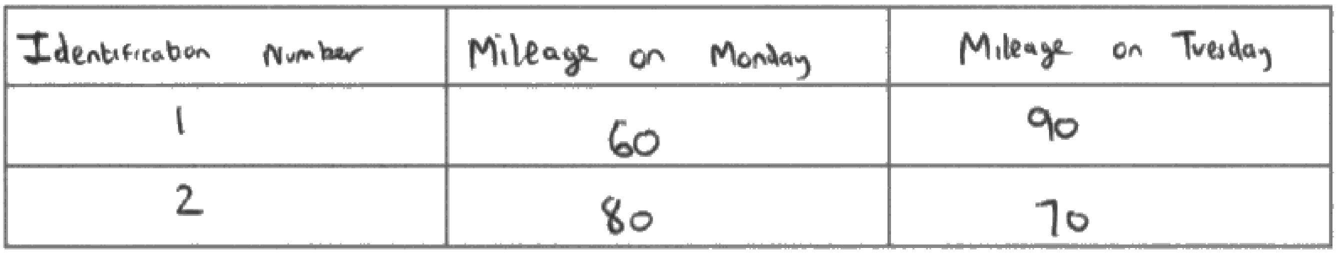
delivery Man	Mile age	Id
1	460	Y
7	56	N

The following response was awarded full marks for headings and single data types. Data types are all strings as the response clearly incorporates alphabetic characters.



van identfaction number	mileage	driver
4QW2AZP	30 miles	Joe Savash
5WQA2PZ	70 miles	Tim Bucket

The following response was awarded full marks for headings and single data types. Data types are all integers.

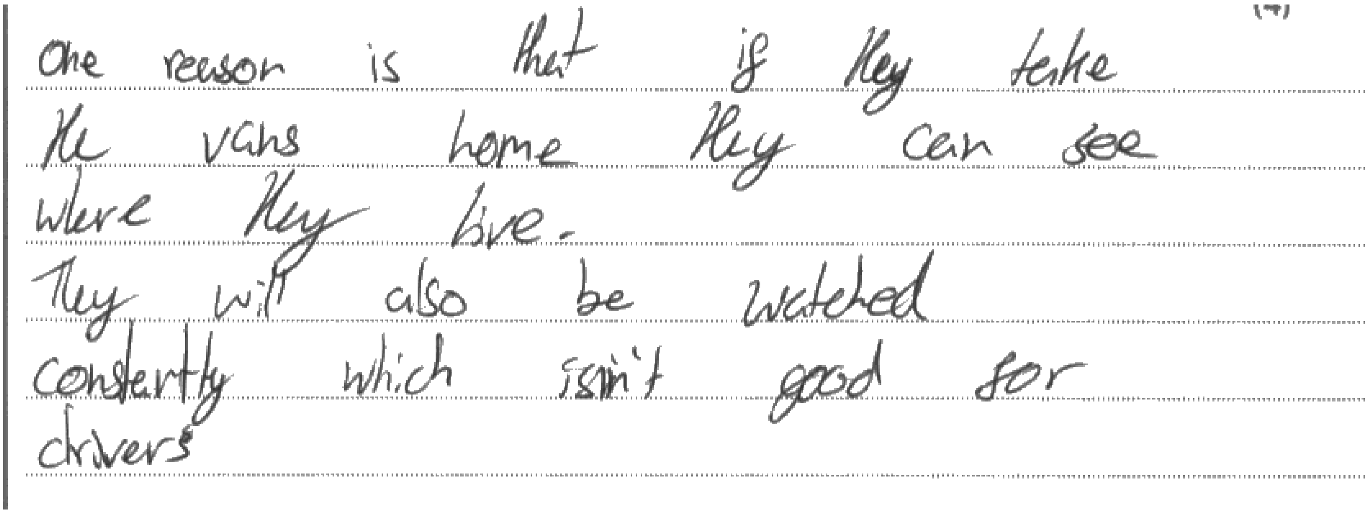


Identification Number	Mileage on Monday	Mileage on Tuesday
1	60	90
2	80	70

Question 6 (c)

Candidates responded well to this question, demonstrating the ability to link facts.

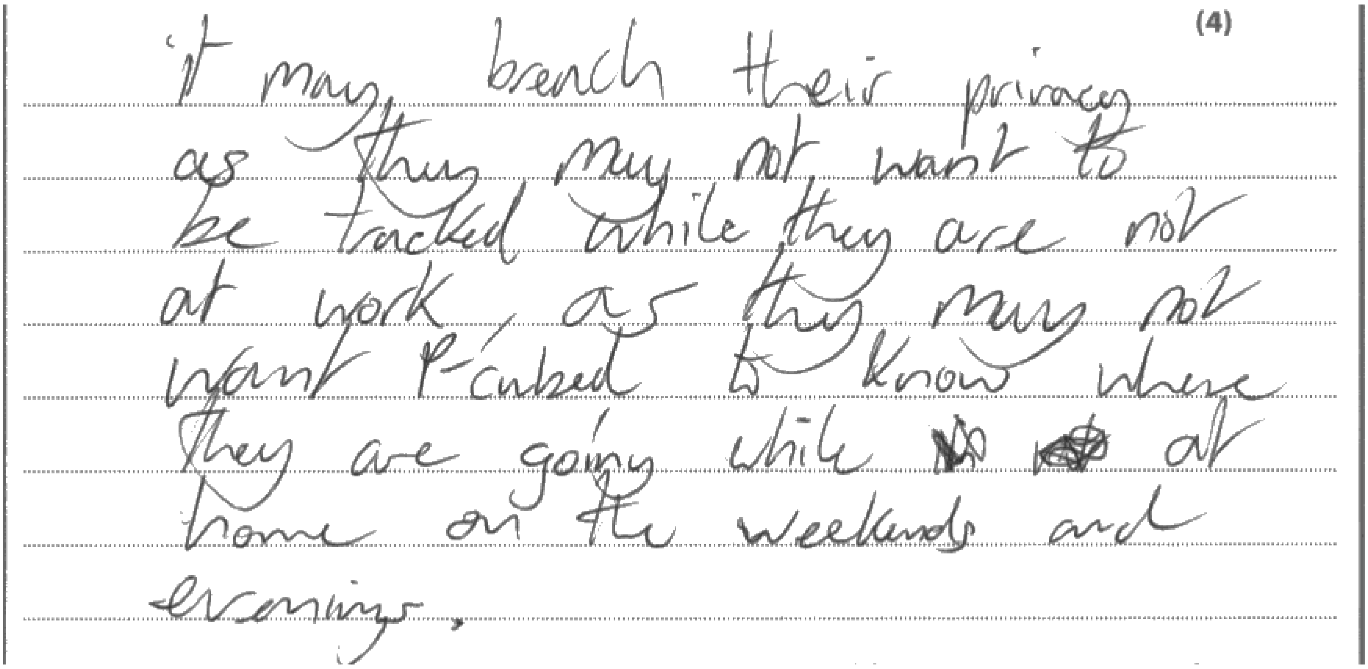
This response was awarded one mark, as there is just enough to demonstrate an understanding that personal data is being stored.



(4)

One reason is that if they take the vans home they can see where they live.
They will also be watched constently which isn't good for drivers

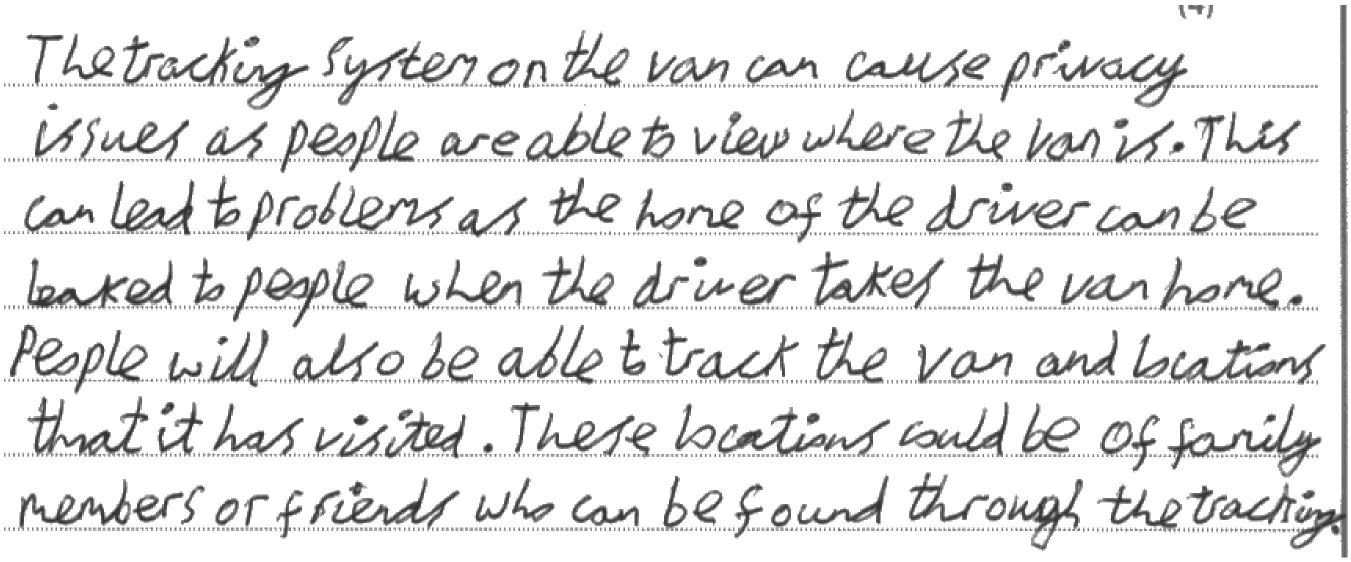
The following response was awarded two marks, in line with bullet 1 in the mark scheme.



(4)

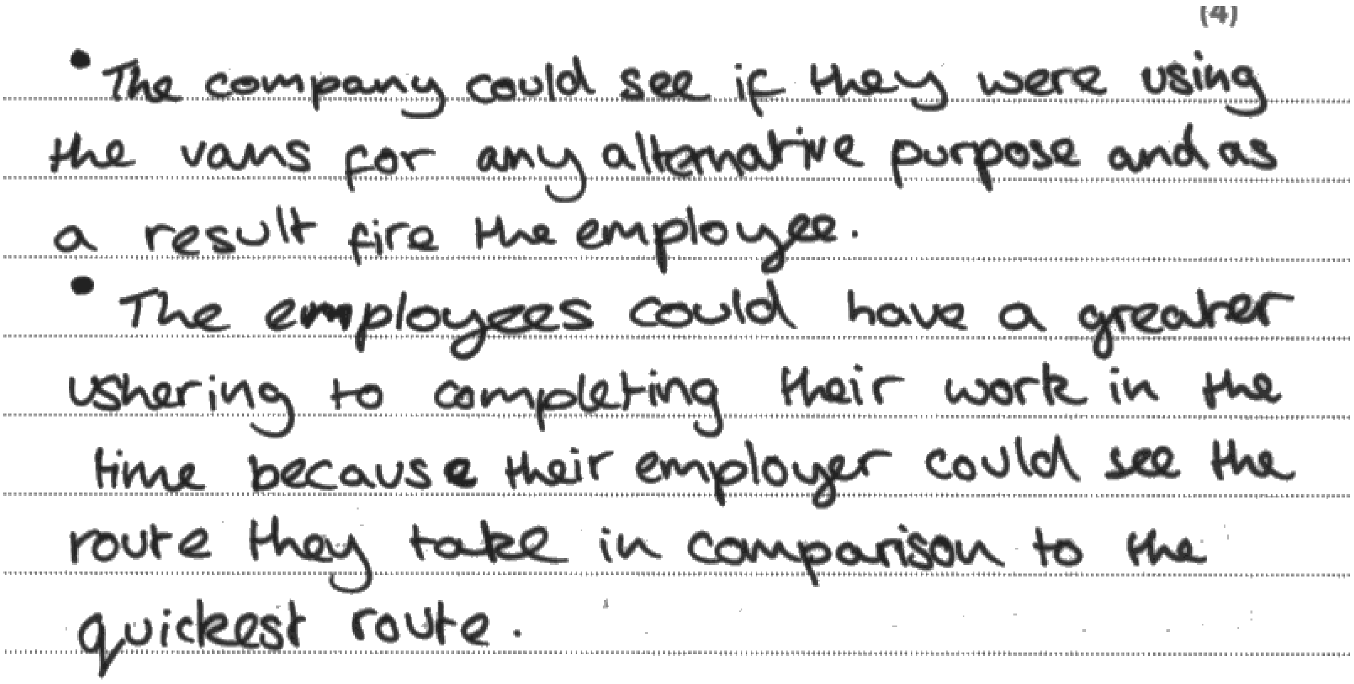
it may breach their privacy as they may not want to be tracked while they are not at work, as they may not want P-Cubed to know where they are going while at home on the weekends and evenings.

The following response was awarded three marks, with two linked and one independent.


(4)

The tracking System on the van can cause privacy issues as people are able to view where the van is. This can lead to problems as the home of the driver can be leaked to people when the driver takes the van home. People will also be able to track the van and locations that it has visited. These locations could be of family members or friends who can be found through the tracking

The following response was awarded four marks, for two linked responses. Using the vans for own purpose and some ramification (job loss) match bullet 5. Pressure to complete tasks influences judgements made about driving (route) can be awarded to bullet 2.

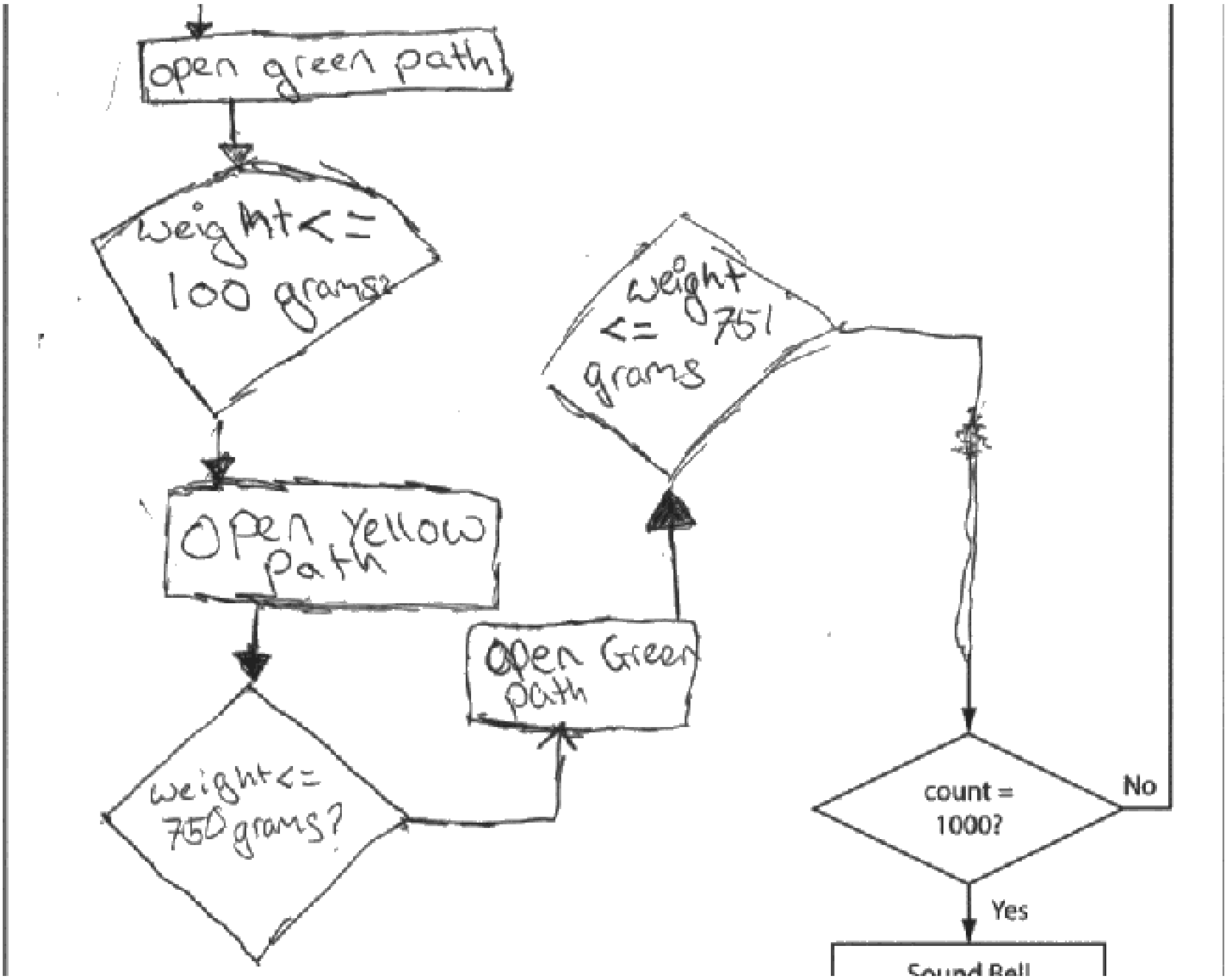

(4)

- The company could see if they were using the vans for any alternative purpose and as a result fire the employee.
- The employees could have a greater ushering to completing their work in the time because their employer could see the route they take in comparison to the quickest route.

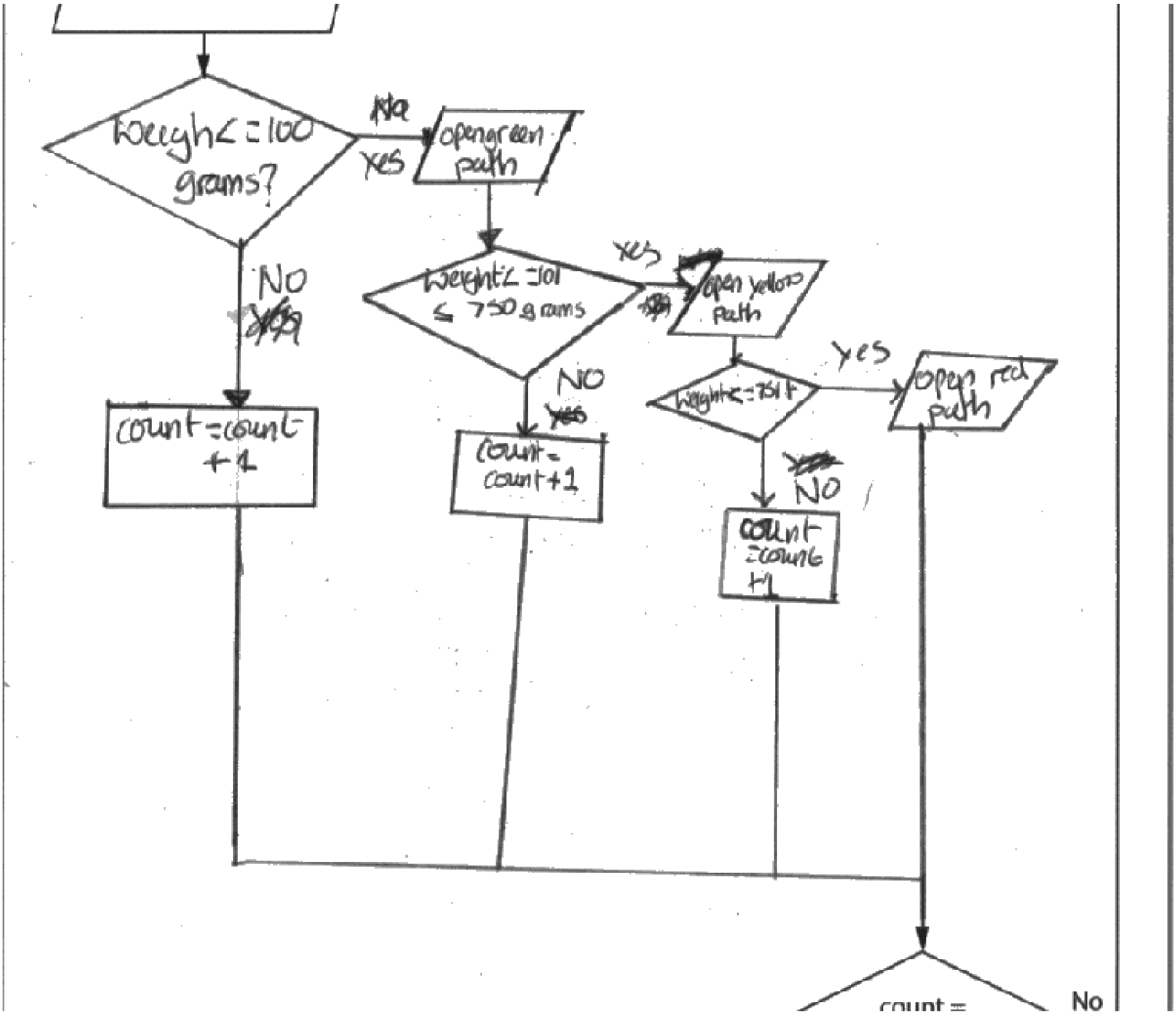
Question 7 (a)

Candidates responded well to this question. There were several different ways that the symbols could be arranged and connected to provide a solution to the problem. Frequent errors included using the symbols more than once and adding symbols. The question expressly states to use each symbol only once.

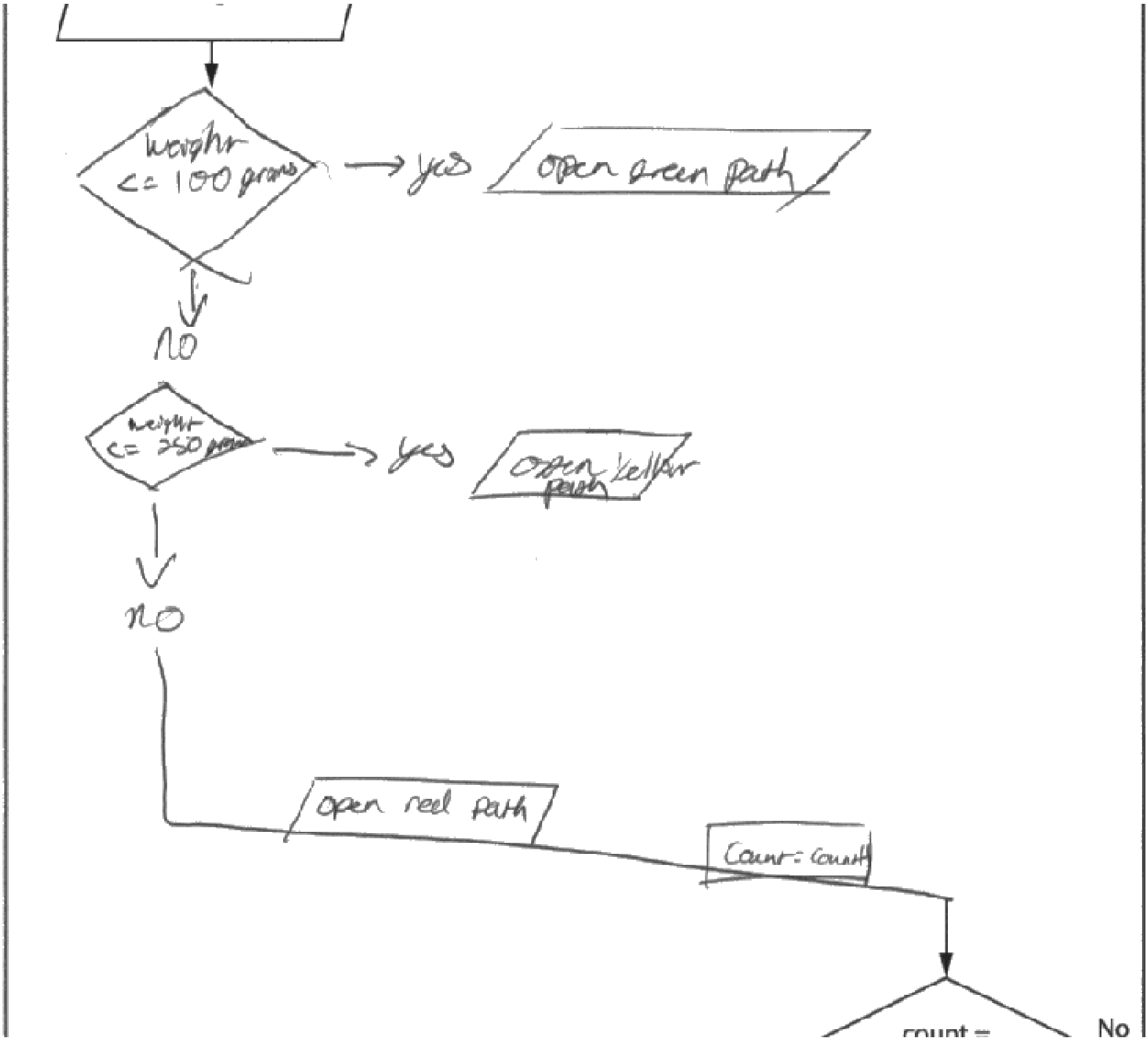
The following response was awarded no marks.



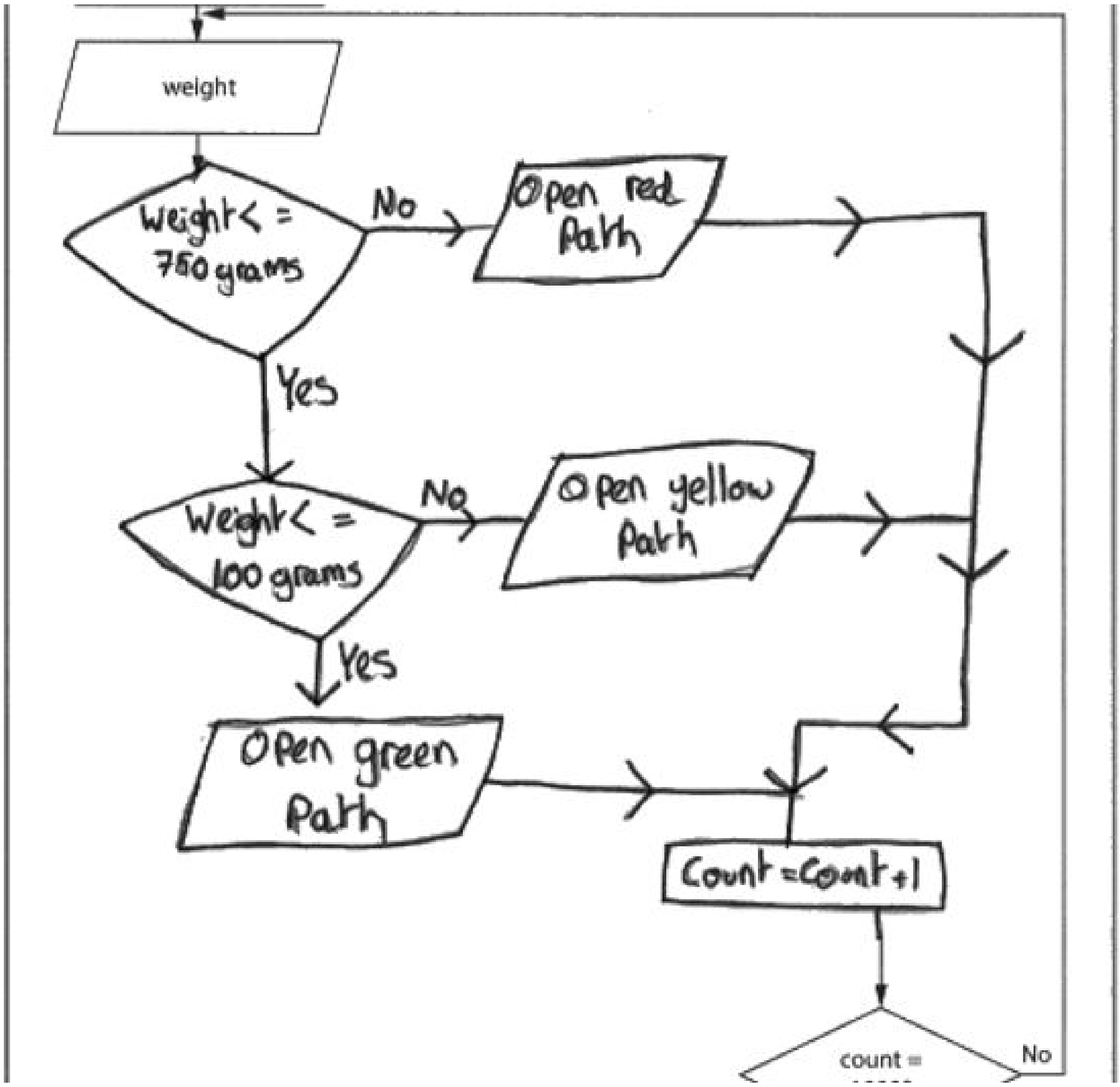
The following response was awarded two marks – one mark for green path, and one mark for yellow path.



The following response was awarded four marks. One for the order of the decision boxes, and one for each of the three coloured paths.



The following response was awarded six marks. The order of the tests is reversed from that in the mark scheme, but the correct use of labels provides a working solution.



Question 7 (b) (i)

Candidates found this question challenging. Many recognised that a primary key is associated with uniqueness, but could not identify what had to be unique. Responses that did not provide enough detail and lacked the use of subject-specific terminology were common.

The following response was awarded no marks, as a primary key does not identify a single field.

> It is a unique Field ~~the~~ Fand in each table

The following response was awarded no marks, as the term item could be interpreted to be either a record, entity, row, field, or attribute.

> A primary key is a unique key used to identify each individual item.

Responses that demonstrated confusion with foreign key were seen often. Although a primary key can be used as foreign key in another table, that is not its main purpose.

The following response scored 0 marks.

> (1)
>
> The primary key is a key which can be used to connect two tables in the original table.

Responses that recognised the concept of uniquely identifying a row in a database table were awarded.

The following response scored 1 mark.

> So that each record has its own unique identification number to be easily identified from.

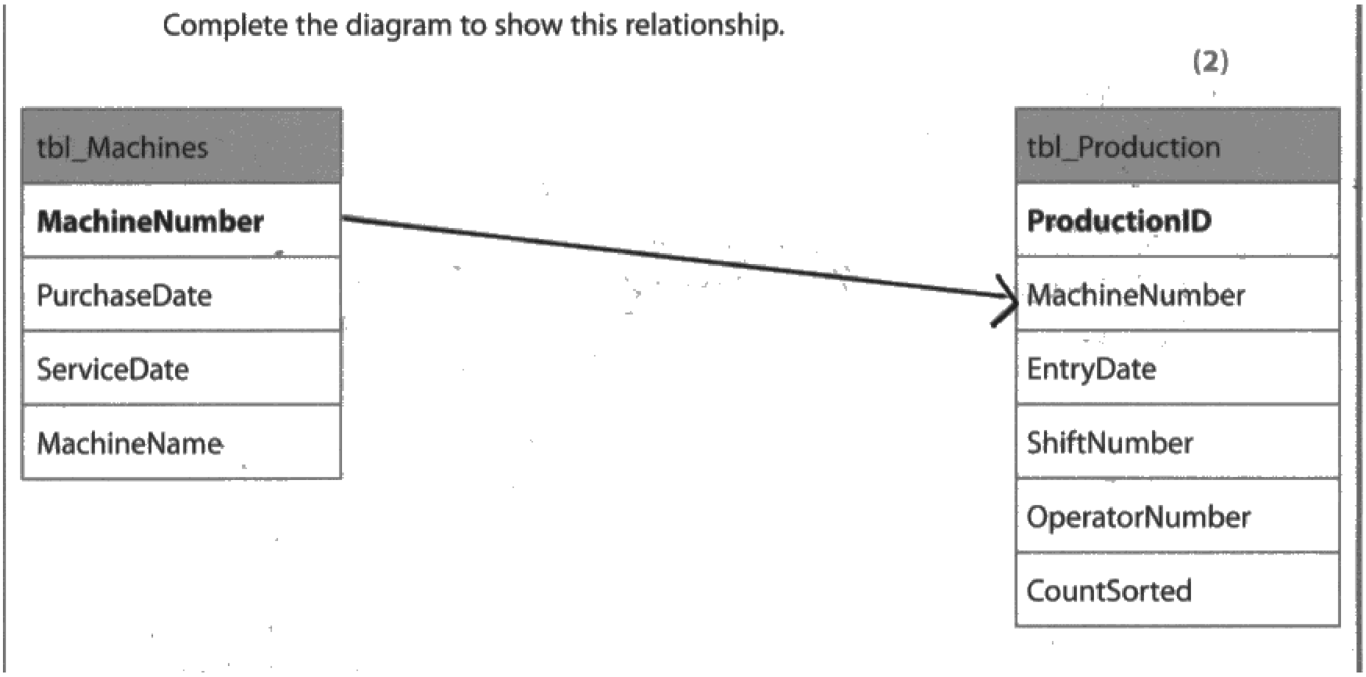
Question 7 (b) (ii)

Candidates responded well to this question.

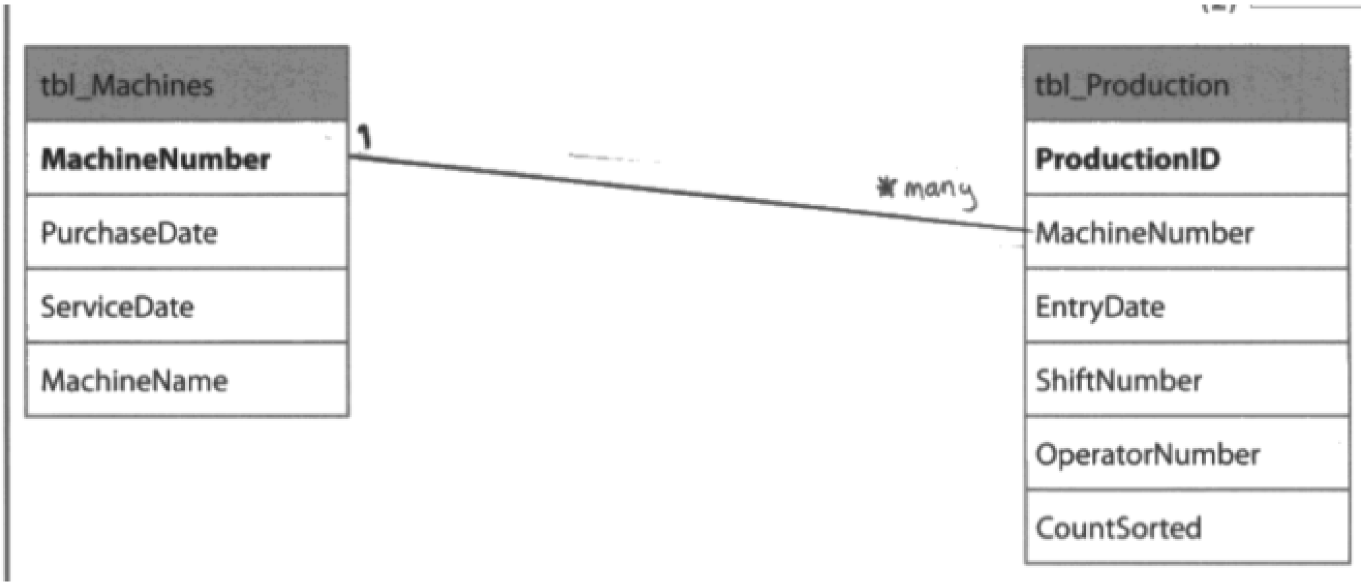
Question 7 (b) (iii)

Candidates found this question challenging. Several different types of conventional notation were acknowledged in the mark scheme. However, if a response could not clearly be interpreted, it could not be awarded marks. Crow's feet, used to correctly connect the two fields, were awarded 2 marks.

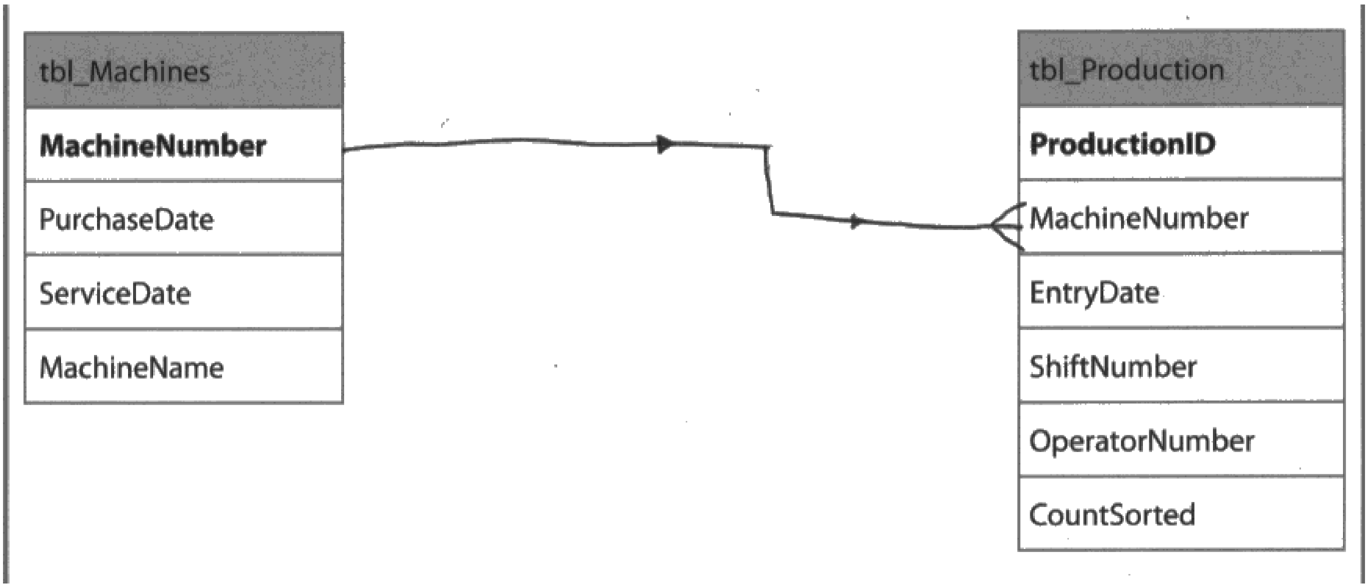
The following response was awarded one mark for the line connecting the correct fields.



The following response was awarded two marks. This diagram has the correct fields connected and has indicated the degree of the relationship.



The following response was awarded two marks. The correct fields are connected and the crow's feet indicate the degree of the relationship.



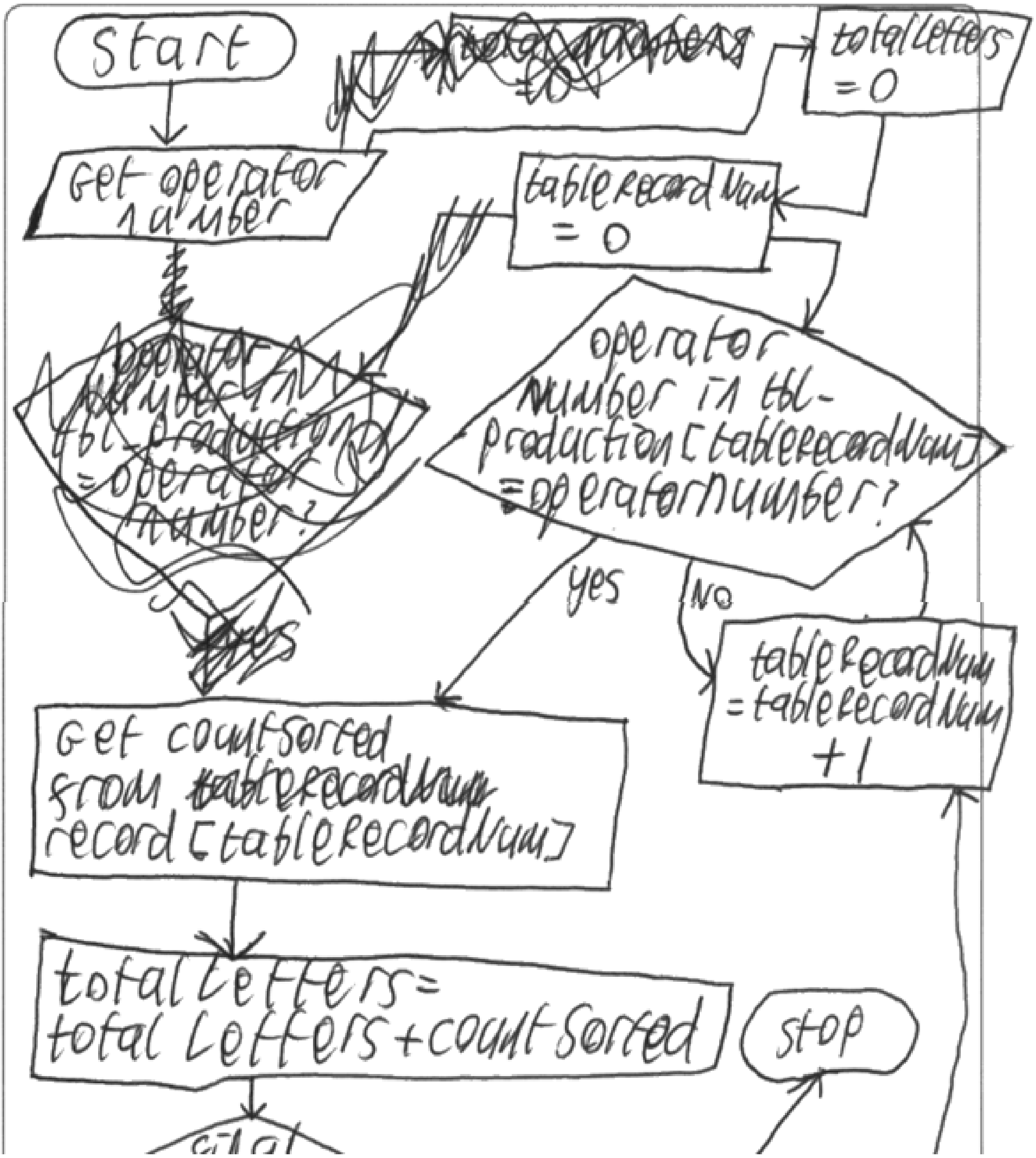
tbl_Machines
MachineNumber
PurchaseDate
ServiceDate
MachineName

tbl_Production
ProductionID
MachineNumber
EntryDate
ShiftNumber
OperatorNumber
CountSorted

Question 7 (b) (iv)

Candidates found this question challenging. Many responses made an attempt to count something. Some made an attempt to count letters. Fewer made an attempt to process the records in tbl_Production in a loop. The most often overlooked requirement was consideration for checking the date to see if it is the current year.

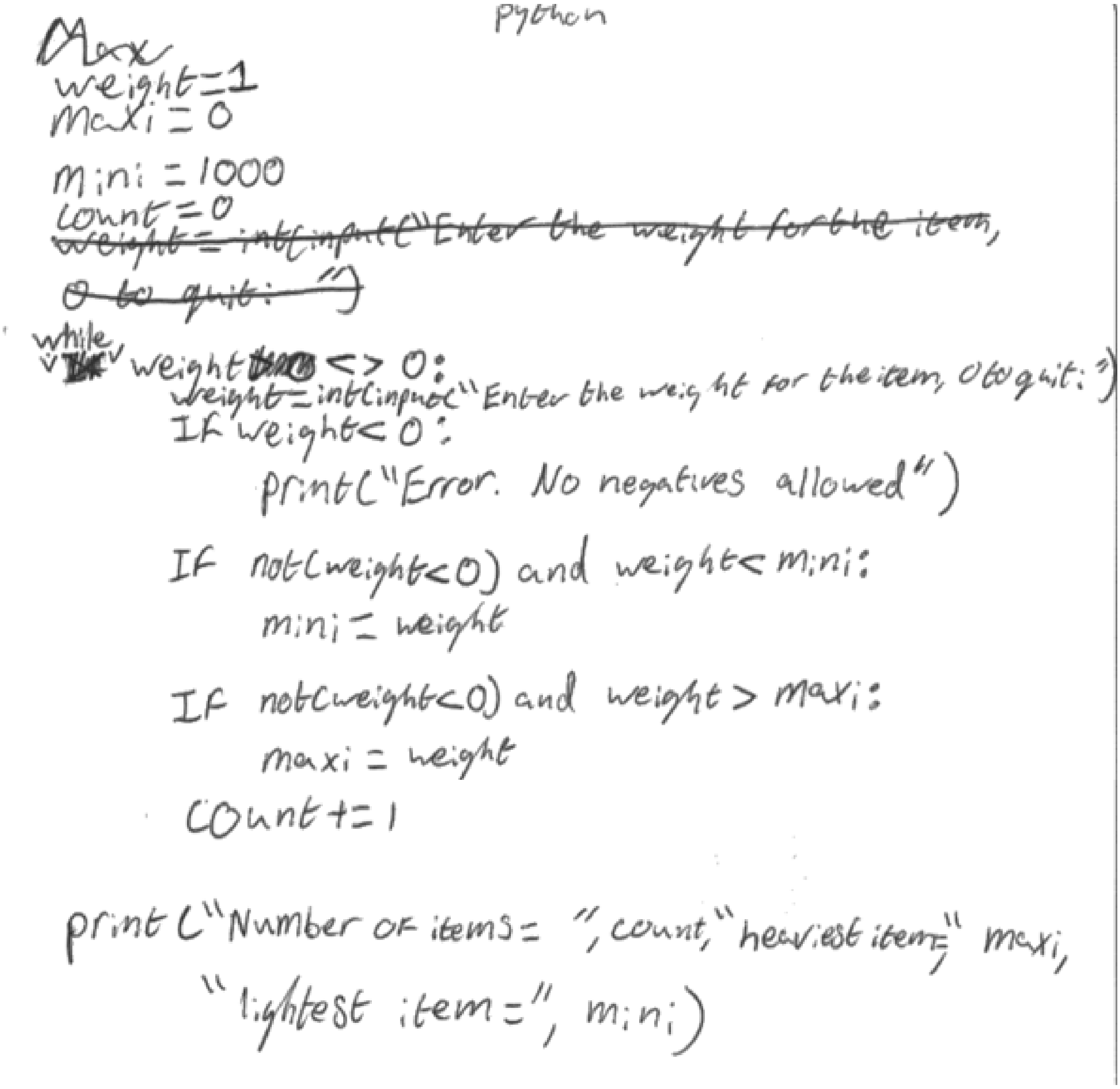
This response was awarded five marks. It is missing consideration for the date and the decision box should have only a single arrow in.


Start
Get operator number
totalLetters =0
tableRecordNum = 0
operator number in tbl-Production[tableRecordNum] =operatorNumber?
yes
No
tableRecordNum =tableRecordNum +1
Get countSorted from record[tableRecordNum]
totalLetters= totalLetters+countSorted
stop

Question 8

The full range of marks was awarded for responses to this question. used a variety of programming languages and pseudocode to express their solutions. Many attempted to use various facilities of their programming language, such as sort, min, and max. Many chose to store the data into arrays or lists. This additional complexity led to candidates making mistakes. Those candidates who chose to implement the straightforward solution described in the mark scheme were less likely to introduce mistakes.

The following response was awarded 8 marks.

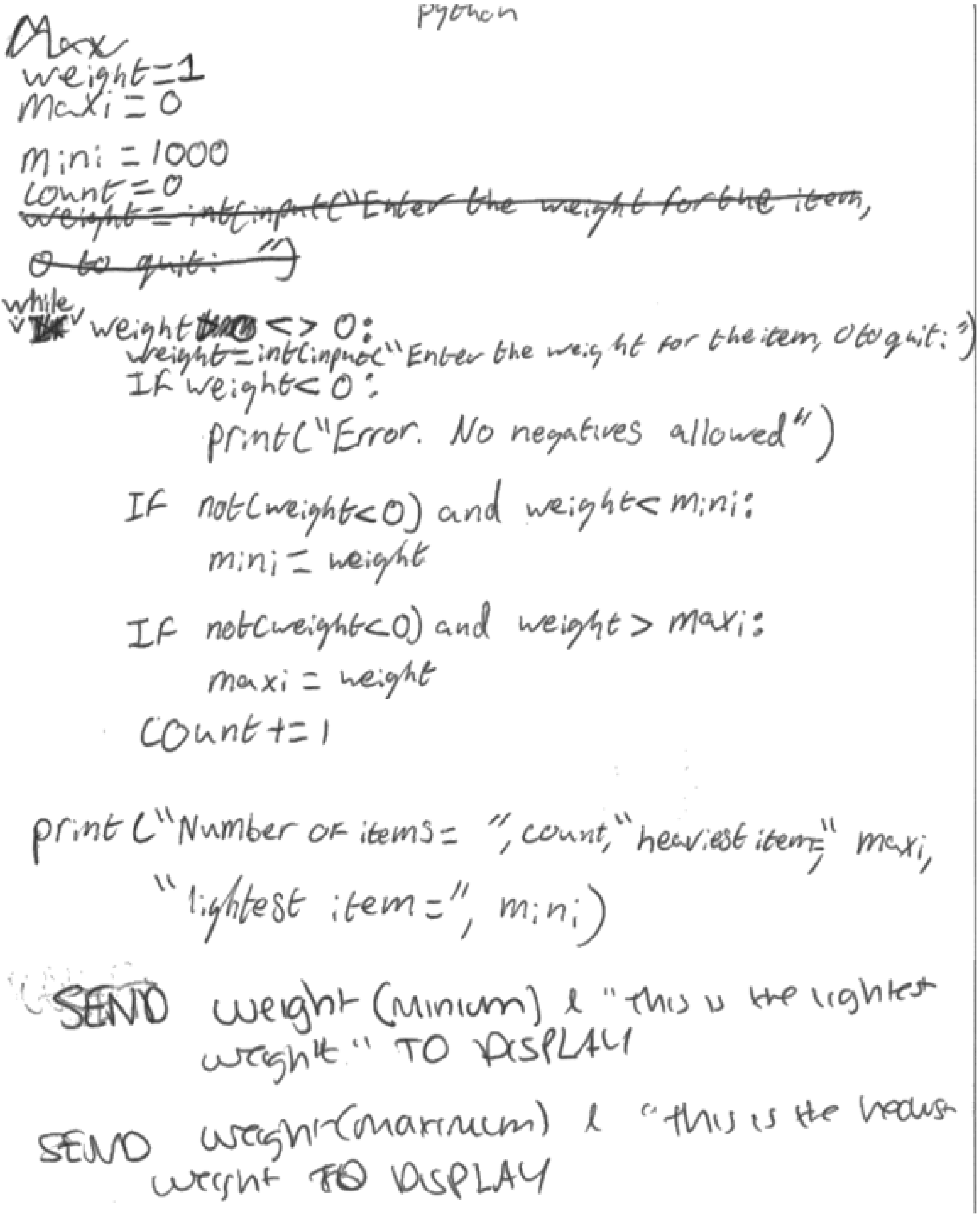


Max

Python

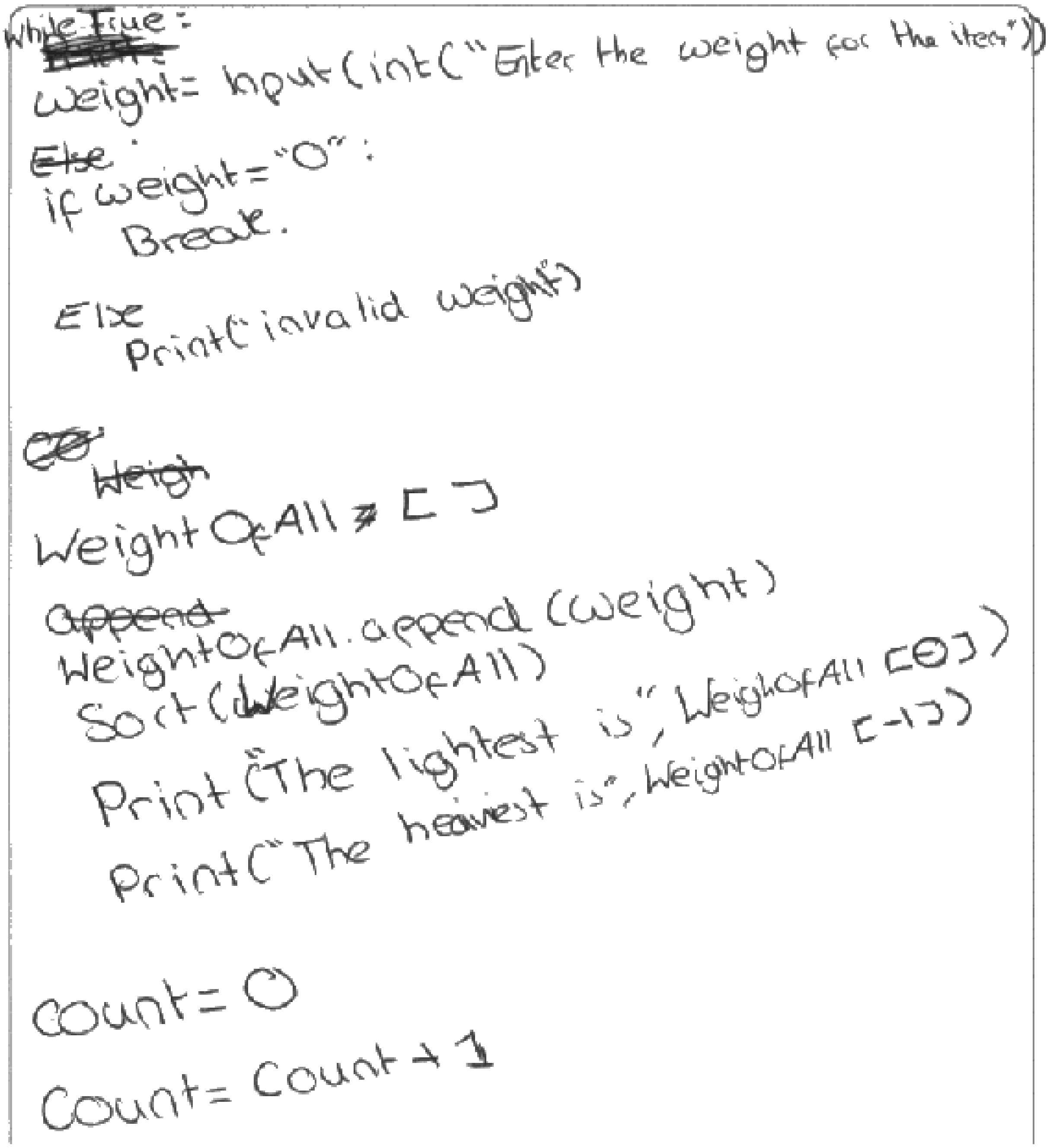
```
weight=1
maxi=0
mini=1000
count=0
~~weight=int(input("Enter the weight for the item,~~
~~0 to quit: ")~~
while weight <> 0:
    weight=int(input("Enter the weight for the item, 0 to quit: ")
    If weight<0:
        print("Error. No negatives allowed")

    If not(weight<0) and weight<mini:
        mini=weight

    If not(weight<0) and weight>maxi:
        maxi=weight
    count+=1

print("Number of items= ", count, "heaviest item=", maxi,
    "lightest item=", mini)
```

The following response was awarded three marks. This takes in the weights from the user correctly and validates for negative numbers. It does the counting as well. However, it prints out the count inside the while loop. The 'repeat' at the top has no matching end, indicated either by keyword blocking or indentation. The attempt to use min/max can't work as there's no array. There is a recognisable notation, but indentation would improve readability. Variable names are meaningful. Response uses looping correctly, so the outermost part of the program is appropriate.



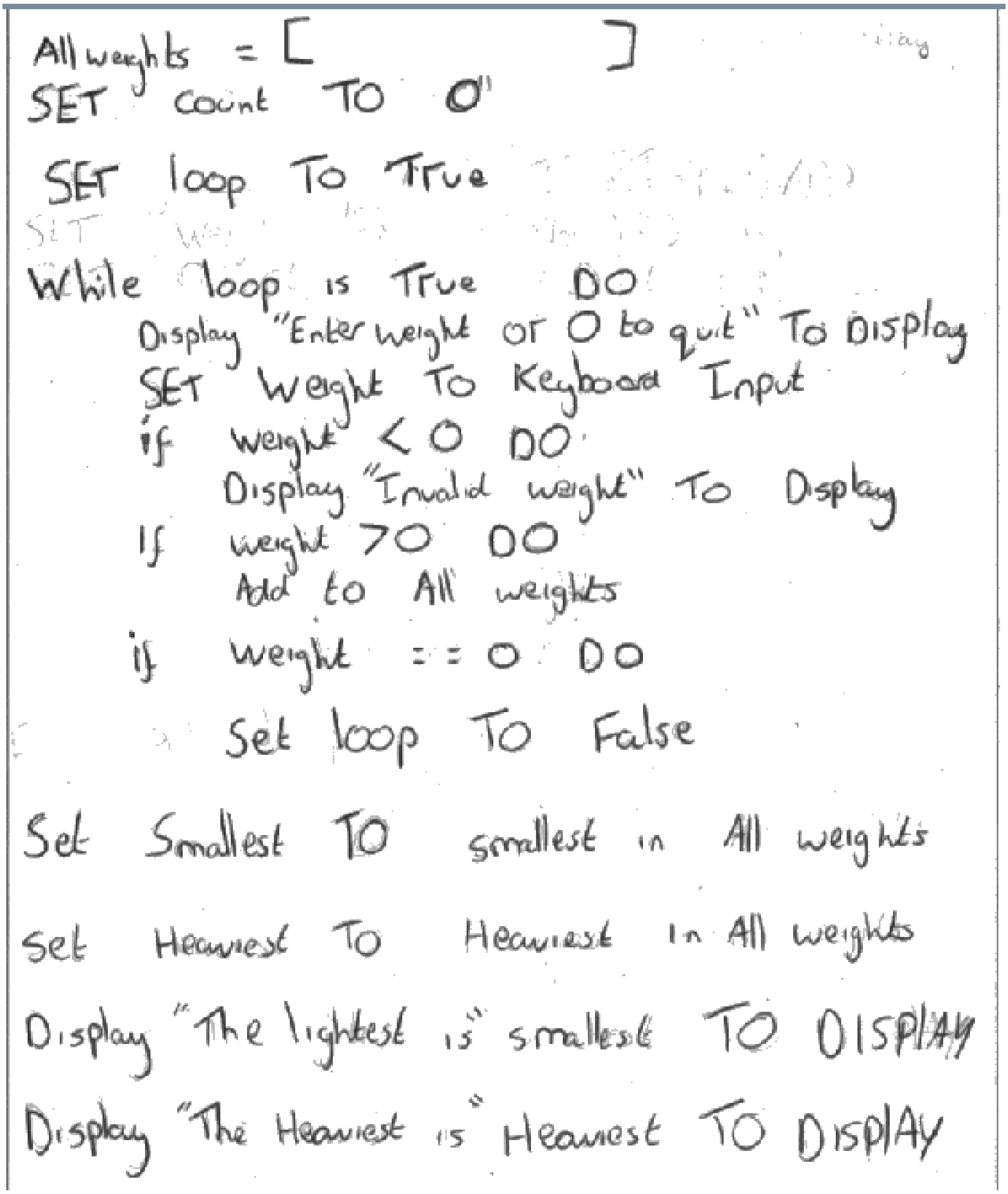
```
Max                       python
weight=1
maxi=0

mini=1000
count=0
~~weight=int(input("Enter the weight for the item,~~
~~0 to quit: ")~~

while
weight <> 0:
    weight=int(input("Enter the weight for the item, 0 to quit: ")
    If weight<0:
        print("Error. No negatives allowed")

    If not(weight<0) and weight<mini:
        mini= weight

    If not(weight<0) and weight > maxi:
        maxi = weight
    count+=1

print("Number of items= ", count, "heaviest item=", maxi,
    "lightest item=", mini)

SEND weight (minium) & "this is the lightest
    weight" TO DISPLAY

SEND weight(maximum) & "this is the heaviest
    weight TO DISPLAY
```

The following response was awarded two marks. The solution incorporates some of the required constructs, but they are not used in the correct order. There is an attempt to use Python lists and subprograms, but these are not used accurately. The notation is readable, although because the code slopes right, it is difficult to identify the indentation.



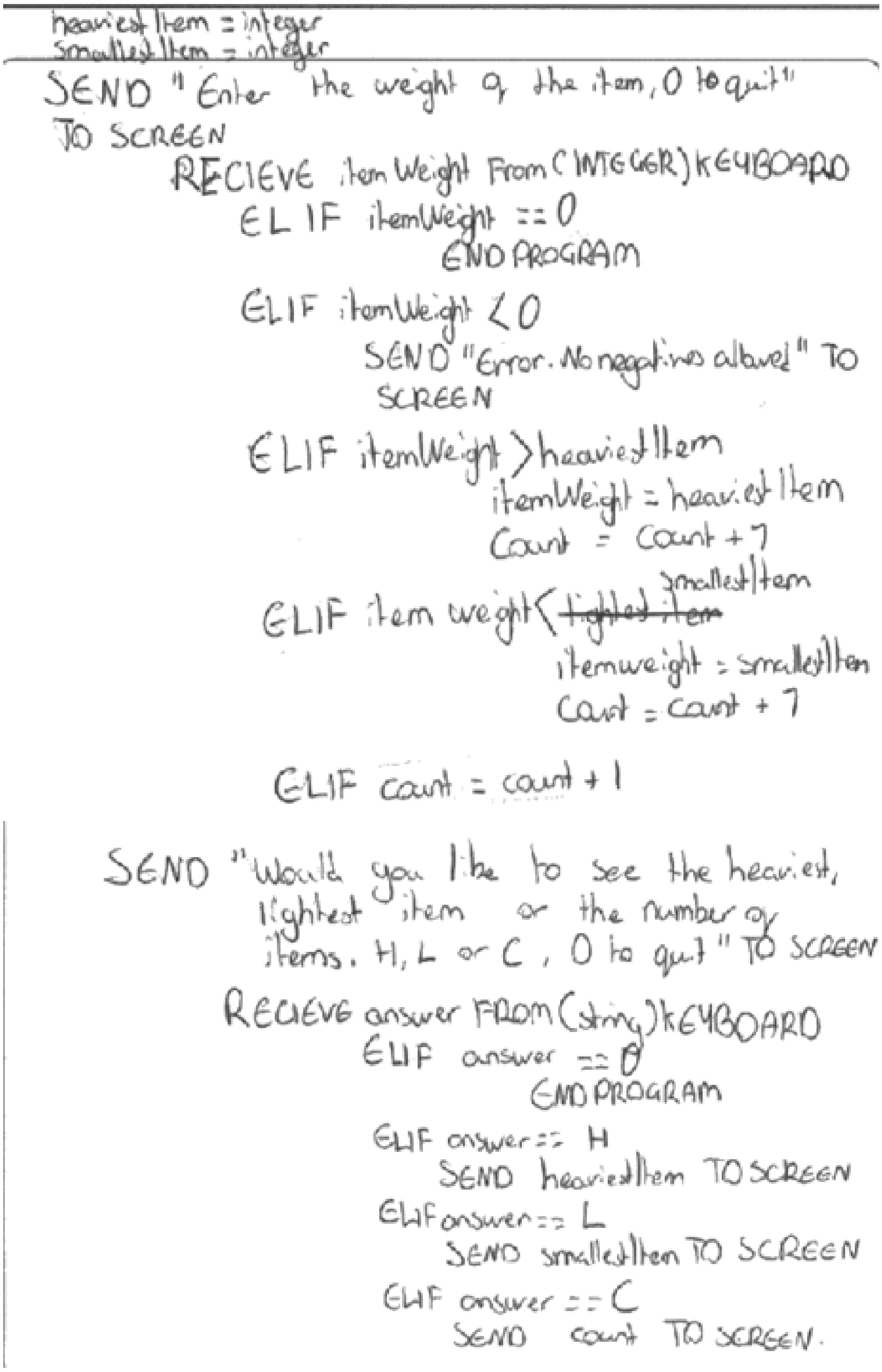
```
While True :
weight= input(int("Enter the weight for the item"))
~~Else~~
if weight="0":
    Break.

 Else
    Print(" invalid weight")


~~CO~~
  ~~Weigh~~
WeightOfAll = [ ]
 ~~append~~
 WeightOfAll.append (weight)
  Sort(WeightOfAll)
   Print ("The lightest is", WeighOfAll [0])
    Print(" The heaviest is", WeightOfAll [-1])

count=0
Count= Count + 1
```

The following response was awarded five marks. The loop is functionally correct, validation for negative numbers is implemented, and the loop terminates. The array of weights is not needed. The notation is adequate and follows the pseudocode convention. The solution does not address counting, finding the minimum and the maximum.



```
All weights = [                ]
SET Count TO 0
SET loop To True
While loop is True DO
    Display "Enter weight or 0 to quit" To Display
    SET Weight To Keyboard Input
    if weight < 0 DO
        Display "Invalid weight" To Display
    If weight > 0 DO
        Add to All weights
    if weight == 0 DO
        Set loop To False
Set Smallest TO smallest in All weights
Set Heaviest To Heaviest In All weights
Display "The lightest is" smallest TO DISPLAY
Display "The Heaviest is" Heaviest TO DISPLAY
```

The following response was awarded six marks. The solution receives the weight, checks for ending program, counts, identifies both minimum and maximum. However, it doesn't loop so only really processes a single weight. The lower part of the code is not required by the solution. The notation can be followed.



```
heaviestItem = integer
smallestItem = integer
SEND "Enter the weight of the item, 0 to quit"
TO SCREEN
        RECIEVE itemWeight From (INTEGER) KEYBOARD
            ELIF itemWeight == 0
                    END PROGRAM

            ELIF itemWeight < 0
                    SEND "Error. No negatives allowed" TO
                    SCREEN

            ELIF itemWeight > heaviestItem
                    itemWeight = heaviestItem
                    Count = Count + 7
                                          smallestItem
            ELIF item weight < ~~lightest item~~
                    itemweight = smallestItem
                    Count = count + 7

            ELIF count = count + 1

    SEND "Would you like to see the heaviest,
          lightest item or the number of
          items. H, L or C , 0 to quit" TO SCREEN
        RECIEVE answer FROM (string) KEYBOARD
            ELIF answer == 0
                    END PROGRAM
            ELIF answer == H
                    SEND heaviestItem TO SCREEN
            ELIF answer == L
                    SEND smallestItem TO SCREEN
            ELIF answer == C
                    SEND count TO SCREEN.
```

Paper Summary

Based on their performance on this paper, candidates are offered the following advice:

- Revise subject-specific terminology
- Revise the rules of mathematical precedence (BIDMAS)
- Recognise that fundamental data types may not be the same as in programming languages
- Practise problem solving using generic programming constructs rather than language-specific functionality
- Read the question carefully to understand what is and is not allowed.
- Take care with presentation, both in drawings and in writing. Clearly presented responses allow candidates to better demonstrate their understanding.

Grade Boundaries

Grade boundaries for this, and all other papers, can be found on the website on this link:

http://www.edexcel.com/iwantto/Pages/grade-boundaries.aspx

Pearson Education Limited. Registered company number 872828
with its registered office at 80 Strand, London WC2R 0RL.