OCR
Oxford Cambridge and RSA

Tuesday 21 May 2024 – 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.
- We advise you to spend approximately 50 minutes on Section A and approximately 40 minutes on Section B.

DC (ST) 342920/6

OCR is an exempt Charity

Turn over

SECTION A

We advise you to spend approximately 50 minutes on Section A.

1 Tick (✓) **one** box in each row to identify the programming construct where each keyword is used.

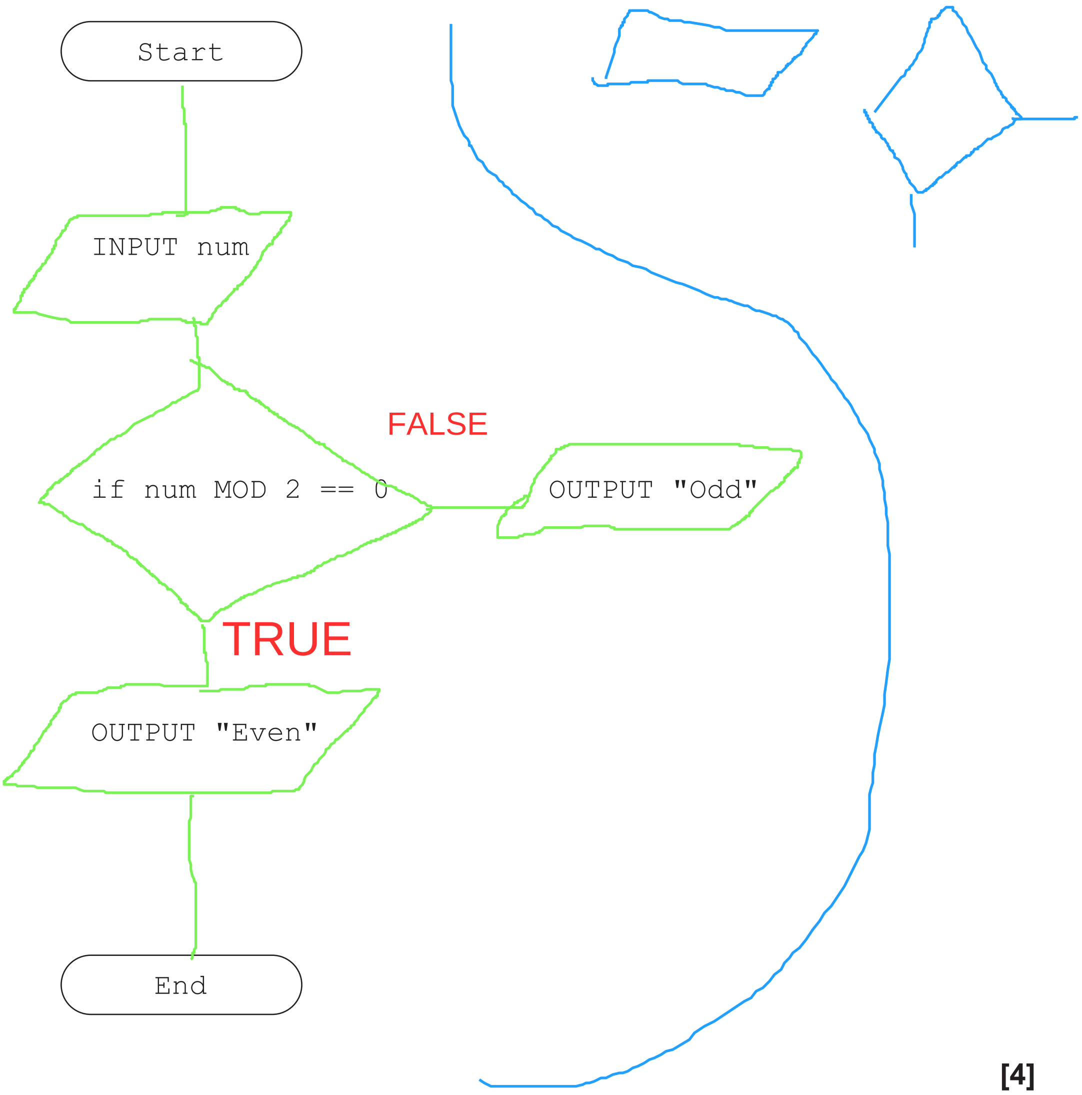
Keyword	Programming construct	
	Selection	Iteration
`if`	✓	
`for`		✓
`while`		✓

[3]

2 An algorithm decides if a number is odd or even.
An odd number divided by 2 will give the remainder 1.

The flowchart statements have been written for the algorithm, but the flowchart is incomplete.

Complete the flowchart.



[4]

3

(a) State what is meant by the term syntax error. Give **one** example of a syntax error in a program.

Definition ....

incorrect grammar in programming language

....

Example ....

printttt()

.... **[2]**

(b) A student writes an algorithm to input two numbers and add them together to create a total.

If the total is between 10 and 20 inclusive, "success" is output.

If the total is not between 10 and 20 inclusive, "warning" is output.

```
01 num1 = input("Enter a number")
02 num2 = input("Enter a number")
03 total = num1 + num1
04 if total >= 10 then
05     print("success")
06 else
07     print("warning")
08 endif
```

The algorithm does not work correctly.

Identify the line number of the **two** logic errors in the algorithm and refine the code to correct each logic error.

Line number 03 ....

Correction total = num1+num1 ....

....

Line number 04 ....

Correction if 10 <= total <= 20 ....

.... **[4]**

Turn over

(c)
(i) Show how a binary search will be used to find the number 10 in the following data set:

1 2 5 6 7 10 20

Start at middle value in sorted list (6)

As middle < target (6<10) , discard left side (smaller)

repeat until middle = target

[3]

(ii) State **one** pre-requisite for a binary search algorithm.

LIST MUST BE SORTED

[1]

(iii) Tick (✓) **one** box to identify the name of the sorting algorithm that splits data into individual items before recombining in order.

☐ Bubble sort

☐ Insertion sort

☑ Merge sort

[1]

4 A program allows users to search for and watch videos. Users give a rating to the videos they watch.

(a) Identify **one** input and **one** output for the program.

Input Rating to video

Output Result of search

[2]

(b) Describe **one** method of defensive design that can be used when creating the program.

Authentication - verifies if user is allowed to access site via username , password.

[2]

all columns where c=1
all columns where A AND B =1

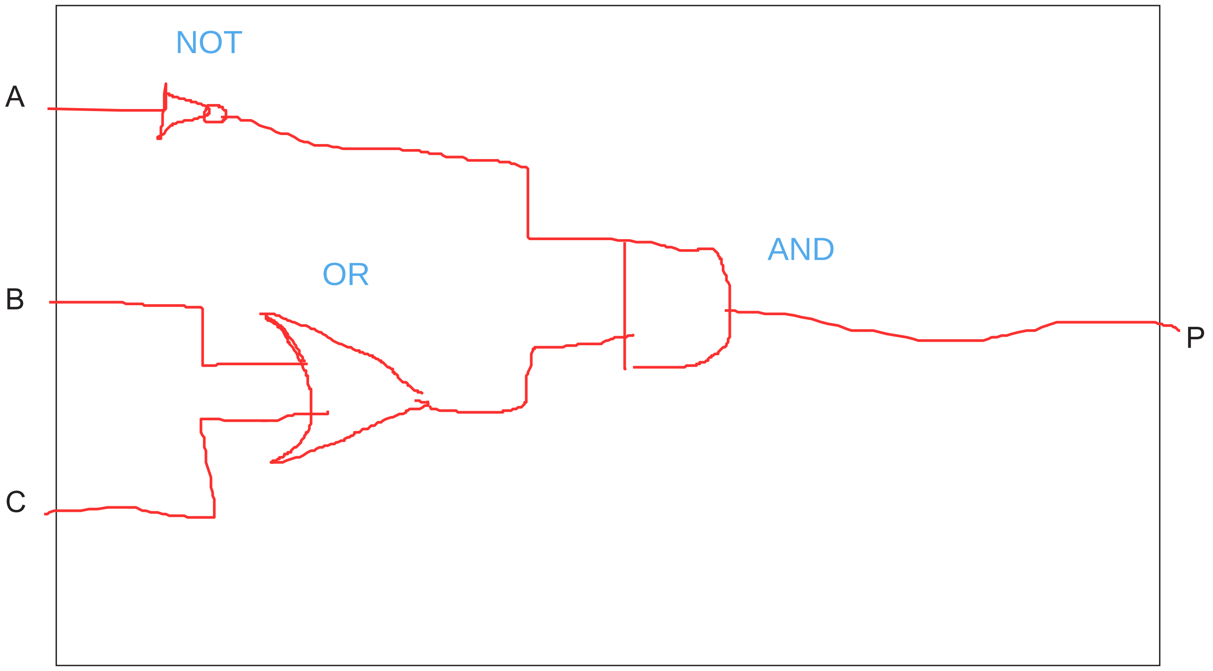
5

(a) Complete the truth table for **P = (A AND B) OR C**

A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

[4]

(b) Draw a logic circuit for **P = NOT A AND (B OR C)**



[3]

Turn over

6 The variable `message` is assigned a value.

```
message = "abcd1234"
```

(a) Complete the table to show the output when each statement executes.

The first output has been completed for you.

Statement	**Output**
`print(message.length)`	8
`print(message.upper)`	ABCD1234
`print(message.left(4))`	abcd
`print(int(message.right(4))*2)`	2468

[3]

(b) Write an algorithm in pseudocode to:

- store `"Hello"` in the variable `word1`
- store `"Everyone"` in the variable `word2`
- concatenate `word1` and `word2` to store `"HelloEveryone"` in the variable `message`

word1= "Hello"

word2 = "Everyone"

message = word1+word2

[3]

7 Programs can be written in high-level languages or low-level languages.

(a) Give **two** reasons why some programs are written in a low-level language.

1 ....

Faster execution time

....

Machine code does not have to be translated

2 ....

.... **[2]**

(b) Describe the benefits of using a compiler instead of an interpreter when writing a program.

....

Produces executable file

....

End users don't have access to source code - so can't steal or copy program

....

....

....

.... **[3]**

8 An algorithm stores the position of a character on a straight line as an integer. A user can move the character left or right.

The following algorithm:

- generates one random number between 1 and 512 (inclusive) to store as the position
- prompts the user to input a direction to move (left or right)
- takes a direction as input until a valid direction is input.

```
p = random(1, 512)
print("The position is ", p)
a = ""
while a != "left" and a != "right"
    a = input("Enter direction, left or right")
endwhile
```

(a) Describe **two** ways to improve the maintainability of the algorithm.

1 ....

Comments so easier to debug / understand code

....

....

Use subroutines to store constants that will not change- so it only has to be set once

2 ....

....

....

[4]

(b) If the character moves left, 5 is subtracted from the position.
If the character moves right, 5 is added to the position.

The position of the character can only be between 1 and 512 inclusive.

The function `moveCharacter()`:

- 1 takes the direction (left or right) and current position as parameters
- 2 changes `position` based on `direction`
- 3 sets `position` to 1 if the new position is less than 1
- 4 sets `position` to 512 if the new position is greater than 512
- 5 returns the new position.

Complete the function `moveCharacter()`

```
function moveCharacter(direction, position)

   if direction =="left" then              1
       position = position - 5             2
   elseif direction == "right" then
       position = position +5
   endif
if position < 1 then                       3
     position =1
elseif position > 512 then
     position = 512                        4
endif

      return position                      5

endfunction
```

[6]

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

SECTION B

We advise you to spend approximately 40 minutes on Section B.

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.

9 Students take part in a sports day. The students are put into teams.

Students gain points depending on their result and the year group they are in. The points are added to the team score.

The team with the most points at the end of the sports day wins.

(a) Data about the teams and students is stored in a sports day program.

(i) Identify the most appropriate data type for each variable used by the program.

Each data type must be different.

Variable	**Example**	**Data type**
`teamName`	`"Super-Team"`	string
`studentYearGroup`	`11`	integer
`javelinThrow`	`18.2`	real

[3]

(ii) The student names for a team are stored in an array with the identifier `theTeam`

An example of the data in this array is shown:

Index	0	1	2	3	4	5
Data	`Ali`	`Eve`	`Ling`	`Nina`	`Sarah`	`Tom`

`theTeam`

A linear search function is used to find whether a student is in the team. The function:

- takes a student name as a parameter
- returns `True` if the student name is in the array
- returns `False` if the student name is **not** in the array.

Complete the design of an algorithm for the linear search function.

```
function linearSearch(studentName)
    for count = 0 to theTeam.length() -1
        if theTeam[count] == studentName then
            return True
        endif
    next count
    return False
endfunction
```

[4]

(b) This algorithm calculates the number of points a student gets for the distance they throw in the javelin:

```
01    javelinThrow = input("Enter distance")
02    yearGroup = input("Enter year group")
03    if javelinThrow >= 20.0 then
04        score = 3
05    elseif javelinThrow >= 10.0 then
06        score = 2
07    else
08        score = 1
09    endif
10    if yearGroup != 11 then
11        score = score * 2
12    endif
13    print("The score is", score)
```

Complete the trace table for the algorithm when a student in year 10 throws a distance of 14.3

You may not need to use all the rows in the table.

Line number	javelinThrow	yearGroup	score	Output
01	14.3			
02		10		
06			2	
11			4	
13				The score is 4

[4]

(c) The height a student jumps in the high jump needs to be input and validated.
The height is entered in centimetres (cm) and must be between 40.0 and 180.0 inclusive.

(i) Write an algorithm to:

1 • take the height jumped as input

2 • output "VALID" or "NOT VALID" depending on the height input.

You must use **either**:

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

```
height = input("enter height")
if height <40 or h > 180 then
        print("not valid")
else
        print("valid")
endif
```

[4]

(ii) The algorithm is tested using a range of tests.

Complete the table to identify an example of test data for each type of test.

Test data (height jumped in cm)	Type of test	Expected output
80	Normal	"VALID"
180.0	Boundary	"VALID"
"subscribe"	Erroneous	"NOT VALID"

[3]

(d) The individual results for each student in each event are stored in a database.

The database table `TblResult` stores the times of students in the 100 m race. Some of the data is shown:

StudentID	YearGroup	TeamName	Time
11GC1	11	Valiants	20.3
10VE1	10	Super-Team	19.7
10SM1	10	Super-Team	19.2
11JP2	11	Champions	19.65

Complete the SQL statement to show the Student ID and team name of all students who are in year group 11

```
SELECT StudentID, TeamName

FROM TBblResult

WHERE YearGroup =11
```

[4]

(e) Abstraction and decomposition have been used in the design of the sports day program.

(i) Identify **one** way that abstraction has been used in the design of this program.

Focuses on student names and events

[1]

(ii) Identify **one** way that decomposition has been used in the design of this program.

breaks the database down into a table per event

[1]

(f) An algorithm works out which team has won (has the highest score).

Write an algorithm to:

1 - prompt the user to enter a team name and score, or to enter "stop" to stop entering new teams
2 - repeatedly take team names and scores as input until the user enters "stop"
3 - calculate which team has the highest score
4 - output the team name and score of the winning team in an appropriate message.

You must use **either**:

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

```
highscore = 0
while team !="stop":   2
     team = input("enter team")   1
     score = input("enter score")
     if score > highscore:   3
          highscore = score
          highteam = team
print(highscore, " is the highscore")   4
print(highteam, "is the winning team")
```

[6]

END OF QUESTION PAPER

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

EXTRA ANSWER SPACE

If you need extra space use these lined pages. You must write the question numbers clearly in the margin.