



Please write clearly, in **BLOCK CAPITALS** and black ink

Centre number _____ Candidate number _____

Forename(s) _____

Surname _____

Date of Exam _____

Time allowed: 2 hours

GCSE Computer Science

Paper 1: Computational thinking and programming skills

Total Marks

PAPER 1A

Instructions

- Write in black ink or black ball-point pen. Use pencil only for drawing.
- Write your answer to each question in the space provided
- Answer all questions
- Do all rough work in this book
- Cross through any work you do not want marked
- **Questions that require a coded solution must be answered in Python 3**
- **You are not allowed to use a calculator**

Information

- The total mark for this paper is **90**
- The student version of this paper has **23** pages

Advice

- For multiple-choice questions, completely fill in the lozenge next to the answer you want to select.

CORRECT METHOD



INCORRECT METHOD



- To change your answer, cross out your original answer like this:
- If you want to go back to an answer you previously crossed out, circle the answer you now want to select like this:





Answer ALL questions.

0 1 . 1 Define the term decomposition.

[2 marks]

0 2 The pseudo-code in **Figure 1** assigns values to variables. The values represent the number of whole pounds that a person earns.

Figure 1

```
salary ← 16  
bonus ← 2  
salary ← salary + bonus
```

0 2 . 1 Shade **one** lozenge that shows the value stored in the variable salary once the program in **Figure 1** has finished running.

[1 mark]

- | | | |
|---|----|-----------------------|
| A | 2 | <input type="radio"/> |
| B | 16 | <input type="radio"/> |
| C | 18 | <input type="radio"/> |
| D | 32 | <input type="radio"/> |

0 2 . 2 Shade **one** lozenge that shows the data type for the variable salary in **Figure 1**.

[1 mark]

- | | | |
|---|-----------|-----------------------|
| A | Boolean | <input type="radio"/> |
| B | character | <input type="radio"/> |
| C | integer | <input type="radio"/> |
| D | real | <input type="radio"/> |



0 3

The algorithm in **Figure 2** is used to calculate the area of a circle according to the formula:

$$area = \pi r^2$$

The radius entered by the user must be between 1 and 100.

- Line numbers are included but are not part of the algorithm.

Figure 2

```
1  CONSTANT PI ← 3.141
2  SUBROUTINE CIRCLE_AREA(r)
3      area ← PI * r * r
4      RETURN area
5  ENDSUBROUTINE
6  radius ← STRING_TO_REAL(USERINPUT)
7  IF radius ≥ 1 OR radius ≤ 100 THEN
8      result ← REAL_TO_STRING(CIRCLE_AREA(radius))
9      OUTPUT result
9  ENDIF
```

0 3

1

Shade **one** lozenge which explains the meaning of the relational operator \geq in line 7 in **Figure 2**.

[1 mark]

- A Greater than ☐
- B Greater than or equal to ☐
- C Less than ☐
- D Less than or equal to ☐

0 3

2

Shade **one** lozenge which shows the line number where the subroutine is called in **Figure 2**.

[1 mark]

- A Line 2 ☐
- B Line 4 ☐
- C Line 5 ☐
- D Line 8 ☐



0 3 . **3** Shade **one** lozenge which shows the data type of the variable radius in **Figure 2**.

[1 mark]

- A** Character ☐
- B** Integer ☐
- C** Real ☐
- D** String ☐

0 3 . **4** Shade **one** lozenge which shows the value stored in the variable result when CIRCLE_AREA returns the value 12.56 in **Figure 2**.

[1 mark]

- A** 12.56 ☐
- B** 12 ☐
- C** 13 ☐
- D** '12.56' ☐

0 3 . **5** State the reason that PI (π) is stored as a constant rather than a variable in **Figure 2**.

[1 mark]

0 3 . **6** The variable area is a local variable.

Explain why the use of a local variable is more appropriate than a global one in this situation.

[2 marks]



0 4

A programmer has written a Python program that asks the user to input a number. The program then calculates the times table for the number input for 10 rows. For example, if the user enters 5, the program will output:

```
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
4 x 5 = 20
5 x 5 = 25
6 x 5 = 30
7 x 5 = 35
8 x 5 = 40
9 x 5 = 45
10 x 5 = 50
```

Complete the program below by filling in the gaps using the items in **Figure 3**. You will not need to use all the items in **Figure 3**. Each item in **Figure 3** should only be used once.

[5 marks]

Figure 3

0,10	0,9	1,10	1,11
print	output	if	elif
while	for	str	=
int	float	*	
+	x		

```
timesTable = _____ (input("Enter times table: "))
```

```
_____ i in range(_____):
```

```
    result = timesTable _____ i
```

```
    _____(i, "x", timesTable, "=", result)
```



0 5

The algorithm in **Figure 4** is a searching algorithm.

- Array indexing starts at 0.
- Line numbers are included but are not part of the algorithm.

Figure 4

```
1  names ← ["Amy", "Ava", "Joe", "Tim"]
2  OUTPUT "Enter name to search for: "
3  item ← USERINPUT
4  found ← False
5  i ← 0
6  WHILE NOT found AND i < LEN(names)
7      IF names[i] = item THEN
8          found ← true
9          OUTPUT names[i]
10         ENDIF
11         i ← i + 1
12 ENDWHILE
13 IF NOT found THEN
14     OUTPUT "Name not found"
15 ENDIF
```

0 5

. 1

State the data structure that is used to store the names in the variable names in the algorithm shown in **Figure 4**.

[1 mark]

0 5

. 2

Shade **one** lozenge to show which of the following contains the **false** statement about the algorithm in **Figure 4**.

[1 mark]

- A** The algorithm uses Boolean operators ☐
- B** The algorithm uses definite iteration ☐
- C** The algorithm uses selection ☐



- 0 5** . **3** Complete the trace table for the algorithm shown in **Figure 4** when the user enters "Joe". Some values have already been entered.

[6 marks]

names				item	found	i	OUTPUT
[0]	[1]	[2]	[3]				
"Amy"	"Ava"	"Joe"	"Tim"	"Joe"	False		

- 0 5** . **4** The algorithm is for a linear search.
State **one** advantage of using a linear search rather than a binary search.

[1 mark]

- 0 6** Write a Python program that converts a temperature in Centigrade to the equivalent temperature in Fahrenheit.

The program should:

- Allow the user to enter the temperature in Centigrade (no validation is required)
- Validate that the temperature entered is between 0 and 100 Centigrade (no other validation is required and both 0 and 100 are to be treated as valid temperatures)
 - Convert the temperature to Fahrenheit by the following formula:
Fahrenheit = Centigrade x (9 ÷ 5) + 32
- If a valid temperature is entered, output the converted temperature in Fahrenheit
- Otherwise output "Temperature must be between 0 and 100"
 - The program does not need to ask the user to enter the temperature again if they enter an invalid number

You should use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[7 marks]

[illegible]



0 7

Write a Python program that allows the user to input a word. The program then outputs each character in the word with the corresponding position starting at 1. For example, if the user enters "Hello", the program will output:

```
1 H
2 e
3 l
4 l
5 o
```

Your program should work as follows:

- Gets the user to enter a string and store it in a suitable variable
- Output each character along with its position in the string. The positions output should start at 1

You should use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[8 marks]





0 8

Four in a row is a board game which has a grid with a width of seven holes and height of six holes.

Two players take turns to place a piece in one of the seven columns. One player is yellow, whilst the other player is red.

The following subroutines control the dropping of pieces into a column.

DROP(colour, column)	Drops one piece of colour 'R' or 'Y' in column number 0 to 6.
HEIGHT(column)	Returns the number of pieces currently in the column as an integer between 0 and 6.
FULL(column)	returns True if the column is full, False if the column is empty or partially full

The game has started and three moves have been made.

	0	1	2	3	4	5	6
5							
4							
3							
2							
1				'R'			
0			'R'	'Y'			

0 8

1

State a suitable data structure that could be used to store the grid.

[1 mark]

0 8

2

State the value that will be returned with the following statements.

[2 marks]

HEIGHT(3) _____

FULL(3) _____

0 | 8

3

Draw the grid after the following sequence has run.

```
DROP( 'Y', 2)
```

```
DROP( 'R', 3)
```

DROP('Y', 2)

	0	1	2	3	4	5	6
5							
4							
3							
2							
1				'R'			
0			'R'	'Y'			

[3 marks]

08

4

Develop an algorithm using pseudo-code that will check if a column is full before dropping a piece into it.

Your algorithm will get the input from the user for both the colour and the column they want to drop their piece into. The piece will either be dropped in the column, or if there isn't space, the algorithm will tell the user to try a different column.

[5 marks]

[illegible]

Develop an algorithm which will determine if the game is a draw by checking each column, using iteration, to see if it is full. If all columns are full, "Draw" will be displayed.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



09 . 1

A farmer has a computer-controlled tractor. The tractor needs to plough a field that has been divided into squares. The following instructions are available:

- Forward(*n*) moves the tractor *n* squares forward
- TurnLeft(*deg*) turns the robot *deg* degrees left (anti-clockwise)

Draw the path of the tractor through the grid below if the following program is executed (the tractor starts in the square marked by the \longrightarrow facing in the direction of the arrow).

Forward(4)

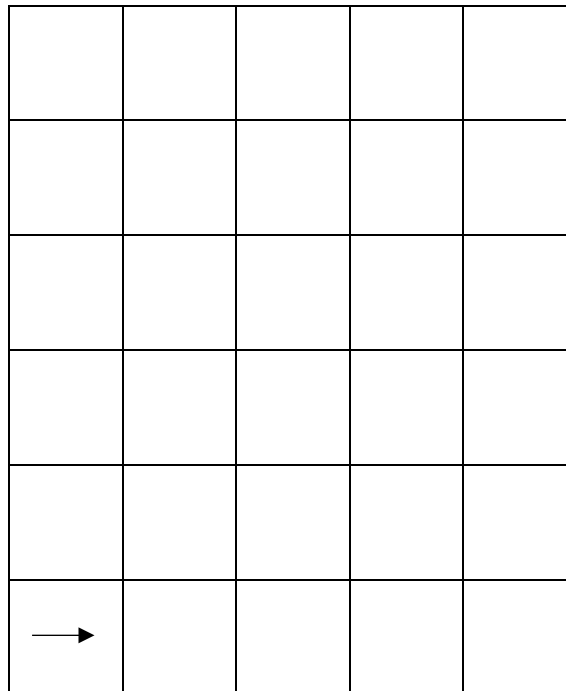
TurnLeft(90)

Forward(1)

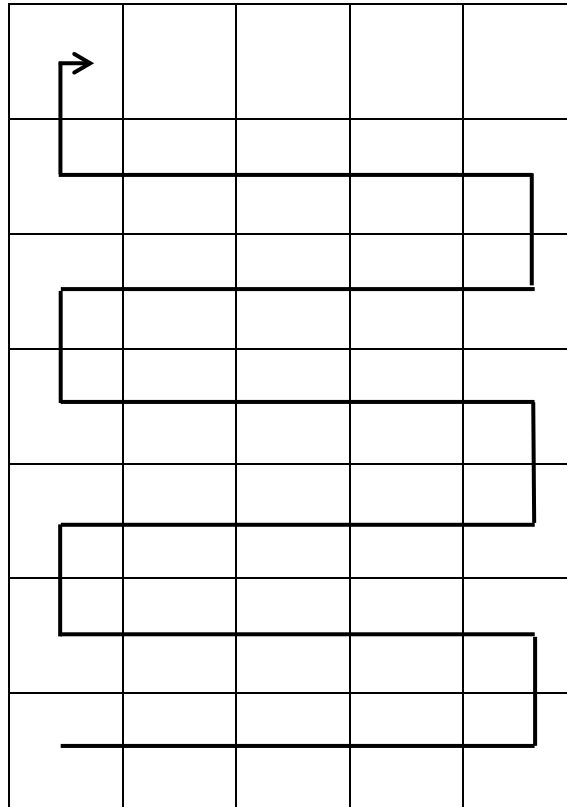
TurnLeft(90)

Forward(4)

[3 marks]



Develop an algorithm using pseudo-code which will plough the whole field as indicated by the arrow on the grid. The algorithm must use iteration. Note that the tractor can use the TurnLeft subroutine to make a turn.

[illegible]



1	0
---	---

Below is an array of numbers.

5	6	4	2	3	7	8	1
---	---	---	---	---	---	---	---

Describe the stages required to sort this array using a merge sort algorithm.

[5 marks]



1 1

The following Python program asks the user to input a sentence that must be at least three characters long and then works out the number of words it contains. The first part of the program validates the sentence entered and is shown in **Figure 5**.

Figure 5

```
validSentence = False
sentence = ""

while not validSentence:
    sentence = input("Enter sentence: ")
    if not len(sentence) >= 3:
        print("Invalid choice")
    else:
        print("Valid choice")
        validSentence = True
```

1 1

. 1

Shade **one** lozenge which shows invalid data that could be input to the program in **Figure 5**.

[1 mark]

- A The dog. ☐
- B The dog ☐
- C I. ☐
- D Me. ☐



1 1 . 2

Complete the following test plan for the code shown in Figure 5.

[2 marks]

Test type	Test data	Expected result
Normal data	Apple	Valid choice message displayed
Invalid data		
Boundary data		



- 1 1 . 3** The remainder of the program, in **Figure 6**, determines the number of words that have been entered by the user and stored in the variable named sentence.

Figure 6

```
spaceCount = 0
for letter in sentence:
    if letter == " ":
        spaceCount = 1
totalWords = spaceCount

print("Number of words:", spaceCount)
```

The program works by counting the number of spaces between each word in the sentence entered.

The program currently always reports that 1 word has been entered no matter how many characters or spaces were entered.

The program contains three lines of code that each contain one error.

Using Python only, rewrite the code in the space below to correct the three errors.

[3 marks]



1 2

Write a Python program that asks the user to enter the number of whole seconds they took in a race. The program then outputs the number of minutes and seconds the race took them. The format of the time must be mm:ss (two digits for the minutes and two digits for the seconds).

Example 1, if the user enters 127 seconds, the program will output 02:07

Example 2, if the user enters 180 seconds, the program will output 03:00

Example 3, if the user enters 117 seconds, the program will output 01:57

The program should:

- Ask the user to enter the number of seconds (there is no need to validate this number)
- Calculate the number of minutes that were required by performing an integer division on the seconds by 60
- Calculate the number of seconds that were required by using modulus
- If the minutes are less than 10, concatenate a leading zero
- If the seconds are less than 10, concatenate a leading zero
- Output the minutes and seconds with a colon between them

You should use meaningful variable name(s), correct syntax and indentation in your answer.

The answer grid below contains vertical lines to help you indent your code accurately.

[8 marks]





1 3

A Python program has been written that creates 5 random integers between 1 and 100 and stores them in a list. The program is shown in **Figure 7**. The function generateRandom returns a random number between 1 and 100.

Figure 7

```
import random

randomNumbers = [0]*5

for i in range(0,5):
    randomNumbers[i] = generateRandom()

print(randomNumbers)
```

1 3

1

State the purpose of the instruction.

```
import random
```

[1 mark]

1 3

2

State the contents of randomNumbers after the instruction.

```
randomNumbers = [0]*5
```

[1 mark]



1 3 . **3** The subroutine `generateRandom` has not yet been written.

Write Python code for the subroutine `generateRandom`. The subroutine will be a function that has no arguments and returns a random integer between 1 and 100.

[5 marks]
