OCR
Oxford Cambridge and RSA

Thursday 25 May 2023 – Afternoon

GCSE (9–1) Computer Science

J277/02 Computational thinking, algorithms and programming

Time allowed: 1 hour 30 minutes

Do not use:
- a calculator

Please write clearly in black ink. **Do not write in the barcodes.**

Centre number □□□□□ Candidate number □□□□

First name(s) MyCSTutor.co.uk

Last name Computer Science Worked Solutions

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 **[ ]**.
- This document has **20** pages.

ADVICE
- Read each question carefully before you start your answer.

DC (ST) 326517/4

OCR is an exempt Charity

Turn over

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.

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		✓

[4]

(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`

total = num1+num2

[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.

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

[1]

(ii) Working out if a number is odd or even.

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

[1]

(iii) Finding the difference between two measurements.

```
measurement1 = 300

measurement2 = 100

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

[1]

(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")
```

Complete the following trace table for the given algorithm.

Line number	`start`	Output
01	3	
03		3
04	2	
03		2
04	1	
03		1
04	0	
03		0
04	-1	
06		Finished

[3]

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 ....

incorrect grammar in programming language

Logic error ....

Unexpected result / output

[2]

(b) Identify **two** logic errors in the pseudocode algorithm.

Write the refined line to correct each error.

Error 1 line number 02

Corrected line ....

for scoreCount - 0 to scores.length -1

Error 2 line number 03

Corrected line ....

total = total + scores[scoreCount]

[4]

3 An insertion sort is one type of sorting algorithm.

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

```
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
```

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

holds name so that the value can be swapped without being lost

[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.

We don't know how many swaps is needed - condition-controlled runs until a condition is met

[2]

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

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

Bubble sort compares pairs of values repeatedly swapping, where as insertion sort moves values into correct position once

[2]

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

1 Both create sorted lists

2 Both use loops

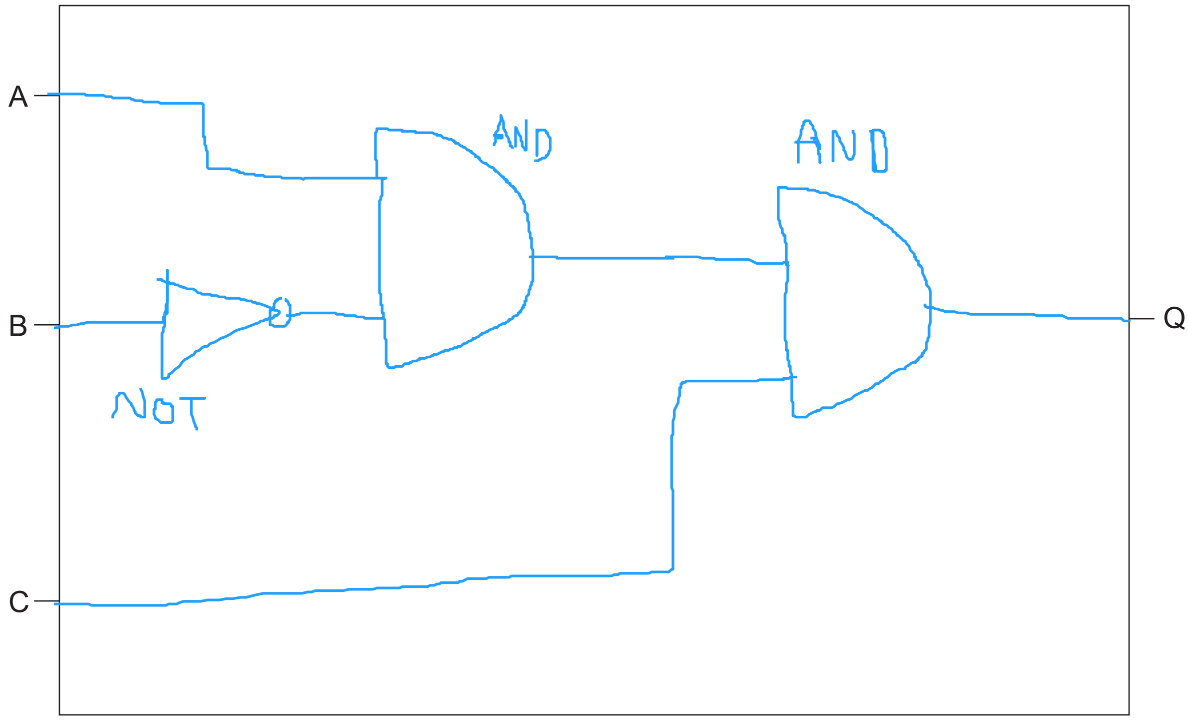
[2]

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.



[3]

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

Truth table 1:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Logic gate 1: OR ....

Truth table 2:

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Logic gate 2: AND ....

[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.

Check that program works as intended

check that intended output /result is given

check for bugs

[2]

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

Test type	Description
boundary	use data on the edge of acceptability

[2]

(iii) Complete the table by identifying **and** describing **two** features of an IDE that can be used when testing a program.

Feature	Description
Debugger	finds bugs within program
Stepping	execute program line by line

[4]

(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 Type check

Use

Check that input data is of correct data type

EG check that a input is an integer

Validation method 2 Range check

Use

Checks boundaries - makes sure input is between limits

[6]

(c) Write an algorithm to play this game. The rules are repeated from the start of the question here:

1 - the player is asked 3 addition questions
2 - each question asks the player to add together two random whole numbers between 1 and 10 inclusive
3 - if the player gets the correct answer, 1 is added to their score
4 - at the end of the game their score is displayed.

```
score = 0
 for count = 1 to 3
      num1 = random(1,10)
      num2=random(1,10)
      ans = input("What is"+num1+ " + " + num2 + "?")
      if ans = num1 + num2 then
           score = score + 1
       end if
 next count
print("You scored " + score)
```

[6]

SECTION B

We advise you to spend at least 40 minutes 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) Below is a table showing some variables within the program.

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`				✓	

[4]

(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.

```
 if SystemArmed and (DoorSensorArmed or WindowSensor) then
      SoundAlarm()
endif
```

[4]

(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.

04 **[1]**

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

sensorID **[1]**

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

Boolean **[1]**

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

Line 02 **[1]**

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

1 Selection

2 Sequence

[2]

(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.

```
SELECT SensorID
FROM events
WHERE Length > 20 AND sensorType = "Door"
```

[3]

(e) A program written in a high-level language is used to access the data from the database. This program has a procedure, `SaveLogs()`, that stores the data to an external text file.

The procedure `SaveLogs()`:

① - takes the string of data to be stored to the text file as a parameter
② - takes the filename of the text file as a parameter
③ - stores the string of data to the text file.

Write the procedure `SaveLogs()`

You must use **either**:

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

```
procedure Savelogs(data,filename) ① ②
       logFile = open(filename)
        logFile.writeLine(data) ③
       logFile.close()
endprocedure
```

[6]

(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.

Casting

.... **[1]**

(ii) Write a program that:

- 1 asks the user to input a date
- 2 totals the number of seconds sensors have been activated on the date input
- 3 outputs the calculated total in an appropriate message including the date, for example:
  `Sensors were activated for 40 seconds on 05/02/2023`

You must use **either**:

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

```
total = 0
date = input("enter date")
for item in events:
    if item[0] == date then
        total = total + item[3]
    endif

print("There were " + total + "events on " + date)
```

[6]

END OF QUESTION PAPER

ADDITIONAL ANSWER SPACE

If additional space is required, you should use the following lined page(s). The question number(s) must be clearly shown in the margin(s).