OXFORD AQA

INTERNATIONAL QUALIFICATIONS

INTERNATIONAL GCSE
COMPUTER SCIENCE

Paper 1 Programming

Friday 15 May 2026 | 08:00 UK Time | Time allowed: 2 hours

Materials

For this paper you must have access to:

- a computer
- a printer
- appropriate software
- electronic versions of the **Preliminary Material** and the **Skeleton Program**
- a hard copy of the **Preliminary Material** and the **Skeleton Program**.

Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 80.
- No extra time is allowed for printing and collating.
- The question paper is divided into **three** sections.
- You may use a bilingual dictionary.
- You must **not** use an English dictionary.
- You are **not** allowed to use a calculator.

Instructions

- Type the information required on the front of your **Electronic Answer Document**.
- Answer **all** questions.
- Enter your answers into the **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 of your **Electronic Answer Document** (not the front cover).
- Include the evidence required for your answers to **Sections B** and **C** in your **Electronic Answer Document**.
- You **must save** your **Electronic Answer Document** at regular intervals.
- The questions in **Sections B** and **C** require you to make changes to the **Skeleton Program**.
- All of the programming questions in **Sections B** and **C** can be answered independently of each other. If you cannot answer one of the questions you can still attempt to solve later questions.
- You are advised to keep a backup copy of the original **Skeleton Program** so that you can go back to it if you accidentally make changes to the program which means it can no longer be compiled/executed while answering the questions in **Sections B** and **C**.

Secure all your printed Electronic Answer Document pages together and hand them to the invigilator.

Section A (Non-programming questions)

You are advised to spend no more than **30 minutes** on this section.

Type your answers to **Section A** in your **Electronic Answer Document**.

You **must save** your **Electronic Answer Document** at regular intervals.

The questions in this section are about programming and how the **Skeleton Program** works. Do **not** make any changes to the **Skeleton Program** when answering these questions.

0 1 . 1 State the name of a variable in the **Skeleton Program** that can only hold one of two values.

[1 mark]

0 1 . 2 State the name of a variable in the subroutine `GenerateRandomEvent` that is used to hold a value that is **not** a whole number.

[1 mark]

0 1 . 3 State the name of the subroutine in the **Skeleton Program** that uses the remainder operation.

[1 mark]

0 1 . 4 The subroutine `PlayGame` uses two `WHILE` loops.

Explain why the second loop is a `WHILE` loop instead of a `FOR` loop.

[1 mark]

0 1 . 5 How many parameters does the subroutine `GetBuildingByInitial` have?

[1 mark]

0 1 . 6 Describe the data structure used for the variable `Buildings` in the subroutine `SetupBuildings`

[2 marks]

0 1 . 7 State **two** advantages of subroutines that use parameters.

[2 marks]

0 1 . 8 Describe the purpose of the selection structure in the subroutine `CalculateCityIncome`

[2 marks]

0 1 . 9 Describe the conditions that must be true for the subroutine `GenerateRandomEvent` to display the text `People are getting happier`

[2 marks]

0 2 . 1 Explain why the subroutine `AddBuildingToCity` has more than one `RETURN` statement.

[2 marks]

0 2 . 2 State **two** string handling operations used in the subroutine `AddBuildingToCity`

[2 marks]

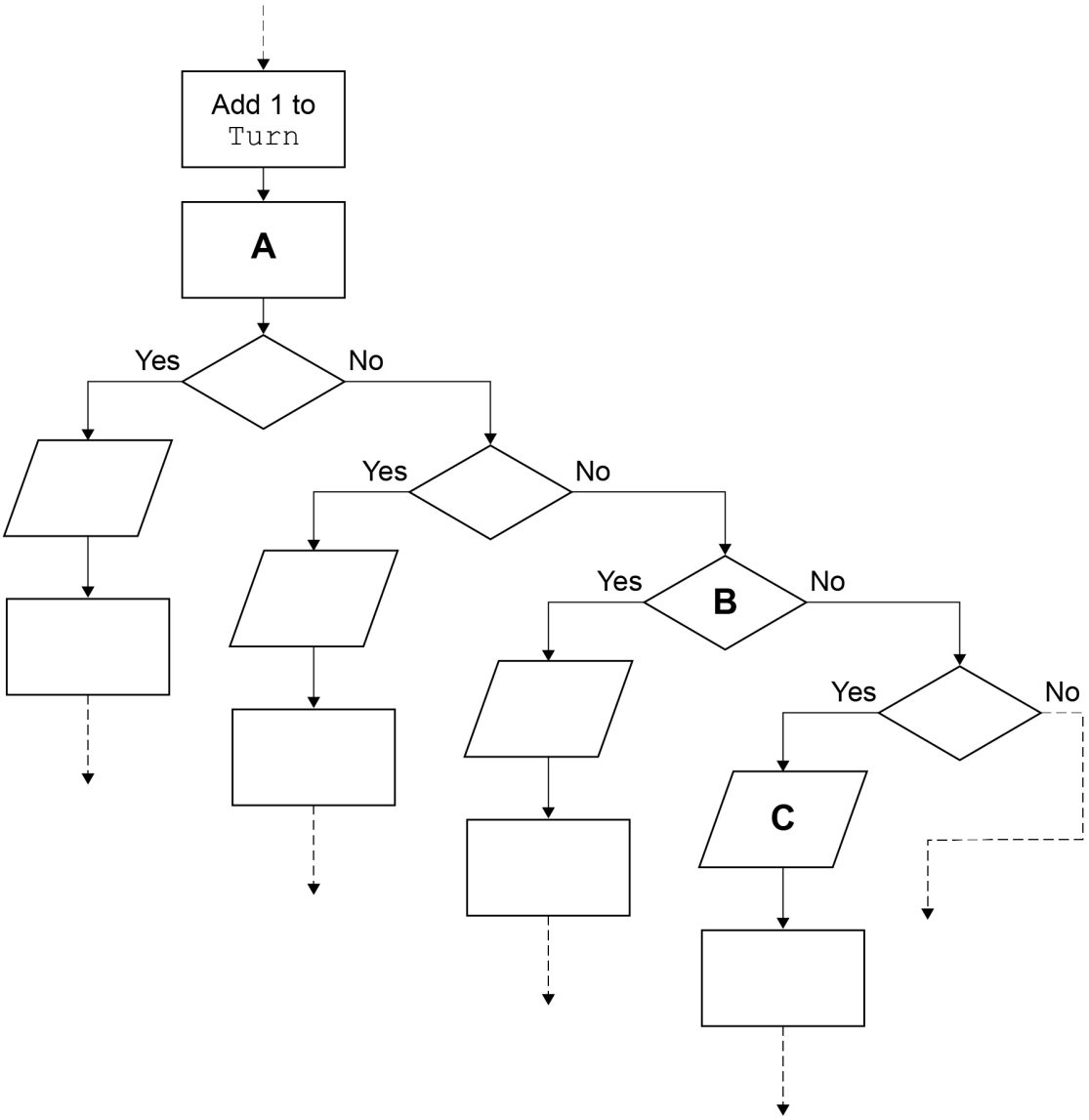
0 2 . 3 Describe how the variable `Coordinates` is used to determine the values of the variables `Row` and `Column` in the subroutine `AddBuildingToCity`

[2 marks]

Turn over for the next question

0 3 **Figure 1** contains an incomplete flowchart that shows how **part** of the subroutine `PlayGame` works. Some of the flowchart symbols have been labelled with the letters **A–C**.

Figure 1



Complete **Table 1** by writing each of the letters **A–C** next to the text that should be written in place of that letter on the flowchart.

You should enter each letter only once and you should enter no more than one letter on each row. You will only need to use three rows.

Table 1

Text	Letter (A–C)
Output “Pollution too high”	
`Money` is less than zero	
Output “Out of money”	
`Turn` is greater than `TURNS`	
Increase `Money` by city income	
Output “All turns completed”	
Output “Population too unhappy”	
`Pollution` is greater than 20	
Set `GameOver` to TRUE	
`Happiness` is less than zero	
Output “Final Statistics”	

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

[3 marks]

Turn over for the next section

Section B (Short programming questions)

You are advised to spend no more than **45 minutes** on this section.

0 4 The **Skeleton Program** is to be improved so that it displays a message when a game has ended.

Change the subroutine `Main` so that the message `Thank you for playing` is displayed after a game has been played and before the player is asked if they want to play again.

The subroutine should also display a **blank line** after the new message.

Test your changes by running the **Skeleton Program** and then:

- enter `E`
- enter `3`
- enter `E` five times.

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 4 . 1 **All** the PROGRAM SOURCE CODE for the subroutine `Main`

[3 marks]

0 4 . 2 SCREEN CAPTURE(S) showing the results of the requested test. You are only required to provide screen captures to show the output from the final entry of `E` to the end of the output.

[1 mark]

0 5 The **Skeleton Program** is to be extended so that after the city is displayed, the number of empty locations is also displayed.

Modify the subroutine `DisplayCity` so that **after** the city is displayed the following text is displayed:

```
N locations are empty
```

where `N` is the number of locations that do **not** contain a building.

If more than half of the city is empty the following text should **also** be displayed:

```
More buildings are needed
```

Your code should work correctly for a city of any size.

Test your changes by running the **Skeleton Program** and then:

- enter `E`
- enter `3`
- enter `C`
- enter `H`
- enter `1 2`
- enter `C`
- enter `F`
- enter `2 0`
- enter `E`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 5 . 1 **All** the PROGRAM SOURCE CODE for the subroutine `DisplayCity`

[6 marks]

0 5 . 2 SCREEN CAPTURE(S) showing the results of the requested test. You are only required to provide screen captures to show the output from the final entry of `E` to the end of the output.

[1 mark]

Turn over for the next question

0 6 The **Skeleton Program** is to be extended so that each column of the city has a column number displayed above it as in **Figure 2**.

Figure 2

```
0 1 2
- - -
- - -
- - -
```

Change the subroutine `DisplayCity` so that column numbers are displayed above each column. You should start numbering with 0

Your code must use iteration to work for a city of any size up to 10

Test your changes by running the **Skeleton Program** and then:

- enter `E`
- enter `5`
- stop the program and run it again
- enter `E`
- enter `3`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 6 . 1 **All** the PROGRAM SOURCE CODE for the subroutine `DisplayCity`

[3 marks]

0 6 . 2 SCREEN CAPTURE(S) showing the results of the requested tests.

[2 marks]

0 7 The **Skeleton Program** is to be extended so that the energy use of a building is stored in the same way as the other fields such as income and pollution.

Task 1

Modify the `Building` record so that each building has a field called `Energy` which has its value set when a new record is created.

Task 2

Modify the subroutine `SetupBuildings` so that the correct energy value for each building is stored.

- A house uses 1 unit of energy.
- A park uses 0 units of energy.
- A factory uses 10 units of energy.
- A school uses 4 units of energy.

Task 3

Modify the subroutine `AddBuildingToCity` so that, **after** the call to `GetBuildingByInitial` is made, the following message is displayed:

```
X uses Y energy per day
```

where `X` is the name of the building and `Y` is the amount of energy that building uses.

Test your changes by running the **Skeleton Program** and then:

- enter `E`
- enter `3`
- enter `C`
- enter `H`
- enter `1 0`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 7 . 1 **All** the PROGRAM SOURCE CODE for the `Building` record.

[2 marks]

0 7 . 2 **All** the PROGRAM SOURCE CODE for the subroutines `SetupBuildings` and `AddBuildingToCity`

[3 marks]

0 7 . 3 SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

0 8 The **Skeleton Program** is to be improved so that the cost of each building is displayed when a player is choosing a building to construct.

Modify the subroutine `AddBuildingToCity` so that the cost of each type of building is displayed.

If the cost value is greater than zero then this should be displayed as a cost, otherwise it should be displayed as a benefit.

For the existing data in the `Buildings` data structure, the display should look like this:

```
(H)ouse cost: 10
(P)ark cost: 5
(F)actory benefit: 10
(S)chool cost: 20
```

Test your changes by running the **Skeleton Program** and then:

- enter `E`
- enter `3`
- enter `C`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 8 . 1 **All** the PROGRAM SOURCE CODE for the subroutine `AddBuildingToCity`

[5 marks]

0 8 . 2 SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

Turn over for the next section

Section C (Longer programming questions)

You are advised to spend no more than **45 minutes** on this section.

0 9 The **Skeleton Program** is to be extended so that the player can resize the city during the game.

Task 1

Create a new subroutine called `ResizeCityGrid` that:

- takes `TheCity` and `NewSize` as parameters
- creates a new empty grid of size `NewSize` by calling the subroutine `CreateEmptyGrid`
- copies the contents of the locations from the old grid into the same locations in the new grid and only copies locations that fit within the new size, eg:

Example A

City of size 4

```
- H - P
- - F -
S - - H
- H H -
```

→

City resized to size 3

```
- H -
- - F
S - -
```

Example B

City of size 3

```
- H -
- - F
S - -
```

→

City resized to size 4

```
- H - -
- - F -
S - - -
- - - -
```

- updates `TheCity.Size` and `TheCity.Grid` to use the new size and the new grid
- displays the new city.

Task 2

Modify the subroutine `GetAction` so the player is prompted to enter:

- `C` to construct a building
- `R` to resize the city
- `E` to end the turn.

Task 3

Modify the subroutine `PlayGame` so that:

- if the player enters `R` they are asked to enter a new city size
- if the entered value is greater than 0 the new subroutine `ResizeCityGrid` is called
- otherwise the message `Size must be greater than zero` is displayed.

Task 4

Test your changes by running the **Skeleton Program** and then:

- enter `P`
- enter `A`
- enter `R`
- enter `4`
- enter `R`
- enter `5`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

0 9 . 1 **All** the PROGRAM SOURCE CODE for the new subroutine `ResizeCityGrid`

[9 marks]

0 9 . 2 **All** the PROGRAM SOURCE CODE for the subroutine `GetAction`

[1 mark]

0 9 . 3 **All** the PROGRAM SOURCE CODE for the subroutine `PlayGame`

[4 marks]

0 9 . 4 SCREEN CAPTURE(S) showing the results of the requested test.

[1 mark]

Turn over for the next question

1 0 The **Skeleton Program** is to be improved. When a new building has been constructed, the program will display the names of all buildings next to the new building.

Task 1

Create a new subroutine called `DisplaySurroundingBuildings` that takes `Row`, `Column` and `TheCity` as parameters. The new subroutine should be called from the subroutine `AddBuildingToCity` after a building is successfully constructed.

The subroutine should display the names of all of the buildings in the locations surrounding the location specified by `Row` and `Column`.

If a type of building appears more than once in a surrounding location then its name should be displayed more than once.

Your solution should work for a city of any size.

Below are three example cities. The bold **S** represents a newly constructed school. The surrounding locations are shaded in grey. The output is what should be displayed by your new subroutine after the school is constructed.

Example A

City

–	–	–
–	**S**	H
–	–	–

Output `Constructed next to House`

Example B

City

H	–	F	–	–	F
–	–	–	–	–	–
P	–	**S**	H	F	–
–	S	S	P	S	–
–	–	F	–	–	P
H	–	–	–	F	–

Output `Constructed next to House School School Park`

Example C

City

P	P	-
S	-	H
-	F	-

Output `Constructed next to Park Park Factory`

Task 2

Test your changes by running the **Skeleton Program** and then:

- enter `P`
- enter `B`
- enter `C`
- enter `P`
- enter `2 2`
- enter `C`
- enter `S`
- enter `1 1`

Evidence that you need to provide
Include the following evidence in your Electronic Answer Document.

1 0 . 1 **All** the PROGRAM SOURCE CODE for the new subroutine `DisplaySurroundingBuildings`

[13 marks]

1 0 . 2 **All** the PROGRAM SOURCE CODE for the subroutine `AddBuildingToCity`

[1 mark]

1 0 . 3 SCREEN CAPTURE(S) showing the results of 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.oxfordaqa.com

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

Copyright © 2026 OxfordAQA International Examinations and its licensors. All rights reserved.