



A-level

COMPUTER SCIENCE

Paper 1

Wednesday 10 June 2026

Afternoon

Time allowed: 2 hours 30 minutes

Materials

For this paper you must have:

- a computer
- a printer
- appropriate software
- the Electronic Answer Document
- an electronic version and a hard copy of the Skeleton Program
- an electronic version and a hard copy of the Preliminary Material.

You must **not** use a calculator.

Instructions

- Type the information required on the front of your Electronic Answer Document.
- Before the start of the examination make sure your **Centre Number**, **Candidate Name** and **Candidate Number** are shown clearly **in the footer** of every page (also at the top of the front cover) of your Electronic Answer Document.
- Enter your answers into the Electronic Answer Document.
- Answer **all** questions.
- Save your work at regular intervals.

Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.
- No extra time is allowed for printing and collating.
- The question paper is divided into **four** sections.

Advice

You are advised to allocate time to each section as follows:

Section A – 40 minutes; **Section B** – 20 minutes; **Section C** – 20 minutes; **Section D** – 70 minutes.

At the end of the examination

Tie together all your printed Electronic Answer Document pages and hand them to the Invigilator.

Warning

It may not be possible to issue a result for this paper if your details are not on every page of your Electronic Answer Document.

Section A

You are advised to spend no longer than **40 minutes** on this section.

Enter your answers for **Section A** in your Electronic Answer Document.

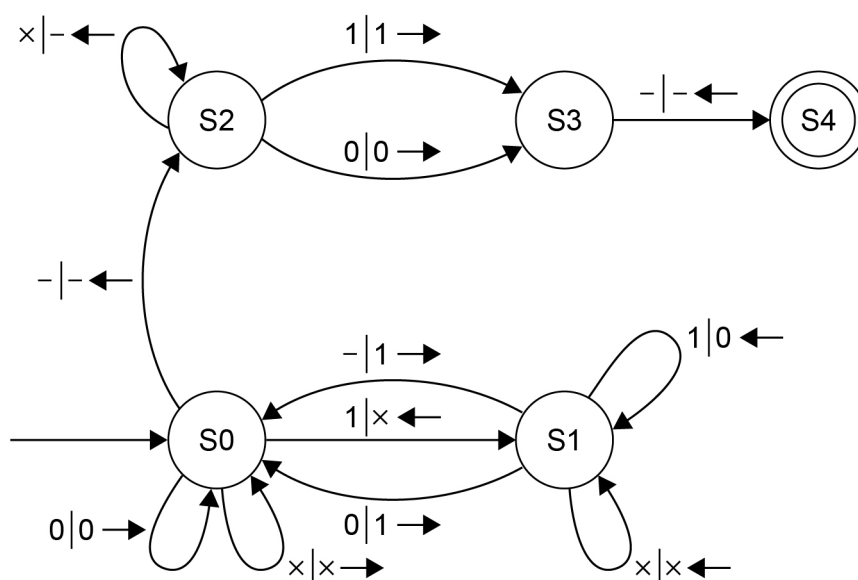
You **must save** this document at regular intervals.

0 1

Figure 1 shows the transition function (represented as a state transition diagram) and part of the tape for a Turing machine.

The current position of the read/write head is indicated by the * symbol.

Figure 1



Tape contents						
-	-	-	1*	1	1	-

Current state: S0

The label $1|x \leftarrow$ on the transition from S0 to S1 means if the machine is in state S0 and a 1 is read from the tape, then an x should be written to the tape and then the read/write head is moved one cell to the left.

0 1 . 1

What is the maximum size for the tape of a Turing machine?

[1 mark]

Complete the unshaded cells of **Table 1** to show the result of tracing the computation of this Turing machine for the first 23 steps only.

Show the contents of the tape, the current position of the read/write head and the current state as the input is processed.

The top row of the table shows the initial state, the initial position of the read/write head and the starting contents of the tape.

Table 1

[illegible]

Copy the contents of the unshaded cells in **Table 1** into the table in your Electronic Answer Document.

[6 marks]

Question 1 continues on the next page

Turn over ►

0 1 . 3 State the purpose of the Turing machine in **Figure 1**, on page 2. [1 mark]

0 1 . 4 Describe the importance of Turing machines to computation theory. [2 marks]

0 2 Circular and priority are types of queues that can be implemented as a static data structure using a fixed-length array.

0 2 . 1 Describe the steps involved in adding an item to a circular queue.
In your answer to this question you **must** describe how any pointers are used or changed. [5 marks]

0 2 . 2 Describe the steps involved in adding an item to a priority queue.
In your answer to this question you **do not** need to describe how any pointers are used or changed. [3 marks]

0 2 . 3 Describe **three** differences between dynamic and static data structures. [3 marks]

0 3 . 1 Explain what is wrong with the following description:
“Big-O is used to express how the time taken by an algorithm grows relative to the amount of memory it uses.” [1 mark]

Figure 2 shows three functions called f , g and h .

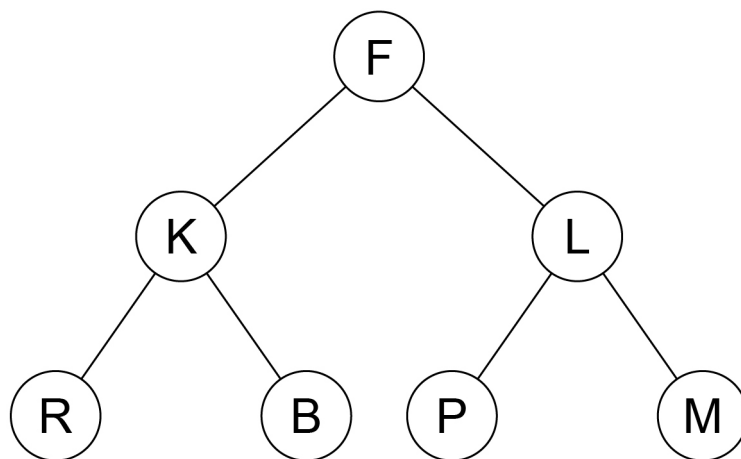
Figure 2

<pre> FUNCTION f(n) IF n=1 THEN RETURN 1 ELSE RETURN n*f(n-1) ENDIF ENDFUNCTION </pre>	<pre> FUNCTION g(n) s ← n+1 s ← s*2 s ← s+1 RETURN s ENDFUNCTION </pre>	<pre> FUNCTION h(n) s ← 0 FOR i ← 1 TO n FOR j ← 1 TO n s ← s+1 ENDFOR ENDFOR s ← s+1 RETURN s ENDFUNCTION </pre>
--	---	---

- 0 3 . 2** State the Big-O time complexity of the function f shown in **Figure 2**. [1 mark]
- 0 3 . 3** State the Big-O time complexity of the function g shown in **Figure 2**. [1 mark]
- 0 3 . 4** State the Big-O time complexity of the function h shown in **Figure 2**. [1 mark]
- 0 3 . 5** State the **maximum** number of comparisons needed for a binary tree search algorithm being used on a binary tree containing 31 nodes where every node has two child nodes unless it is a leaf node. [1 mark]
- 0 3 . 6** State the **maximum** number of comparisons needed for a binary tree search algorithm being used on **any** binary tree containing 31 nodes. [1 mark]

Figure 3 shows a binary tree.

Figure 3



- 0 3 . 7** State the order that the nodes in the binary tree in **Figure 3** will be visited with an in-order traversal. [2 marks]
- 0 3 . 8** State the order that the nodes in the binary tree in **Figure 3** will be visited with a pre-order traversal. [2 marks]
- 0 3 . 9** State the order that the nodes in the binary tree in **Figure 3** will be visited with a post-order traversal. [2 marks]

Turn over ►

0 4

Figure 4 shows two regular expressions.

The metacharacter | means or.

The expressions A | B and B | A are considered to be the same expression.

Figure 4

Expression 1

1+0|10?11

Expression 2

101*(01)?

0 4 . 1

Write a string that matches both the regular expressions in **Figure 4**.

[1 mark]

0 4 . 2

Complete the unshaded cells of **Table 2** to show which expressions match with the string 111 in addition to all the strings that 101*(01)? already matches.

It is acceptable for an expression to also match other additional strings as well as 111

Write Y or N in each row of **Table 2**.

Table 2

Expression	Y/N
101*(01)? 111	
10?1*(01)?	
101*(01)?1+	
10*1*(01)?	
101(11)*(01)?	
(111)(10?)101*(01)?	

Copy the contents of the unshaded cells in **Table 2** into the table in your Electronic Answer Document.

[3 marks]

Section B

You are advised to spend no more than **20 minutes** on this section.

Enter your answers to **Section B** in your Electronic Answer Document.

You **must save** this document at regular intervals.

The question in this section asks you to write program code
starting from a new program/project/file.

You are advised to **save** your program at regular intervals.

0 5

Write a program that gets the user to enter a limiting value, n . It should keep doing this until the user enters a value that is greater than 1

The program should then output all the numbers between 2 and n that form a square number when the number is added to the sum of its digits.

Example

If the limiting value, n , entered by the user is 17, then:

- 2 will be output by the program as $2 + 2 = 4$, and 4 is a square number
- 8 will be output by the program as $8 + 8 = 16$, and 16 is a square number
- 17 will be output by the program as $17 + 1 + 7 = 25$, and 25 is a square number.

A square number is one that can be formed by multiplying a positive integer by itself. For example, 9 is a square number because it is equal to 3×3

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

0 5 . 1

Your PROGRAM SOURCE CODE.

[12 marks]

0 5 . 2

SCREEN CAPTURE(S) showing the results of testing the program by entering the value 1 and then the value 245

[1 mark]

Turn over for the next section

Turn over ►

Section C

You are advised to spend no more than **20 minutes** on this section.

Enter your answers to **Section C** in your Electronic Answer Document.

You **must save** this document at regular intervals.

These questions refer to the **Preliminary Material** and the **Skeleton Program**, but **do not** require any additional programming.

Refer **either** to the **Preliminary Material** issued with this question paper **or** your electronic copy.

0 6

Describe **three** circumstances, other than when picking up food or adding food to its nest, when an ant will not move to the neighbouring cell containing the strongest pheromone belonging to an ant from the same nest as itself when the simulation advances.

[3 marks]

0 7

This question is about the `GetIndicesOfNeighbours` subroutine in the `Simulation` class.

0 7 . 1

Describe what adding a value of -1 to the `ListOfNeighbours` list indicates.

[1 mark]

0 7 . 2

Explain why the 5th element of the list returned will always be -1

[1 mark]

0 8

This question is about the `AdvanceStage` subroutine in the `Nest` class.

0 8 . 1

This subroutine uses overriding.

What is overriding?

[1 mark]

0 8 . 2

State the names of **two** properties/attributes used in this subroutine that are inherited.

[1 mark]

0 8 . 3

Describe the difference between private and protected specifiers.

[1 mark]

0 8 . 4

State the final value that `AntsToCull` will have if there is no food in the nest and there are 10 ants that belong to the nest.

[1 mark]

0 8 . 5

How many of the following five statements about the `AdvanceStage` subroutine are true?

- It is more likely that an ant created by the subroutine will be a queen ant than a worker ant.
- It is possible that all the ants created by the subroutine could be worker ants.
- It is possible that all the ants created by the subroutine could be queen ants.
- Some new ants will be created every time the subroutine is called.
- The total number of ants created by the subroutine will always be less than or equal to the number of queen ants that are in the nest when the subroutine is called.

[1 mark]

0 9

The Skeleton Program uses global variables and local variables.

Describe **two** differences between global variables and local variables.

[2 marks]

1 0

Two different types of relationship between objects in object-oriented programming are composition and aggregation.

Describe the difference between these two types of relationship.

[1 mark]

Turn over for the next section

Turn over ►

There are no questions printed on this page

Section D

You are advised to spend no more than **70 minutes** on this section.

Enter your answers to **Section D** in your Electronic Answer Document.

You **must save** this document at regular intervals.

These questions require you to load the **Skeleton Program** and to make programming changes to it.

1 1

This question refers to the `DisplayMenu` and `Main` subroutines.

The Skeleton Program is to be changed so that the user can choose to add food to any cell in the grid.

Task 1

Modify the `DisplayMenu` subroutine so that the new option is displayed.

Task 2

Modify the `Main` subroutine so that if the new option is chosen the user is asked to enter the row and column of the cell to use and then 500 is added to the amount of food currently in that cell.

Task 3

Test that the changes you have made work:

- run the Skeleton Program
- enter 1
- enter 1
- choose the new menu option to add food
- enter a row and column for a cell that is currently empty
- choose the new menu option to add food again
- enter the row and column for the same cell that was previously chosen
- enter 3
- enter the row and column for the same cell that was previously chosen.

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 1 . 1

Your PROGRAM SOURCE CODE for the amended subroutines `DisplayMenu` and `Main`

[4 marks]

1 1 . 2

SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

Turn over ►

When the user chooses option 2 from the menu (Display area details) the Skeleton Program should display a grid showing information about that area, instead of the current list about each cell.

- If there is a nest in a cell then an **N** should be shown.
- If there are ants in a cell then
 - an **A** should be shown if there is one ant
 - a **B** should be shown if there are two ants
 - a **C** should be shown if there are more than two ants.
- If there is at least one pheromone in a cell then a **P** should be shown.
- If there is food in a cell then an **F** should be shown.

Each cell should take up the same amount of space when displayed on the screen regardless of what the cell contains.

A straight line should be shown before each cell and at the end of each row.

Each row should be displayed on a new line.

Figure 5 shows what the display should look like if the user chooses to look at the area between row 1, column 1 and row 5, column 5 when there is:

- food at row 2 column 1
- food at row 2 column 2
- a nest at row 2 column 4
- six ants at row 2 column 4
- food at row 4 column 4

Figure 5

	F		F			NC		
							F	

What you need to do

Task 1

Modify the `GetAreaDetails` subroutine in the `Simulation` class so that the user-selected area is displayed as a grid.

You will need to add new code to the subroutine, alter existing code in the subroutine and delete existing code from the subroutine.

Task 2

Test that the changes you have made work:

- run the Skeleton Program
- enter 1
- enter 2
- enter 1
- enter 1
- enter 5
- enter 5

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 2 . 1 Your PROGRAM SOURCE CODE for the amended subroutine `GetAreaDetails` **[7 marks]**

1 2 . 2 SCREEN CAPTURE(S) showing the requested test. **[1 mark]**

Turn over for the next question

Turn over ►

1 3

This question extends the Skeleton Program by adding an obstacle that prevents ants from entering certain cells.

What you need to do

Task 1

Modify the `Cell` class so:

- there is a private attribute/property called `HasObstacle`
- `HasObstacle` is set to `False` in the `Cell` constructor
- there is a subroutine called `ContainsAnObstacle` that will return the value of `HasObstacle`
- there is a subroutine called `AddObstacle` that adds an obstacle to the cell by changing the value of `HasObstacle`

Task 2

Modify the `Simulation` class so:

- there is a subroutine called `AddObstaclesToColumn` that takes one integer parameter and changes each of the cells in the column indicated by the parameter so that they contain obstacles
- the `GetIndicesOfNeighbours` subroutine is modified so that an ant will not be able to move into a cell if it contains an obstacle.

Task 3

Modify the `Main` subroutine so that when simulation number 2 is chosen the cells in column 3 will contain obstacles.

Task 4

Test that the changes you have made work:

- run the Skeleton Program
- enter 2
- enter 5
- enter 5
- enter 1

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1 3 . 1

Your PROGRAM SOURCE CODE for the amended class `Cell`, the amended subroutine `GetIndicesOfNeighbours`, the new subroutine `AddObstaclesToColumn` and the amended subroutine `Main`

[11 marks]

1 3 . 2

SCREEN CAPTURE(S) showing the requested test.

[1 mark]

1	4
---	---

This question extends the Skeleton Program so that there is a new type of ant, a soldier ant.

A soldier ant will move one cell at a time in a random direction but will never move more than two cells away from its home nest.

When the simulation advances, if a soldier ant is in the same cell as any ants from another nest it will attack them. When an ant is attacked there is a 90% chance that it will be destroyed.

What you need to do

Task 1

Create a new class called `SoldierAnt` that inherits from the `Ant` class.

The constructor should set the type of ant to be `soldier`

The `ChooseCellToMoveTo` subroutine should be overridden so that a soldier ant moves in the way described above.

- Your code should call the subroutine `ChooseRandomNeighbour` to select a cell and keep doing this until the selected cell is within two cells of the soldier ant's home nest.
- The subroutine `ChangeCell` can be used to move the soldier ant.

Task 2

Modify the `SetUpANestAt` subroutine in the `Simulation` class so that, if the simulation starts with more than one nest, two soldier ants are created in **each** nest.

Task 3

Modify the `AdvanceStage` subroutine in the `Simulation` class so that soldier ants destroy ants from other nests in the way described above.

Task 4

Test that the changes you have made can create soldier ants correctly:

- run the Skeleton Program
- enter 4
- enter 3
- enter 2
- enter 4

Evidence that you need to provide

Include the following evidence in your Electronic Answer Document.

1	4	.	1
---	---	---	---

Your PROGRAM SOURCE CODE for the new class `SoldierAnt`, the amended subroutine `AdvanceStage` and the amended subroutine `SetUpANestAt`

[11 marks]

1	4	.	2
---	---	---	---

SCREEN CAPTURE(S) showing the requested test.

[1 mark]

END OF QUESTIONS

There are no questions printed on this page

Copyright information

For confidentiality purposes, all acknowledgements of third-party copyright material are published in a separate booklet. This booklet is published after each live examination series and is available for free download from www.aqa.org.uk

Permission to reproduce all copyright material has been applied for. In some cases, efforts to contact copyright-holders may have been unsuccessful and AQA will be happy to rectify any omissions of acknowledgements. If you have any queries please contact the Copyright Team.

Copyright © 2026 AQA and its licensors. All rights reserved.

