**AQA**

**Surname** _____

**Forename(s)** _____

**Centre Number** _____

**Candidate Number** _____

**Candidate Signature** _____

I declare this is my own work.

# GCSE

# COMPUTER SCIENCE

**Paper 1  Computational thinking and programming skills – Python**

# 8525/1B

**Friday 19 May 2023          Afternoon**

**Time allowed: 2 hours**

**At the top of the page, write your surname and forename(s), your centre number, your candidate number and add your signature.**

**[Turn over]**

**MATERIALS**

For this paper you must have:

- **the Diagram Booklet.**

You must **NOT** use a calculator.

## INSTRUCTIONS

- Use black ink or black ball-point pen. Use pencil only for drawing.

- Answer ALL questions.

- You must answer the questions in the spaces provided.

- If you need extra space for your answer(s), use the lined pages at the end of this book. Write the question number against your answer(s).

- Do all rough work in this book. Cross through any work you do not want to be marked.

- Questions that require a coded solution must be answered in Python.

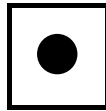- You should assume that all indexing in code starts at 0 unless stated otherwise.

**INFORMATION**

The total number of marks available for this paper is 90.
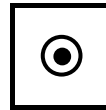
**ADVICE**

For the multiple-choice questions, completely fill in the lozenge alongside the appropriate answer.
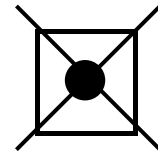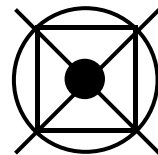
CORRECT  METHOD

WRONG  METHODS

If you want to change your answer you must cross out your original answer as shown.

If you wish to return to an answer previously crossed out, ring the answer you now wish to select as shown.

**DO  NOT  TURN  OVER  UNTIL  TOLD  TO  DO  SO**

**Answer ALL questions.**

| 0 | 1 |
|---|---|

**FIGURE 1, provided in the Diagram Booklet, shows an algorithm, represented using pseudo-code, which assigns a different value to four variables.**

| 0 | 1 | . | 1 |
|---|---|---|---|

**Define the term ALGORITHM.  [2 marks]**

_____

_____

_____

_____

0 1 . 2

The variable $x$ is assigned a value using the statement:

```
x ← LEN(state)
```

Using FIGURE 1, provided in the Diagram Booklet, what is the value of $x$?

Shade ONE lozenge.  [1 mark]

○ **A** 1

○ **B** 5

○ **C** 10

○ **D** 12

**[Turn over]**

| 0 | 1 | . | 3 |

**What is the result of concatenating the contents of the variables** `city` **and** `landmark` **in FIGURE 1?**

**Shade ONE lozenge. [1 mark]**

○ **A** `San Francisco Alcatraz Island`

○ **B** `San Francisco,Alcatraz Island`

○ **C** `San Francisco, Alcatraz Island`

○ **D** `San FranciscoAlcatraz Island`

0 1 . 4

The subroutine SUBSTRING **extracts characters from a given string.**

**For example,** SUBSTRING(3, 5, 'Computing') **would return** put

**The variable** y **is assigned a value using the statement:**

y ← SUBSTRING(4, 7, landmark)

**Using FIGURE 1, provided in the Diagram Booklet, what is the value of** y**?**

**Shade ONE lozenge. [1 mark]**

O **A** Alca

O **B** Atra

O **C** land

O **D** traz

**[Turn over]**

0 7

**FIGURE 1 is provided in the Diagram Booklet.**

| 0 | 1 | . | 5 |

The subroutine `POSITION` **finds the first position of a character in a string.**

**For example,** `POSITION('Computing', 'p')` **would return** `3`

**The variable** `z` **is assigned a value using the statement:**

`z ← POSITION(landmark, 't')`

**Using FIGURE 1, provided in the Diagram Booklet, what value is assigned to** `z`**?**

**Shade ONE lozenge.  [1 mark]**

○  **A** `-1`

○  **B** `3`

○  **C** `4`

○  **D** `5`
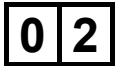
**BLANK  PAGE**

**[Turn over]**

**FIGURE 2, provided in the Diagram Booklet, shows an algorithm that uses integer division which has been represented using pseudo-code.**

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

**Integer division is the number of times one integer divides into another, with the remainder ignored.**

**For example:**

- `14 DIV 5` **evaluates to** `2`

- `25 DIV 3` **evaluates to** `8`

| 0 | 2 | . | 1 |
|---|---|---|---|

**Where is iteration FIRST used in the algorithm in FIGURE 2?**

**Shade ONE lozenge.  [1 mark]**

◯ **A  Line number 2**

◯ **B  Line number 4**

◯ **C  Line number 6**

◯ **D  Line number 11**

**[Turn over]**

| 0 | 2 | . | 2 |

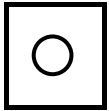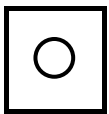**In the algorithm in FIGURE 2, provided in the Diagram Booklet, what will be output when the user input is** `10`**?**

**Shade ONE lozenge.  [1 mark]**

○  **A**  `0`

○  **B**  `1`
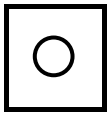
○  **C**  `2`
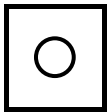
○  **D**  `4`

0 2 . 3

**In the algorithm in FIGURE 2, provided in the Diagram Booklet, what is the largest possible value of the variable** `counter` **when the user input is** `36`**?**

**Shade ONE lozenge.  [1 mark]**

○ **A**  $0$

○ **B**  $2$

○ **C**  $4$

○ **D**  $5$

**[Turn over]**

**0 3**

**Explain ONE advantage of the structured approach to programming. [2 marks]**

_____

_____

_____

_____

_____

5

| 0 | 4 |

**FIGURE 3, provided in the Diagram Booklet, shows a program written in Python that calculates the area of a rectangle or the volume of a box from the user inputs.**

| 0 | 4 | . | 1 |

**Complete the trace table using the program in FIGURE 3. [3 marks]**

| numOne | numTwo | numThree | FINAL OUTPUT |
|--------|--------|----------|--------------|
| 5 | 6 | −1 | |
| 10 | 4 | 0 | |
| 3 | 5 | 10 | |

**[Turn over]**

**0 4 . 2**

**Describe ONE way that the program in FIGURE 3, provided in the Diagram Booklet, could be made more robust.  [1 mark]**

_____

_____

_____

_____

0 5

FIGURE 4, provided in the Diagram Booklet, shows an algorithm presented as a flowchart.

Complete the trace table for the algorithm in FIGURE 4.

You may not need to use all the rows in the table.
[3 marks]

| a | b | c |
|---|---|---|
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |
|   |   |   |

**0 6**

**FIGURE 5, provided in the Diagram Booklet, shows an algorithm represented using pseudo-code.**

**The algorithm is for a simple authentication routine.**

**The pseudo-code uses a subroutine** `getPassword` **to check a username:**

- **If the username exists, the subroutine returns the password stored for that user.**

- **If the username does not exist, the subroutine returns an empty string.**

**Parts of the algorithm are missing and have been replaced with the labels ⬤L1 to ⬤L4 .**

**State the items from FIGURE 6, provided in the Diagram Booklet,  that should be written in place of the labels in the algorithm in FIGURE 5.**

**You will not need to use all the items in FIGURE 6.**
**[4 marks]**

**⬤L1**  _____

_____

**⬤L2**  _____

L3 _____

_____

L4 _____

_____

**[Turn over]**

10

**BLANK PAGE**

**[Turn over]**

**0 8**

**FIGURE 7, provided in the Diagram Booklet, shows a merge sort being carried out on a list.**

**Explain how the merge sort algorithm works.  [4 marks]**

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

**[Turn over]**

**0 9**

**FIGURE 8, provided in the Diagram Booklet, shows an algorithm, written using pseudo-code, that uses a** RECORD **data structure for storing information about a film.**

**Each record stores four pieces of information about a film:**

- **film title**

- **certificate (eg 12A, PG)**

- **year the film was made**

- **if the film is currently being shown at a cinema.**

**There are records for three films and these films are stored alphabetically in an array called** filmCollection**.**

**The pseudo-code outputs the title of the newest of the three films.**

- **Part of the algorithm has been replaced by the label L1.**

| 0 | 9 | . | 1 |

**How many different values can the field** `beingShown` **have?**

**Shade ONE lozenge.  [1 mark]**

| ○ | **A  2** |

| ○ | **B  3** |

| ○ | **C  128** |

| ○ | **D  256** |

**[Turn over]**

| 0 | 9 | . | 2 |

**Which assignment statement changes the year the film 'Hulk' was made to 2003?**

**Shade ONE lozenge.  [1 mark]**

O    **A** `hulk.year ← 2003`

O    **B** `filmCollection[0].year ← 2003`

O    **C** `Film(year) ← 2003`

O    **D** `hulk(year) ← 2003`

**0 9 . 3**

**What should the label 🅛🅝 in FIGURE 8, provided in the Diagram Booklet, be replaced by?**

**Shade ONE lozenge. [1 mark]**

○ **A** `3`

○ **B** `LEN(filmCollection)`

○ **C** `LEN(filmCollection) - 1`

○ **D** `Position`

**0 9 . 4**

**Write a pseudo-code statement that updates the `antMan` record to show that the film is currently being shown at the cinema. [1 mark]**

_____

_____

**[Turn over]**

**8**

**1 0**

FIGURE 9, provided in the Diagram Booklet, shows an algorithm, represented in pseudo-code, used to display students' test scores. The algorithm does not work as expected and the teacher wants to find the error.

The algorithm should display three test scores for each student:

- Natalie has results of 78, 81 and 72
- Alex has results of 27, 51 and 54
- Roshana has results of 52, 55 and 59.

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

**1 0 . 1**

Complete the trace table, on the opposite page, for the algorithm shown in FIGURE 9.

You may not need to use all the rows in the table.
[5 marks]

| count | i | person | j | result |
|---|---|---|---|---|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**[Turn over]**

**1 0 . 2**

**How could the error in the algorithm in FIGURE 9, provided in the Diagram Booklet, be corrected?**

**Shade ONE lozenge. [1 mark]**

A **Change line number 3 to:** `count ← -1`

B **Change line number 4 to:** `FOR i ← 1 TO 4`

C **Change line number 7 to:** `FOR j ← 0 TO 2`

D **Change line number 9 to:** `result ← scores[j * 3 + i]`

BLANK PAGE

**1 1** FIGURE 10, provided in the Diagram Booklet, shows part of an algorithm that has been written in pseudo-code.

There is an error in the algorithm.

The algorithm should:

- get the start year and end year from the user

- check that the start year is before the end year

- check that the start year is before 2000

- calculate the difference between the two years after a valid start year has been entered.

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

**1 1 . 1** TABLE 1, on the opposite page, shows three tests used to check the algorithm in FIGURE 10, provided in the Diagram Booklet.

**Complete the table to show what the values of the** validChoice **and** difference **variables would be for the given test data.  [4 marks]**

**TABLE 1**

| TEST TYPE | TEST DATA | | validChoice | difference |
|---|---|---|---|---|
| NORMAL | startYear | 1995 | | |
| | endYear | 2010 | | |
| ERRONEOUS | startYear | 2015 | | |
| | endYear | 2000 | | |
| BOUNDARY | startYear | 2000 | | |
| | endYear | 2023 | | |

[Turn over]

BLANK PAGE

**1 1 . 2**

**The algorithm in FIGURE 10, provided in the Diagram Booklet, contains a logic error on LINE 11.**

**Describe how the error on LINE 11 can be corrected.  [1 mark]**

_____

_____

| | 11 |
|---|---|

1 2 . 1

**FIGURE 11, provided in the Diagram Booklet, shows a binary search algorithm that has been programmed in Python.**

**Complete the trace table, on the opposite page, for the program in FIGURE 11 if the user input is `wolf`**

**Part of the table has already been filled in.**

**You may not need to use all the rows in the table. [4 marks]**

| animalToFind | validAnimal | start | finish | mid |
|---|---|---|---|---|
| wolf | False | 0 | 7 | 3 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**1 2 . 2**

**FIGURE 12 shows a line of Python code that creates a list of fruit names.**

**FIGURE 12**

```
fruits = ["banana", "apple", "orange", "pear",
          "grape", "pineapple"]
```

**Extend the program in FIGURE 12. Your answer must be written in Python.**

**The program should get the user to enter a word and perform a LINEAR search on the list `fruits` to find if the word is in the list or not.**

**The program should:**

- **ask the user what word they would like to find**
- **output the message `True` if the word is found**
- **output the message `False` if the word is not found.**

You must write your own linear search routine and NOT use any built-in search function available in Python.

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, on pages 42 and 43, contains vertical lines to help you indent your code. **[7 marks]**

4 1

```
fruits = ["banana", "apple", "orange", "pear",
          "grape", "pineapple"]
```

**12.3**

**State why a binary search cannot be used on the list** `fruits` **[1 mark]**

_____

_____

_____

☐1☐2 . ☐4

**FIGURE 13, provided in the Diagram Booklet, shows an algorithm, represented using pseudo-code, that should display currency names in reverse alphabetical order, starting with** `yen`**.**

**There are errors in the logic of the algorithm.**

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

**Rewrite LINE 1 and LINE 6 from FIGURE 13 to make the algorithm work as intended.  [3 marks]**

**Line 1** _____

_____

_____

**Line 6** _____

_____

_____

15

**13**

**A programmer is writing a game. The game uses a 3 x 3 grid containing nine squares.**

**In the game, a square on the grid is referred to by a letter and a number. For example, square C3 in FIGURE 14, provided in the Diagram Booklet, contains an X.**

**FIGURE 15, provided in the Diagram Booklet, shows part of a Python program that checks the grid reference entered by a player.**

**The grid reference is valid if:**

- **there are exactly two characters**

- **the first character entered is A, B or C**

- **the second character entered is 1, 2 or 3.**

The Python function `upper()` converts letters into uppercase, eg `b1` would be converted to `B1`

Extend the program from FIGURE 15, provided in the Diagram Booklet, so it completes the other checks needed to make sure a valid grid reference is entered.

Your extended program must:

- use the variable `check`

- repeat the following steps until a valid grid reference is entered:

  – get the user to enter a grid reference

  – output an appropriate message if the grid reference entered is not valid.

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, on pages 48 to 50, contains vertical lines to help you indent your code. [6 marks]

[Turn over]

```
check = False
while check == False:
    square = ""
    while len(square) != 2:
        square = input("Enter grid reference (eg C2): ")
        square = square.upper()
```

BLANK PAGE

[Turn over]

**1 4**

**50 students have voted for the music genre they like best.**

**FIGURE 16, provided in the Diagram Booklet, shows an INCOMPLETE algorithm, represented using pseudo-code, designed to output the highest or lowest results of the vote.**

**The programmer has used a two-dimensional array called** `results` **to store the genre and the number of votes for each genre.**

**Parts of the algorithm are missing and have been replaced with the labels** **L1** **to** **L3** .

**State what should be written in place of the labels** **L1** **to** **L3** **in the algorithm in FIGURE 16, provided in the Diagram Booklet.  [3 marks]**

**L1** _____

_____

**L2** _____

_____

**[Turn over]**

9

1 5

A group of people have a meal in a restaurant. Instead of one person paying for the whole meal, each person will pay for what they eat.

Write a Python program that asks each person in the group how much they are paying towards the meal and works out when the bill is fully paid. Each person can pay a different amount.

The program should:

- **get the user to enter the total amount of the bill**

- **get a person to enter how much they are paying towards the bill**

- **subtract the amount entered from the bill:**

  - **if the amount left to pay is more than 0, output how much is left to pay and repeat until the amount left to pay is 0 or less**

  - **if the amount left to pay is 0, then output the message** `Bill paid`

  - **if the amount left to pay is less than 0, then output the message** `Tip is` **and the difference between the amount left to pay and 0**

You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.

The answer grid, on pages 55 to 58, contains vertical lines to help you indent your code. [8 marks]

58

8

**BLANK PAGE**

**[Turn over]**

1 6

**Question 16 is about a dice game played against a computer.**

**The aim of the game is to get as close to a score of 21 as you can, without going over 21. If your score goes over 21 then you lose.**

**The player's score starts at 0.**

**For each turn:**

- **two dice (each numbered from 1 to 6) are rolled**

- **the total of the two dice rolls is added to the player's score**

- **the value of each dice and the player's new total is output**

- **if the current score is less than 21, the player is asked if they would like to roll the dice again: if the player says yes, they get another turn; otherwise, the game ends.**

**At the end of the game, the program should work as follows:**

- **if the final score is 21, output a message to say the player has won**

- **if the final score is greater than 21, output a message to say the player has lost**

- **if the final score is less than 21, the program generates a random number between 15 and 21 inclusive:**

  - **if this random number is greater than the player's final score, output a message to say the player has lost**

  - **otherwise, output a message to say the player has won.**

**FIGURE 17, provided in the Diagram Booklet, shows the output of a program that plays this dice game.**

**Write a Python program to simulate this game.**

**The first line has been written for you in the answer grid.**

**The dice rolls are carried out by the program generating random numbers between 1 and 6. You will need to use the Python function** `random.randrange(a, b)` **which generates a random integer in the range** `a` **to** `b` **starting at** `a` **but finishing one before** `b`**.**

**You SHOULD use indentation as appropriate, meaningful variable name(s) and Python syntax in your answer.**

**The answer grid, on pages 63 to 67, contains vertical lines to help you indent your code. [11 marks]**

**[Turn over]**

**BLANK PAGE**

```
import random
```

**END OF QUESTIONS**

**Additional page, if required.**

**Write the question numbers in the left-hand margin.**

**Additional page, if required.**

**Write the question numbers in the left-hand margin.**

**BLANK PAGE**

| For Examiner's Use | |
|---|---|
| Question | Mark |
| 1 | |
| 2–3 | |
| 4–5 | |
| 6–7 | |
| 8–9 | |
| 10–11 | |
| 12 | |
| 13–14 | |
| 15 | |
| 16 | |
| **TOTAL** | |