



Name: \_\_\_\_\_ EXAM PAPERS PRACTICE

Computational thinking

**Date:**

**Time:**

**Total marks available:**

**Total marks achieved:** \_\_\_\_\_

EXAM PAPERS PRACTICE  
© 2026 Exam Papers Practice. All Rights Reserved

## Questions

Q1.

### **Computational thinking**

A bubble sort is carried out on this list to put it in ascending order.

8 3 2 4 0 3 9

The value '8' starts in position 0.

(i) State the number of passes required to complete the sort.

(1)

.....

(ii) State the number of swaps made on the final pass.

(1)

.....

(iii) State the component of an algorithm used to store whether a swap has been made during a pass.

(1)

.....

(iv) State the position of the item that will be compared with the value in position 0.

(1)

.....

(v) Define the term 'iteration'.

(1)

.....

.....

**(Total for question = 5 marks)**

Q2.

Here is an algorithm that prints colours.



```
1 # ----- Global variables -----
2 inkColours = ["Cyan", "Magenta", "Yellow", "Black"]
3
4 # ----- Subprograms -----
5 def displayAll (inList):
6     for index in range (0, len (inList)):
7         print (index, inList[index])
8
9 # ----- Main Program -----
10 userInput = input ("Would you like to see the colours? ")
11 if ((userInput == "y") or (userInput == "Y")):
12     displayAll (inkColours)
13 else:
14     print ("Thank you")
```

(i) State the type of data structure used to hold the ink colours.

(1)

(ii) Give the contents of `inkColours[2]`

(1)

(iii) State the name of the computational thinking technique used by the subprogram `displayAll()` to hide the logic of printing.

(1)

(iv) State the name of the computational thinking technique used when separating logic into different blocks, such as the subprogram and the main program.

(1)

**(Total for question = 4 marks)**

Q3.

## Computational thinking

Here is an algorithm that uses colours.

```
1 # -----
2 # Global variables
3 # -----
4 theColours = ["Green", "Blue", "Yellow", "Red", "Purple"]
5 colour = ""
6
7 # -----
8 # Main program
9 # -----
10
11 for item in theColours:
12     print (item)
13
14 colour = input ("Enter a colour: ")
15 while (colour != ""):
16     if (colour == "Green"):
17         print ("Green is my favourite colour")
18     else:
19         print (colour + " is a good colour")
20
21     colour = input ("Enter a colour: ")
```

(i) Give the first line number of a condition-controlled loop.

(1)

.....

(ii) Give the first line number of iteration over every item in a data structure.

(1)

.....

(iii) Give the line numbers of a selection.

(1)

.....

**(Total for question = 3 marks)**

Q4.

**Computational thinking**

Programmers consider algorithm efficiency when they write code.

(i) Sorting and searching use algorithms.

Complete the table with the name of a search algorithm and a sort algorithm.

(2)

Algorithm type	Characteristic	Algorithm name
Search	Is a divide and conquer algorithm	
Sort	Is <b>not</b> a divide and conquer algorithm	

(ii) Explain **one** effect the number of comparisons has on the execution time of a sorting algorithm.

(2)

.....

.....

.....

.....

**(Total for question = 4 marks)**

Q5.

**Computational thinking**

Algorithms use relational and arithmetic operators.

(i) Here is a relational operator used in a conditional test.



count > index

State the **two** different results of evaluating a conditional test.

(2)

1 .....

2 .....

(ii) Identify the result of  $5 // 2$

(1)

**A** 0.5

**B** 1

**C** 2

**D** 2.5

**(Total for question = 3 marks)**

Q6.

**Computational thinking**

Here is an algorithm to process sales data.



```
1 # -----
2 # Global variables
3 # -----
4
5 salesData = [[9, 6, 32],
6              [20, 12, 9],
7              [9, 8, 8],
8              [32, 24, 29],
9              [7, 20, 12],
10             [6, 9, 6]]
11 total = 0
12 currMax = 0
13
14 # -----
15 # Main program
16 # -----
17 for sale in salesData:
18     total = 0
19     for item in sale:
20         total = total + item
21         if (item < currMax):
22             currMax = item
23     print (sale, total)
24 print (currMax)
25
```

(i) The data structure `salesData` is shown as a table of records, with fields in each record.

Complete the table with the line number for each traversal.

(2)

	Line number
Traverses the records	
Traverses the fields	

(ii) There is a logic error in the code.

Write the corrected line of code.

(1)

.....

.....

Q7.

An algorithm allows users to enter a whole number.

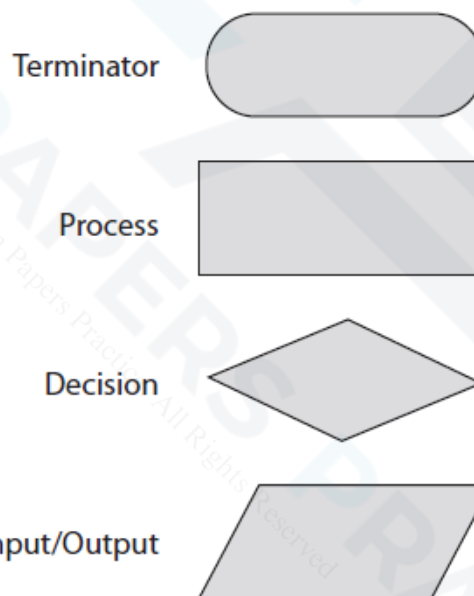
The number can be positive or negative.

The purpose of the algorithm is to report whether the number is even or odd.

The modulus function returns the remainder after division.

The algorithm can be expressed as a flowchart.

Here are some flowchart symbols:



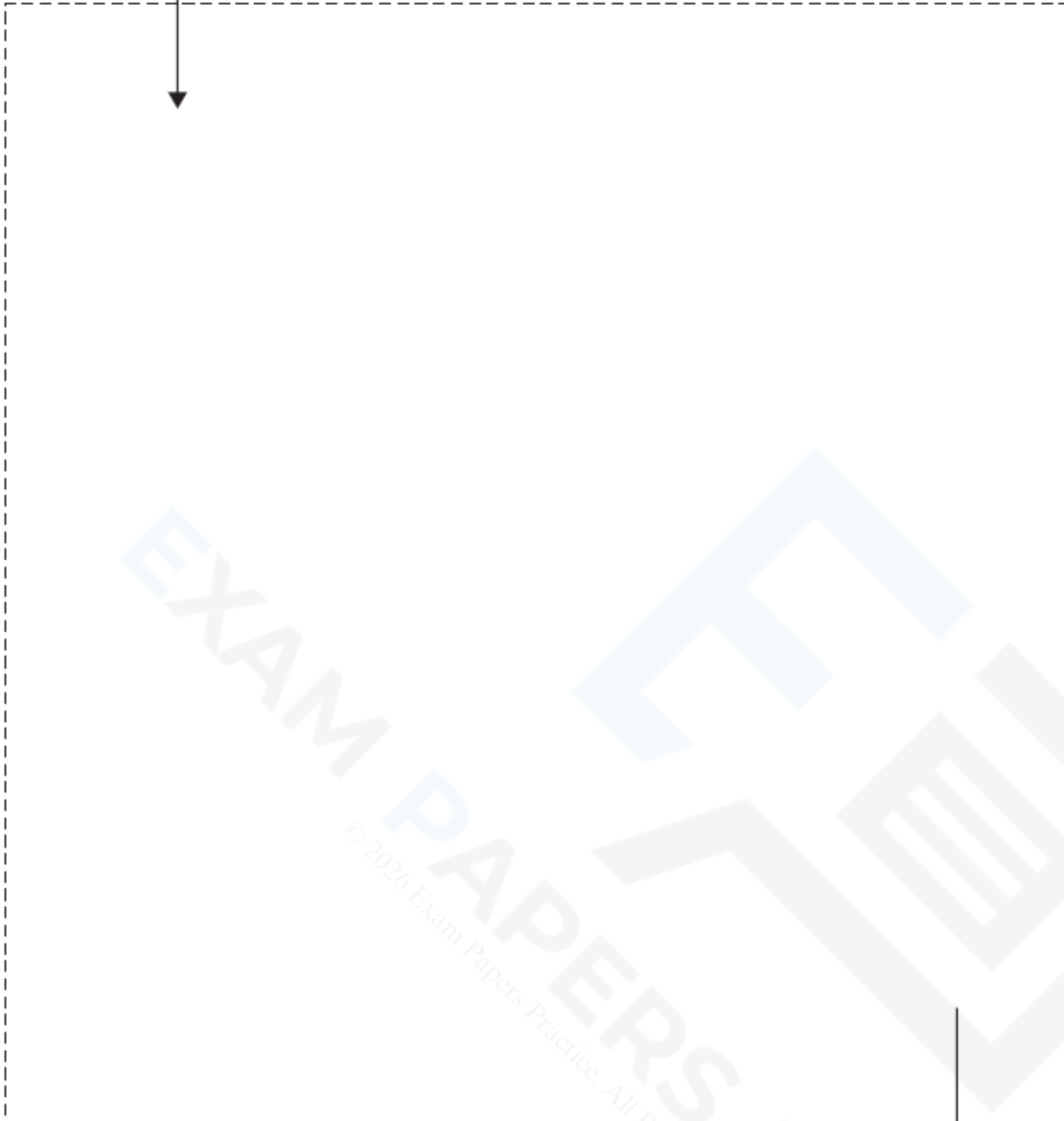
Complete the flowchart to show the algorithm.

You may not need to use all the flowchart symbols.

(4)



Start



Stop

EXAM PAPERS PRACTICE  
© 2026 Exam Papers Practice. All rights Reserved

**(Total for question = 4 marks)**

Q8.

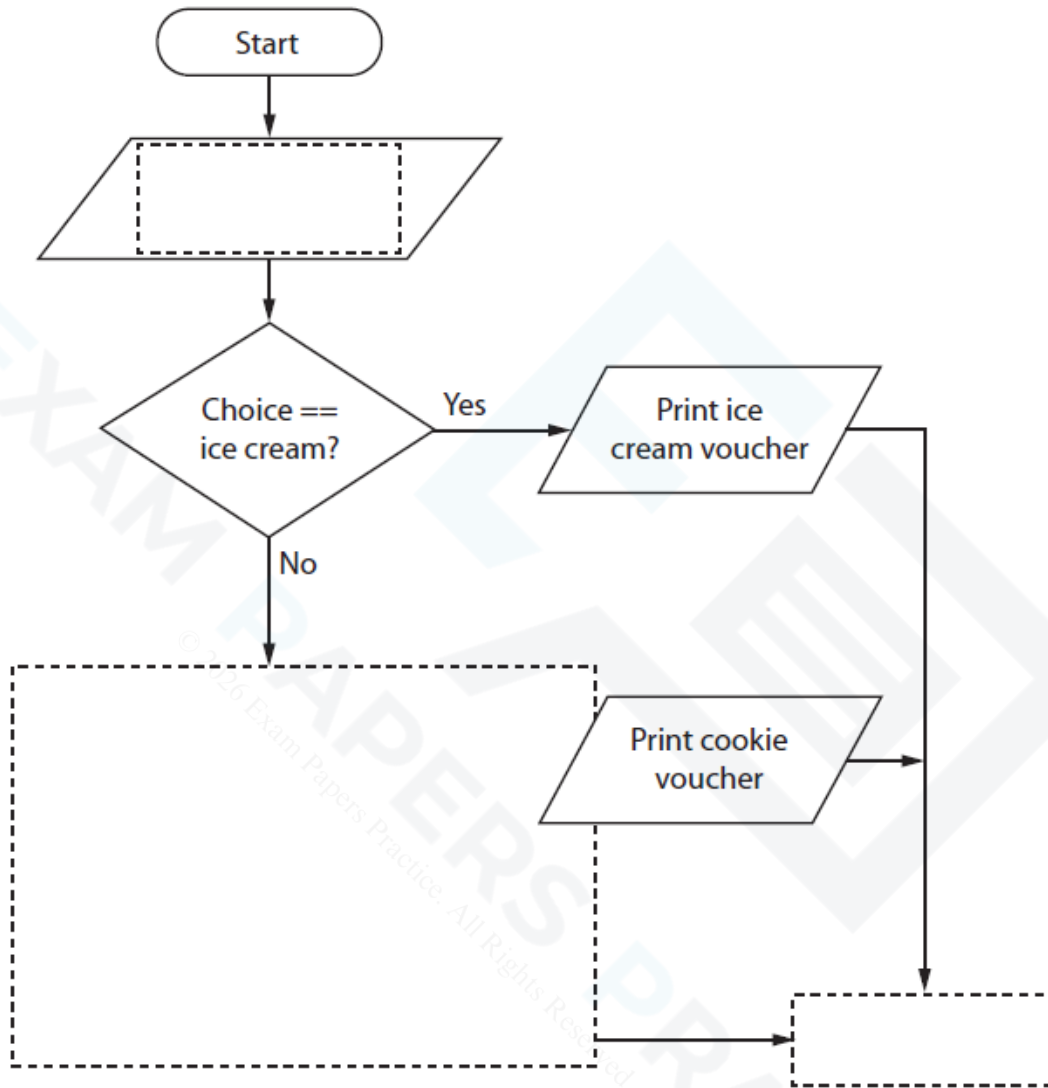
**Computational thinking**

Customers can get a voucher for their favourite item.

- Customers whose favourite item is ice cream get a voucher for ice cream.
- Customers whose favourite item is cookies get a voucher for cookies.
- Customers who do not choose either get a voucher for drinks.

Complete the flowchart to show this process.

(6)



**(Total for question = 6 marks)**

Q9.

Here is an algorithm.



```
1 myNumber = 0
2
3 myNumber = int (input ("Enter a whole number between 1 and 100:"))
4
5 if (myNumber < 1):
6     print ("Too low")
7 elif (myNumber > 100):
8     print ("Too high")
9 elif (myNumber % 10 == 0):
10    print ("Nice round number")
11 elif (myNumber == 100):
12    print ("That's the biggest number")
13 else:
14    print ("Good choice")
```

Complete the table to show the output for the given input.

(4)

Input	Output
200	
33	
100	
0	

**(Total for question = 4 marks)**

Q10.

Here is an algorithm.



```
1 today = ""
2
3 today = int (input ("Day of the week (1-7)"))
4
5 if ((today == 6) or (today == 7)):
6     print ("8am")
7 elif ((today >= 1) and (today <= 3)):
8     print ("7:30am")
9 else:
10    print ("7am")
```

Complete the table to show the output for the given input.

(3)

Input	Output
2	
7	
8	

(Total for question = 3 marks)

Q11.

**Computational thinking**

A binary search algorithm looks for a target in a sorted array.

Here is an array of numbers.

Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
Value	11	22	33	44	55	66	77	88	99	100	101

Complete the table to show the steps of a binary search algorithm to look for the number **78**, which is not in the array.



You must show the calculation of the mid-point.

You may not need to fill in all the rows in the table.

(6)

Start index	End index	Calculation of mid-point	Found Y or N	Discard lower, higher or none

(Total for question = 6 marks)

Q12.

**Computational thinking**

A binary search algorithm is used with this list to find the target value 'b'.

a b c d e f g h i j k

Complete the table to show the **three** characters in the order that the algorithm would compare them against the target value.

(3)



First	
Second	
Third	
Fourth	b

(Total for question = 3 marks)

Q13.

**Computational thinking**

Here is an algorithm.

```
1 # -----
2 # Constants
3 # -----
4
5 NUM = 6
6
7 # -----
8 # Global variables
9 # -----
10
11 total = 0
12
13 # -----
14 # Main program
15 # -----
16
17 for number in range (1, NUM + 1):
18     remainder = NUM % number
19     if (remainder == 0):
20         total = total + number
21 print (total)
22
```

Complete the trace table.

You may not need to fill in all the rows in the table.

number	remainder	total	Output
		0	

(Total for question = 4 marks)

Q14.

Laura owns a fruit shop.

This program checks the weight of boxes of strawberries from Laura's shop.

```

1   count = 0
2   weight = 0
3   accept = 0
4   reject = 0
5
6   while (count < 4):
7       count = count + 1
8       weight = int (input ("Enter weight of box: "))
9       if ((weight < 395) or (weight > 405)):
10          reject = reject + 1
11      else:
12          accept = accept + 1
13
14  print (accept, reject)

```

The inputs are 404, 393, 395, 405.





### Computational thinking

Margaret owns an ice-cream shop.

This program manipulates sales figures from Margaret's shop.

```
2 num = 0
3 x = 999
4 y = 0
5 line = ""
6
7 f = open("SalesFile.txt", "r")
8 for line in f:
9     num = int(line)
10    if num < x:
11        x = num
12    if num > y:
13        y = num
14 print(x, y)
15 f.close()
```

The only inputs from the file to the program are 355, 554, 199 and 409.

Complete the trace table showing the execution of the program with these four inputs.

You may not need to fill in all the rows in the table.

(6)

num	x	y	Display

(Total for question = 6 marks)

Q16.

**Computational thinking**

Complete this truth table.

(3)

A	B	C	(A AND B)	(NOT C)	(A AND B) OR (NOT C)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

**(Total for question = 3 marks)**

Q17.

Algorithms control physical devices using logical operators.

A security system turns on a floodlight when the sunlight falls below a certain level (S) and a movement sensor is activated (M).

Complete the truth table.

(2)



EXAM PAPERS PRACTICE

S	M	S AND M

(Total for question = 2 marks)

Q18.

Complete the truth table for the logical expression NOT (A OR B).

(2)

A	B	A OR B	NOT (A OR B)
0	0		
0	1		
1	0		
1	1		

(Total for question = 2 marks)

Q19.

## Computational thinking

Here is a partially completed truth table.

The logical expression for **Q** is complete.

Complete the truth table to give the:

- logical expression for **S**
- logical expression for **T**
- values for **Q**

(3)

A	B	C	S	T	Q
					NOT S AND T
0	0	0	0	0	
0	0	1	0	1	
0	1	0	0	1	
0	1	1	0	1	
1	0	0	0	0	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	1	1	

(Total for question = 3 marks)



Q20.

**Computational thinking**

Programs can have syntax errors and runtime errors.

(i) Define the term 'syntax error'.

(1)

.....  
.....

(ii) Runtime errors happen when a program is executing.

Explain a runtime error.

(2)

.....  
.....  
.....  
.....  
.....  
.....

**(Total for question = 3 marks)**

Q21.

Programmers use abstraction to model the real world.

Define the term 'abstraction'.

(2)

.....  
.....  
.....  
.....

**(Total for question = 2 marks)**

Q22.

**Computational thinking**

Define the term 'decomposition'.

**(1)**

.....

.....

**(Total for question = 1 mark)**

Q23.

Define the term 'subprogram'.

**(2)**

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q24.



Algorithms use arrays and records to hold data.

Describe a record.

(2)

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q25.

A linear search algorithm can be used on both a sorted and an unsorted array.

Describe how a linear search algorithm operates on an **unsorted** array.

(4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

**(Total for question = 4 marks)**



Q26.

This algorithm searches a sorted array of numbers for a target value. The target value may or may not be in the array.

```
18 while ((index < len (theArray)) and (not found) and (not passed)):  
19     if (theArray[index] == theTarget):  
20         found = True  
21         location = index  
22     elif (theArray[index] > theTarget):  
23         passed = True  
24         index = index + 1
```

The use of the found variable helps to make the algorithm efficient.

Describe how the use of the passed variable also helps to make the algorithm efficient.

(2)

.....  
.....  
.....  
.....

**(Total for question = 2 marks)**

Q27.

Describe **one** difference between a syntax error and a logic error.

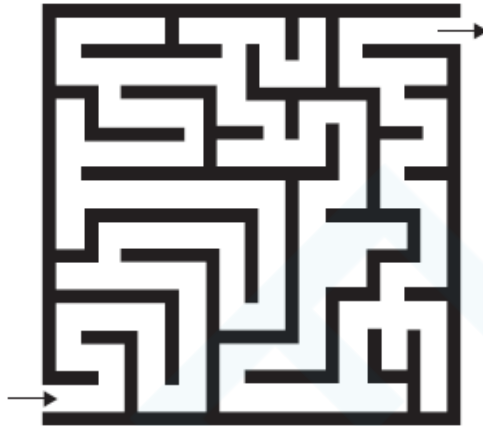
(2)

.....  
.....  
.....  
.....

**(Total for question = 2 marks)**

Q28.

A group of students are working together on a single maze game. The arrow keys control the character. When the character reaches the end of the maze without touching a wall, a happy sound is played. The game also displays a score.



Discuss the use of decomposition and abstraction in developing this game.

Your answer should include:

- a definition of each term
- the benefits each brings to the group of students
- an example of where each could appear in the program code.

(6)

**(Total for question = 6 marks)**

Q29.

An algorithm is required that allows a user to enter two numbers. The algorithm then informs the user which number is greater, or whether the two numbers are equal. The algorithm is expressed in a flowchart.

Here are some flowchart symbols:



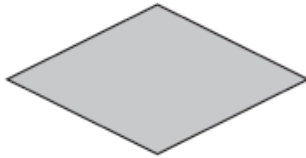
Terminator



Process



Decision



Input/Output



Draw a flowchart to show this algorithm.

(6)

(Total for question = 6 marks)

Q30.

Here is a set of flowchart symbols for an algorithm about flavours.



Draw a flowchart to show the completed algorithm.

Use every symbol exactly once. Use as many arrows and labels as you require. Do **not** add any additional symbols.

(4)



**(Total for question = 4 marks)**

Q31.

**Computational thinking**

Explain **one** benefit of using subprograms.

(2)

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q32.

**Computational thinking**

The identifier `plants` is used for an array of values.

`len(plants) // 2` is used to find the index position of the middle item in `plants`

Explain **one** reason why integer division, rather than division, is used to do this.

(2)

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q33.

Programmers use trace tables with algorithms.

Explain the purpose of a trace table.

(2)

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q34.

**Computational thinking**

The efficiency of algorithms can be analysed.

The worst case is when the most comparisons are required.

The best case is when the fewest comparisons are required.

(i) Explain the worst case for a linear search on an array.

(2)

.....

.....

.....

.....

.....

.....

.....

(ii) Explain the best case for a bubble sort on an array.

(2)

.....

.....

.....

.....

.....

.....

**(Total for question = 4 marks)**

Q35.

**Computational thinking**

A constant is shown here on line 3.

Programmers can use all capitals to show that a value is a constant.

```
2 # Prototype for the main swimming pool
3 MAX_CAPACITY = 120      # Maximum number of swimmers
4
5 numAdult = 14           # Current number of adults swimming
6 numChild = 73          # Current number of children swimming
```

Explain **one** reason why programmers use signals indicating a value is a constant, rather than repeating the same fixed value throughout an algorithm.

(2)

.....

.....

.....

.....

**(Total for question = 2 marks)**

Q36.

**Computational thinking**

Identify the term that means breaking a problem or solution down into smaller parts.

(1)

- A** Abstraction
- B** Computation
- C** Decomposition
- D** Evaluation

(Total for question = 1 mark)

Q37.

Programmers use different types of operators in their programs.

Name the **type** of operator for each example.

(3)

+, \* .....

AND, NOT .....

(Total for question = 3 marks)

Q38.

**Answer the question with a cross in the box you think is correct . If you change your mind about an answer, put a line through the box  and then mark your new answer with a cross .**

An algorithm is designed for a 'guess the roll of a dice' game. The player decides the number of sides on a single dice, between 6 and 12, and guesses the number rolled.



© PAL

(i) Name **two** inputs required by the algorithm.

(2)



1 .....

.....

2 .....

.....

(ii) The player of the game can have three guesses. The variable roll is set to 1 at the start of the game. One is added to roll after each guess.

Identify the statement with the correct relational operator used to check whether the player can have another guess.

(1)

- A roll = 3
- B roll == 3
- C roll > 3
- D roll <= 3

(Total for question = 3 marks)

Q39.

**Computational thinking**

Here are four images of creatures that will be used in a computer game.



(Source: images from © PAL)

Before coding the game, a programmer applies abstraction.

One feature of creatures is their colour.

State **two other** features of the creatures that the programmer could include when creating a general model for a creature.

(2)

1 .....

.....

2 .....

.....

(Total for question = 2 marks)

Q40.

**Computational thinking**

State the type of error that can be found in **algorithms**.

(1)

.....



**(Total for question = 1 mark)**

