

Modified Enlarged 18pt

OXFORD CAMBRIDGE AND RSA EXAMINATIONS

Thursday 25 May 2023 – Afternoon

GCSE (9–1) Computer Science

J277/02 Computational thinking, algorithms and programming

**Time allowed: 1 hour 30 minutes
plus your additional time allowance**

**DO NOT USE:
a calculator**

Please write clearly in black ink.

Centre number

Candidate number

First name(s) _____

Last name _____

READ INSTRUCTIONS OVERLEAF



INSTRUCTIONS

Use black ink.

Write your answer to each question in the space provided. If you need extra space use the lined pages at the end of this booklet. The question numbers must be clearly shown.

Answer ALL the questions.

INFORMATION

The total mark for this paper is 80.

The marks for each question are shown in brackets [].

ADVICE

Read each question carefully before you start your answer.

BLANK PAGE

SECTION A

- 1 (a) The table contains four statements about programming languages.

Tick (✓) ONE box in each row to identify whether each statement describes a low-level programming language or a high-level programming language. [4]

| Statement | Low-level | High-level |
|--|-----------|------------|
| The same language can be used on computers that use different hardware | | |
| It allows the user to directly manipulate memory | | |
| It allows the user to write English-like words | | |
| It always needs to be translated into object code or machine code | | |

- (b) The variables `num1` and `num2` store integers.

Write pseudocode to add the integers stored in `num1` and `num2`. Store the result in a variable with the identifier `total`

[1]

- (c) Three incomplete pseudocode algorithms are given with a description of the purpose of each algorithm.

Write the missing arithmetic operator for each algorithm.

- (i) Outputting 12 to the power of 2. [1]

```
print(12 ..... 2)
```

- (ii) Working out if a number is odd or even. [1]

```
number = 53
if number ..... 2 == 0 then
    print("Even number")
else
    print("Odd number")
endif
```

- (iii) Finding the difference between two measurements. [1]

```
measurement1 = 300
```

```
measurement2 = 100
```

```
difference = measurement1 ..... measurement2
```

(d) Read the following pseudocode algorithm:

```
01  start = 3
02  do
03    print(start)
04    start = start - 1
05  until start == -1
06  print("Finished")
```


2 This pseudocode algorithm totals all the numbers in the 0-indexed array scores

```
01 total = 0
02 for scoreCount = 1 to scores.length - 1
03     scores[scoreCount] = total + total
04 next scoreCount
05 print(total)
```

The function `length` returns the number of elements in the array.

The algorithm contains several errors.

Two types of errors in a program are syntax and logic errors.

(a) State what is meant by a syntax error and a logic error.

Syntax error _____

Logic error _____

[2]

(b) Identify TWO logic errors in the pseudocode algorithm.

Write the refined line to correct each error.

Error 1 line number _____

Corrected line _____

Error 2 line number _____

Corrected line _____

[4]

3 An insertion sort is one type of sorting algorithm.

A student has written the pseudocode algorithm opposite to perform an insertion sort on a 1D array names.

(a) Describe the purpose of the variable `temp` in the insertion sort pseudocode algorithm.

[2]

(b) An insertion sort contains a nested loop; a loop within a loop. In this pseudocode algorithm the outer loop is a count-controlled loop and the inner loop is a condition-controlled loop.

Explain why the inner loop needs to be a condition-controlled loop.

[2]

```
names = ["Kareem", "Sarah", "Zac", "Sundip", "Anika"]
for count = 1 to names.length - 1
    pos = count
    while (pos > 0 and names[pos] < names[pos - 1])
        temp = names[pos]
        names[pos] = names[pos - 1]
        names[pos - 1] = temp
        pos = pos - 1
    endwhile
next count
```

(c) A bubble sort is another type of sorting algorithm.

(i) Describe ONE difference between an insertion sort and a bubble sort.

[2]

(ii) Describe TWO similarities between an insertion sort and a bubble sort.

1

2

[2]

BLANK PAGE

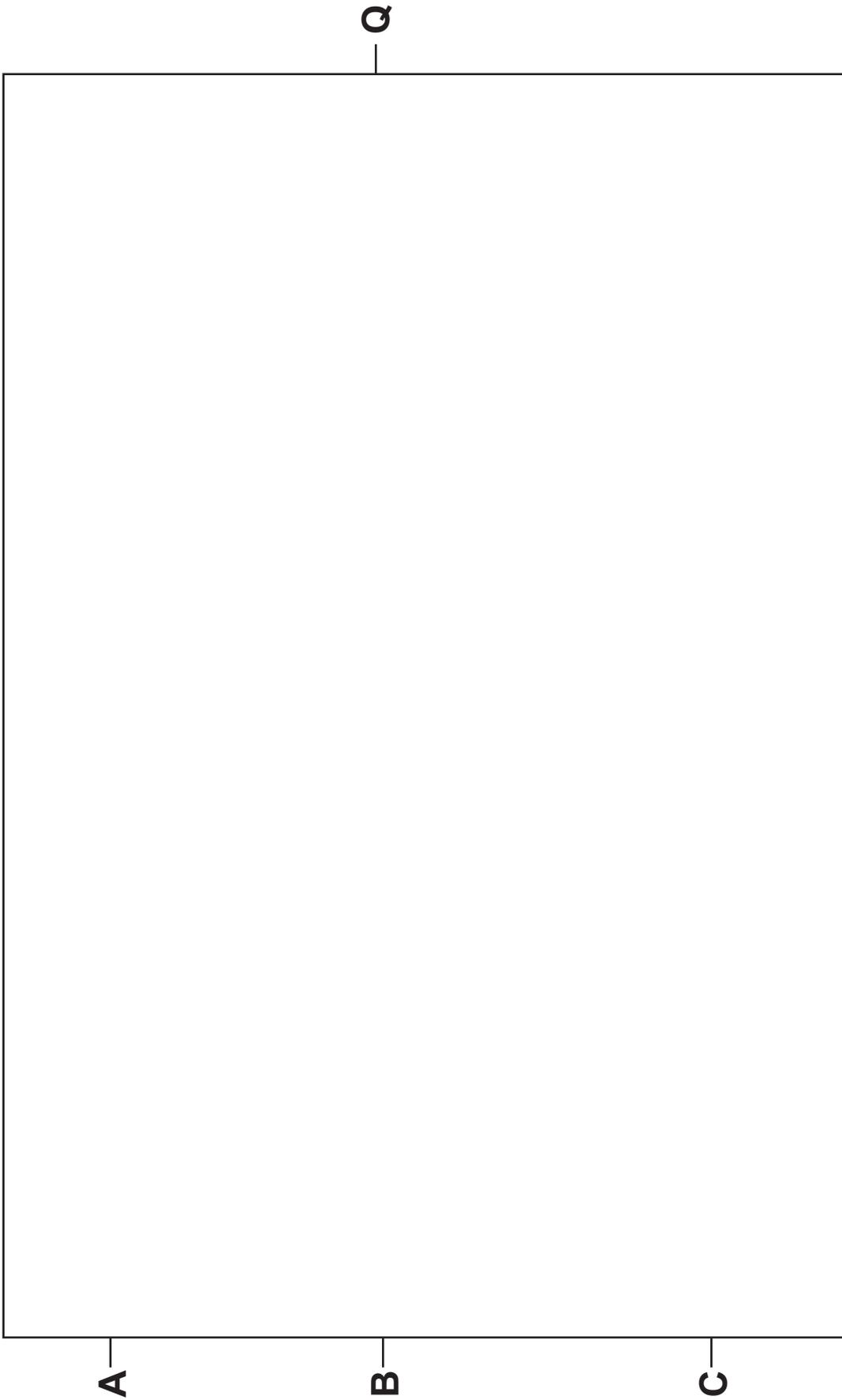
4 A garden floodlight system uses inputs from sensors and switches to decide whether it should be turned on.

The table shows the inputs into the system and the meaning of each input value:

| Letter | Input device | Input of 1 | Input of 0 |
|---------------|----------------------|--|--|
| A | Motion sensor | Motion is detected | Motion is not detected |
| B | Light sensor | Light levels indicate it is daytime | Light levels indicate it is nighttime |
| C | Light switch | The switch is turned on | The switch is turned off |

The floodlight (Q) is designed to be on ($Q = 1$) when the switch is turned on and the motion sensor detects motion at nighttime.

(a) Draw a logic diagram for the floodlight on the opposite page. [3]



(b) Identify the logic gates for truth table 1 and truth table 2. [2]

Truth table 1:

| A | B | Output |
|---|---|--------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Logic gate 1: _____

Truth table 2:

| A | B | Output |
|---|---|--------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Logic gate 2: _____

5 Charlie is developing an adding game. The rules of the game are:

The player is asked 3 addition questions.

Each question asks the player to add together two random whole numbers between 1 and 10 inclusive. If the player gets the correct answer, 1 is added to their score.

At the end of the game their score is displayed.

(a) Charlie has been told that the game will need to be tested before giving it to the players.

(i) Explain why programs should be tested before use.

[2]

(ii) Complete the table by naming and describing ONE type of test that should be used on Charlie's program before releasing it. [2]

| Test type | Description |
|------------------|--------------------|
| | |

- (iii) Complete the table by identifying AND describing TWO features of an IDE that can be used when testing a program. [4]

| Feature | Description |
|---------|-------------|
| | |
| | |

(b) Validating inputs can reduce errors when a program is being run.

Identify TWO methods of validation AND explain how they can be used on this game.

Validation method 1 _____

Use _____

Validation method 2 _____

Use _____

[6]

SECTION B

We advise you to spend at least 40 minutes plus your additional time allowance on this section.

Some questions require you to respond using either the OCR Exam Reference Language or a high-level programming language you have studied. These are clearly shown.

6 OCR Security Services is a company that installs intruder alarm systems in commercial buildings.

The systems use a computer that is connected to the door sensors and window sensors.

The following data is stored in the system:

| Data stored | Variable identifier | Example data |
|--|-----------------------------|----------------------|
| The user's name | UserName | Admin123 |
| A telephone number to call when the alarm is activated | EmergencyPhoneNumber | +449999999999 |
| Whether a door sensor is activated | DoorSensorActive | True |
| Whether a window sensor is activated | WindowSensorActive | True |
| A timer that counts, to the nearest second, how long a door sensor has been activated | DoorActiveTime | 100 |
| A timer that counts, to the nearest second, how long a window sensor has been activated | WindowActiveTime | 100 |
| Whether the system is armed | SystemArmed | True |
| Whether the system is in test mode | TestModeActive | True |

(a) Opposite is a table showing some variables within the program. [4]

Tick (✓) ONE box in each row to identify the most appropriate data type for each variable.

| Variable | Boolean | Char | String | Integer | Real |
|-----------------------------|----------------|-------------|---------------|----------------|-------------|
| UserName | | | | | |
| EmergencyPhoneNumber | | | | | |
| DoorSensorActive | | | | | |
| DoorActiveTime | | | | | |

- (b) The alarm has an algorithm that decides whether to sound the alarm by checking the data that is stored in the following three variables.

`SystemArmed`
`DoorSensorActive`
`WindowSensorActive`

The alarm will only sound when the alarm has been activated **AND** one or both of the door and window sensors are activated. When the system needs to sound the alarm it calls the pre-written procedure `SoundAlarm()`

Write a program that checks the data in the variables and calls `SoundAlarm()` when appropriate.

You must use **EITHER**:

- OCR Exam Reference Language, OR
- A high-level programming language that you have studied.

[4]

BLANK PAGE

- (c) The alarm system can also have motion sensors. Each type of sensor has a code. The code for each sensor is given in the table:

| Code | Sensor |
|------|---------------|
| MS | Motion sensor |
| DS | Door sensor |
| WS | Window sensor |

A program is written to reset the sensors. The program:

Asks the user to enter the code for the sensor they want to reset.

Calls the prewritten function `CheckSensorCode()` to check whether the code entered is a valid code. The sensor number is read as input if the code is valid and the function `ResetSensor()` is called for the sensor.

```
01 sensorType = input("Enter code of the type of sensor to reset")
02 if (CheckSensorCode(sensorType)) then
03     sensorNumber = input("Please input the number of the sensor
        to reset")
04     sensorID = sensorType + sensorNumber
05     ResetSensor(sensorID)
06 endif
```

(i) Give the line number where there is concatenation.

_____ [1]

(ii) Give the identifier of a variable used in the program.

_____ [1]

(iii) Identify the data type of the data returned by the function `CheckSensorCode ()`

_____ [1]

(iv) Give the line number that contains a function call.

_____ [1]

(v) Identify TWO programming constructs that have been used in the program.

1 _____

2 _____ [2]

BLANK PAGE

- (d) The alarm system has a log that stores a record each time a sensor is triggered. This is called an event. The record format is given in the table:

| Fieldname | Description |
|-------------------|---|
| Date | The date the event happened |
| SensorID | The sensor that was activated |
| SensorType | The type of sensor that was activated – Door, Motion or Window |
| Length | The number of seconds the sensor was triggered (to the nearest second) |

The log is stored in a database table called **events**. The current contents of **events** is shown:

| Date | SensorID | SensorType | Length |
|-------------------|-----------------|-------------------|---------------|
| 05/02/2023 | WS2 | Window | 38 |
| 05/02/2023 | MS1 | Motion | 2 |
| 06/02/2023 | DS3 | Door | 1 |
| 06/02/2023 | MS2 | Motion | 3 |
| 06/02/2023 | MS1 | Motion | 2 |
| 07/02/2023 | WS1 | Window | 24 |
| 07/02/2023 | DS1 | Door | 1 |

Write an SQL statement to display the sensor IDs of the door sensors that have been triggered for more than 20 seconds.

[3]

- (f) **OCR Security Services need to identify the total number of seconds the sensors have been activated on a specific date.**

The data from the database table `events` is imported into the program written in a high-level programming language.

The program stores the data in a two-dimensional (2D) string array with the identifier `arrayEvents`

The data to be stored is shown in the table.

| Date | SensorID | SensorType | Length |
|-------------|-----------------|-------------------|---------------|
| 05/02/2023 | WS2 | Window | 38 |
| 05/02/2023 | MS1 | Motion | 2 |
| 06/02/2023 | DS3 | Door | 1 |
| 06/02/2023 | MS2 | Motion | 3 |
| 06/02/2023 | MS1 | Motion | 2 |
| 07/02/2023 | WS1 | Window | 24 |
| 07/02/2023 | DS1 | Door | 1 |

In this table, the value of `events[1, 1]` contains "MS1".

- (i) **An array can only store data of one data type. Any non-string data must be converted to a string before storing in the array.**

Identify the process that converts integer data to string data.

[1]

END OF QUESTION PAPER

