Please check the examination details below before entering your candidate information

Candidate surname	Other names

Pearson BTEC Level 3 Nationals Extended Certificate, Foundation Diploma, Diploma, Extended Diploma

Centre Number	Learner Registration Number
□□□□□□	□□□□□□□

Monday 3 June 2019

Morning (Time: 2 hours) | Paper Reference **31768H**

Computing

Unit 1: Principles of Computer Science

You must have:
Information Booklet (enclosed)

Total Marks

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and learner registration number.
- Answer **all** questions.
- Answer the questions in the spaces provided – *there may be more space than you need.*

Information

- The total mark for this paper is 90.
- The marks for **each** question are shown in brackets – *use this as a guide as to how much time to spend on each question.*

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

Turn over ▶

P61561A

1/1/1/1/1/1

P 6 1 5 6 1 A 0 1 2 0

Pearson

Answer ALL questions. Write your answers in the spaces provided.

Please refer to the Information Booklet in order to answer Question 1.

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

....

....

Reason 2

....

....

Reason 3

....

....

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

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

(2)

....

....

....

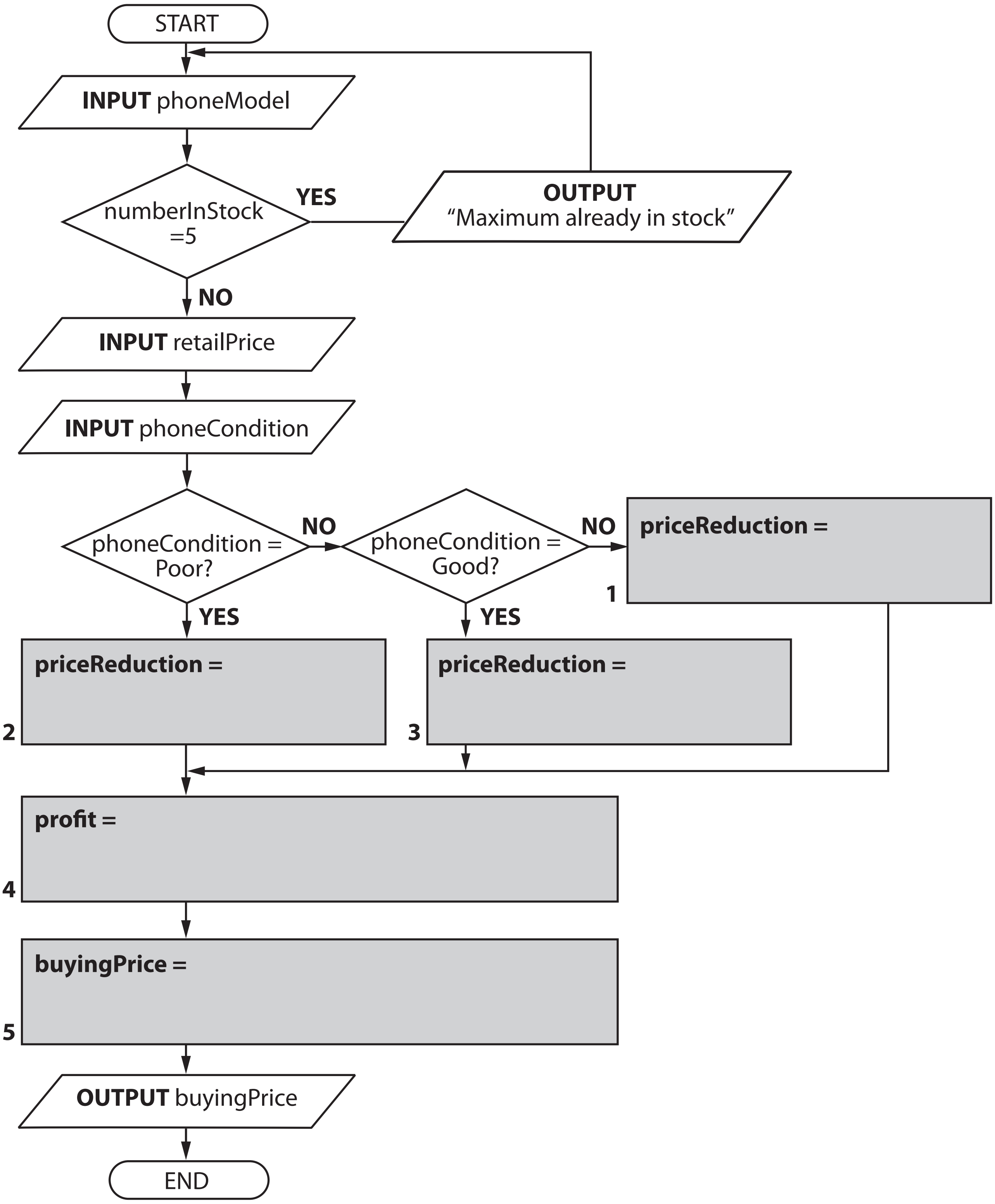
....

(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)



(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)

(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)

(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

....

....

Reason 2

....

....

Reason 3

....

....

(Total for Question 1 = 20 marks)

Please refer to the Information Booklet in order to answer Question 2.

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)

(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)

....

....

....

....

....

....

....

....

(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)

....

....

....

....

....

....

....

....

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

....

....

....

....

Reason 2

....

....

....

....

(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)

Middle Boundary 1..

Middle Boundary 2..

Middle Boundary 3..

(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)

(Total for Question 2 = 24 marks)

Please refer to the Information Booklet in order to answer Question 3.

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)

(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)

(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)

(Total for Question 3 = 20 marks)

Please refer to the Information Booklet in order to answer Question 4.

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)

(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]
```

(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)

(Total for Question 4 = 26 marks)

TOTAL FOR PAPER = 90 MARKS

BLANK PAGE

Pearson BTEC Level 3 Nationals Extended Certificate
Foundation Diploma, Diploma, Extended Diploma

Monday 3 June 2019

Morning | Paper Reference **31768H**

Computing

Unit 1: Principles of Computer Science

Information Booklet

Instructions

- You will need the information in this booklet to answer some questions.
- Read the information carefully.
- You must **not** write your answers in this booklet.
- Only your answers given on the question paper booklet will be marked.
- Do not return this Information Booklet with the question paper.

Turn over ▶

P61561A

1/1/1/1/1/1

P 6 1 5 6 1 A

Pearson

SECTION 1

The information in this section should be used to answer Question 1.

Figure 1 shows the rules the program will follow to calculate the buying price of a used mobile phone.

- The phone model is input.
- There should be no more than 5 phones of each model in stock.
  - If the number of phones in stock is 5 the program will print a message saying 'Maximum already in stock'.
  - If the number of phones in stock is fewer than 5, these rules are used to work out a buying price for a phone.
- The retail price of the phone, when new, is input.
- The condition of the phone is input. This can be 'Poor', 'Good' or 'Excellent'.
- A price reduction is then calculated. This will depend on the condition of the phone.

  The price reduction is calculated by dividing the retail price by 100 and then multiplying this value by the reduction percentage.

Phone condition	**Reduction percentage**
Poor	65%
Good	45%
Excellent	25%

- 20% profit is calculated. The profit is calculated by subtracting the price reduction from the retail price. This value is then divided by 100 and then multiplied by 20.
- The buying price is calculated by subtracting the price reduction and the profit amount from the retail price.
- The program should then output the buying price.

Figure 1

SECTION 2

The information in this section should be used to answer Question 2.

Figure 2a shows an algorithm that George uses to enter the names of students who have achieved a merit. It is written in pseudocode.

```
BEGIN
FOR count 1 to 3:
    INPUT firstname
    INPUT surname
    INPUT reason
OUTPUT("First name:"+(firstname)+"_Surname:"+(surname)+"_Reason:"+(reason[0:3]))
ENDFOR
END
```

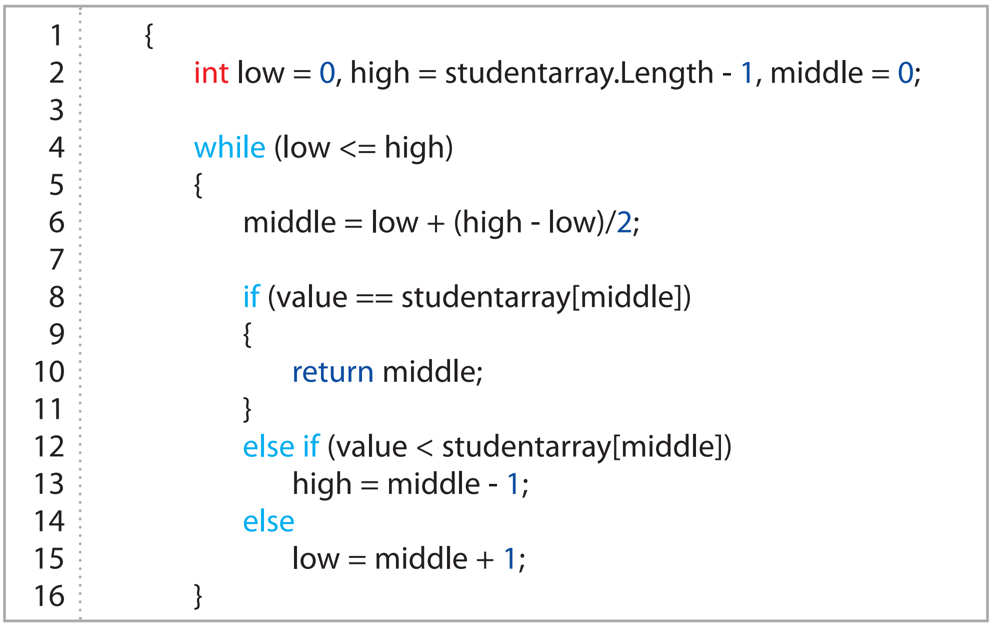
Figure 2a

Figure 2b shows an alternative algorithm George could use. It is written in pseudocode.

```
BEGIN
DEFINE FUNCTION record
    INPUT firstname
    INPUT surname
    INPUT reason
    OUTPUT("First name:"+(firstname)+"_Surname:"+(surname)+"_Reason:"+(reason[0:3]))
ENDFUNCTION
CALL FUNCTION record
CALL FUNCTION record
CALL FUNCTION record
END
```

Figure 2b

Figure 2c shows some programming code that will perform a binary search. It is written in C#.



```
{
    int low = 0, high = studentarray.Length - 1, middle = 0;

    while (low <= high)
    {
        middle = low + (high - low)/2;

        if (value == studentarray[middle])
        {
            return middle;
        }
        else if (value < studentarray[middle])
            high = middle - 1;
        else
            low = middle + 1;
    }
```

Figure 2c

SECTION 3

The information in this section should be used to answer Question 3.

Figure 3a shows the rules the code follows to work out how many litres of paint are needed.

1. The customer surname must be input.
2. The number of walls that need to be painted is input.
3. The height and the width of each wall, in feet, are input.
4. The number of square feet is calculated for each wall by multiplying the height by the width.
5. The total number of square feet for all walls is added together.
6. One litre of paint will cover 100 square feet. The number of litres of paint is calculated by dividing the number of square feet by 100.
7. The number of litres should always be rounded **upwards** to the nearest whole number.
8. Paint is sold in the following sizes of tin:
    - 1 litre
    - 2 litres
    - 5 litres.

    The program should calculate the number of each size of tin needed.
9. The following details are then output with suitable labels:
    - the total number of litres needed
    - the number of each size of tin needed (5 litres, 2 litres, 1 litre).

Figure 3a

Figure 3b shows the programming code that Madison has created.

It is written using Python.

```
#This section will allow the user to input their surname and the number of walls
customerSurname = input ("Enter customer surname")
while len (customerSurname) == 0:
    customerSurname = input ("You must enter a surname to continue")
wallNumber = input("Enter the number of walls that needs painting")
wallNumber = int (wallNumber)

#This section of code will calculate the total number of square feet
squareFeet = 0
totalSquareFeet = 0
for i in range (1,wallNumber):
    height = input ("Enter the height of the wall in feet")
    height = int (height)
    width = input("Enter the width of the wall in feet")
    width = int (width)
    squareFeet = (height)*(width)
    totalSquareFeet = totalSquareFeet+squareFeet

#This section will calculate the total number of litres required
litresRequired = (totalSquareFeet//100)
if not totalSquareFeet%100 == 0:
    litresRequired = litresRequired+1
print (litresRequired)

#This section will calculate the number of each paint tin required
def calculateTins():
    global litresRequired
    fiveLitres = 0
    twoLitres = 0
    oneLitre = 0
    while litresRequired >5:
        fiveLitres = fiveLitres+1
        litresRequired = litresRequired-5
    while litresRequired >2:
        twoLitres = twoLitres+1
        litresRequired = litresRequired-2
    while litresRequired >0:
        oneLitre = oneLitre+1
        litresRequired = litresRequired-1

#This section will output information to the user
    print("The total number of litres required is"+str(litresRequired))
    print(str(fiveLitres)+"tins of 5 litre paint")
    print(str(twoLitres)+"tins of 2 litre paint")
    print(str(oneLitre)+"tins of 1 litre paint")

#This section will run the CalculateTins function
calculateTins()
```

GUIDANCE: The operator // means integer division that will only show the values before the decimal point.

Figure 3b

SECTION 4

The information in this section should be used to answer Question 4.

Figure 4a gives the rules that are followed when making a booking for the dogs Callum has to take for a walk.

The rules that are followed when making a booking are:

1. The dog name is entered
2. The day is entered
3. The code will count the number of dogs already booked on that day.
4. The following 'walk criteria' are checked:
    - Is the **number of dogs booked** less than 4
    - Is the **Vet Registered** "Yes"
    - Is the **Updated Vaccinations** "Yes"
    - Is the **Dog Behaviour** less than 4.
5. If all the 'walk criteria' are met, the dog name will be added to the bookings and a message saying "Booking successful" will be printed.
6. If not, a suitable message will be printed to inform the user of the reason why the booking cannot be made.

Figure 4a

Figure 4b shows some sample data about the dogs that attend a group walk.

ID	Dog Name	Breed	Gender	House Number	Road Name	Vet Registered	Updated Vaccinations	Dog Behaviour
	Winston	Pug	Female	16b	Toll Road	Yes	Yes	1
	Winston	Labrador Retriever	Male	26	King Road	Yes	Yes	2
	Lexi	Beagle	Female	18	Park Lane	No	No	2
	Blu	Border Collie	Male	19	Old Street	Yes	Yes	2
	Joel	Patterdale Terrier	Male	1a	Leake Street	Yes	Yes	3
	Daisy	German Shepherd	Female	157	Bond Street	Yes	Yes	4
	Rufus	Yorkshire Terrier	Male	289	Church Street	No	No	3
	Angus	Jack Russell	Male	12	Lord Road	Yes	Yes	2

Callum wants to create a unique identifier that cannot be repeated for each dog.

1 = Good

2 = Fair

3 = Satisfactory

4 = Aggressive

Figure 4b