Pearson Edexcel Level 1/Level 2 GCSE (9–1)

Specimen Assessment Material for first teaching September 2020

Time: 2 hours	Paper Reference **1CP2/02**

Computer Science

Paper 2: Application of Computational Thinking

You must have:

- a computer workstation with appropriate programming language code editing software and tools, including an IDE that you are familiar with which shows line numbers
- a 'STUDENT CODING' folder containing code and data files
- printed and electronic copies of the Programming Language Subset (PLS) document.

Instructions

- Answer all the questions on your computer.
- Save new or amended code using the file name provided and place it in the 'COMPLETED CODING' folder.
- You must **not** use the internet at any time during the examination.

Information

- The 'STUDENT CODING' folder in your user area includes all the code and data files you need.
- The total mark for this paper is 75.
- The marks for each question are shown in brackets.

Advice

- Read each question carefully before you start to answer it.
- Save your work regularly.
- Check your answers and work if you have time at the end.

Turn over ▶

Pearson

Answer ALL questions.

Suggested time: 10 minutes

1 A program generates the shape of a box for a game. The shape of the box is based on the number that the user inputs.

Open file **Q01.py**

Amend the lines at the bottom of the code to give the:

- name of a parameter
- symbol used to show assignment
- name of a structured data type implemented as a list
- name of a built-in subprogram in the provided code
- name of a user-devised function
- name of a constant
- name of a local variable.

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total for Question 1 = 7 marks)

Suggested time: 20 minutes

2 A programmer is developing a program using turtle graphics. The program should draw a shape based on the number and length of the sides that the user inputs. All sides are the same length.

Open file **Q02.py**

Amend the code to:

- fix the syntax error on original line 5
  `inport turtle`
- fix the syntax error on original line 23
  `tomasz.pencolor "red")`
- fix the NameError on original line 11
  `tomasz == turtle.Turtle()`
- calculate the value of the variable angle by dividing 360 by the number of sides
- fix the logic error on original line 41
  `tomasz.left(70)`
- change the identifier n to a more meaningful name
- display a suitable prompt, indicating length of sides, for the user
- accept the user's input of an integer (no validation is required)
- add a comment on the original line 27, to describe why int() is needed.
  `n = int (input ("Enter number of sides, at least 3: "))`

Do **not** add any additional functionality.

Save your amended code file as **Q02FINISHED.py**

(Total for Question 2 = 10 marks)

Suggested time: 20 minutes

3 A program calculates the bus fare for passengers. The type of fare is entered using a menu. The fare is calculated and displayed. There is a choice to exit the program instead of calculating a fare.

The program is being developed. Some of the lines of code in the program are mixed up. Some sections of the program include different code options and only one of the options should be included in the program.

Open file **Q03.py**

Amend the code to make the program work and produce the correct output.

You will need to rearrange lines of code or choose between alternative lines of code.

Do **not** change the functionality of the given lines of code.

Do **not** add any additional functionality.

Save your amended code file as **Q03FINISHED.py**

(Total for Question 3 = 15 marks)

BLANK PAGE

Suggested time: 20 minutes

4 The flowchart on the facing page is for an algorithm that identifies the minimum and the maximum numbers entered by the user. The numbers are positive real numbers less than 100 000. It also calculates the mean of all the numbers. The results are displayed for the user.

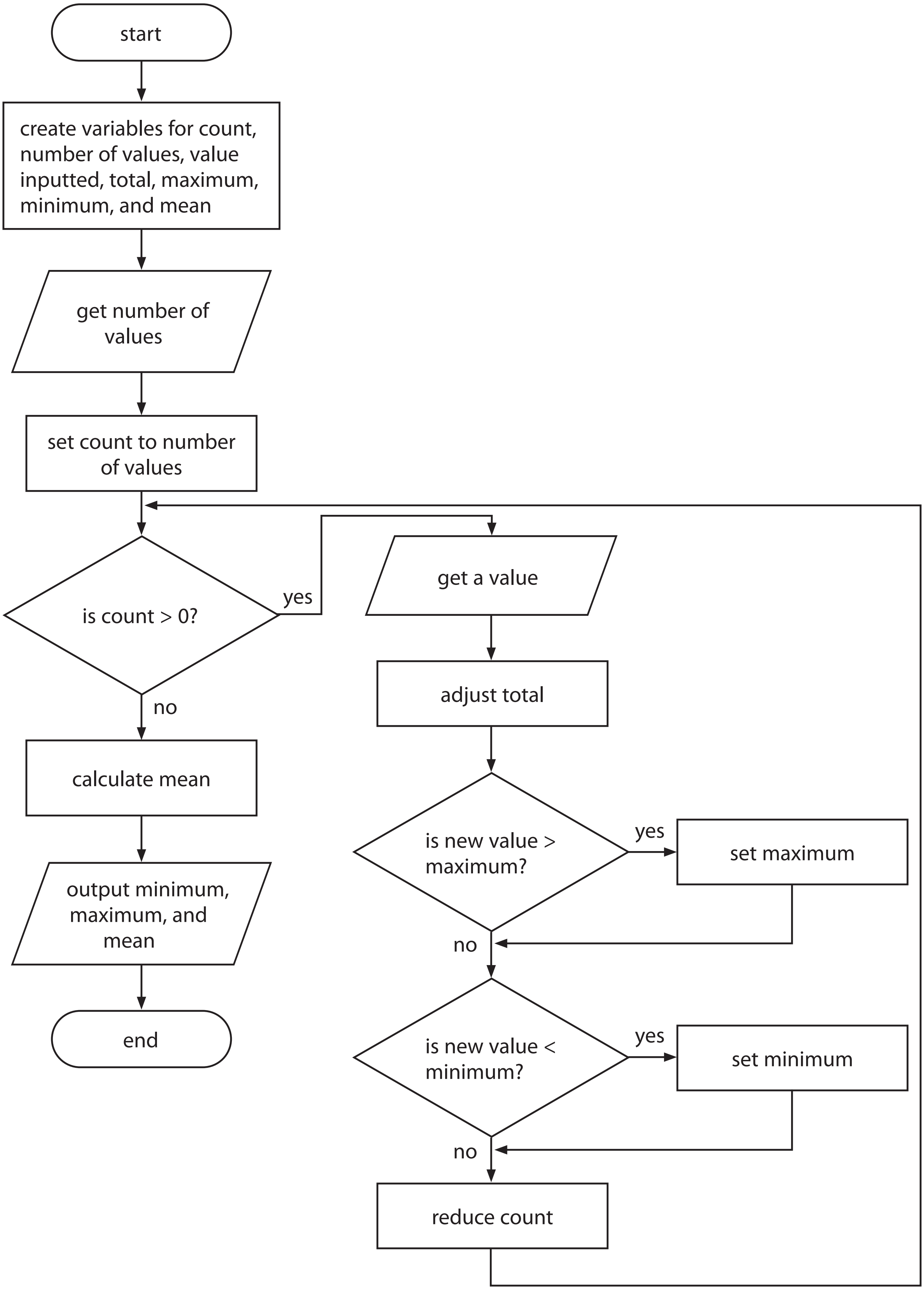
Open file **Q04.py**

Write the code to implement the algorithm in the flowchart.

Do **not** add any additional functionality.

Save your amended code as **Q04FINISHED.py**

(Total for Question 4 = 15 marks)


start
create variables for count, number of values, value inputted, total, maximum, minimum, and mean
get number of values
set count to number of values
is count > 0?
yes
no
get a value
adjust total
is new value > maximum?
yes
set maximum
no
is new value < minimum?
yes
set minimum
no
reduce count
calculate mean
output minimum, maximum, and mean
end

Suggested time: 25 minutes

5 A program is needed to track the number and health scores of native British trees. The tree data is stored as structured data. These are implemented as three lists.

The program needs to be amended to meet these requirements:

- construct a record for each type of tree consisting of fields for name, number and health score
- use commas to separate fields
- write each record to a separate line in an output file.

This is what your final output file should look like. There are exactly 7 lines in the output file.

```
Maple,1357,0.15
Ash,8421,0.18
Sycamore,9287,0.73
Birch,1043,0.38
Hazel,3743,0.92
Willow,2948,0.24
Oak,10826,0.78
```

Open file **Q05.py**

Amend the code to:

- process the lists to generate the output file.

Do **not** add any additional functionality.

Use comments, white space and layout to make the program easier to read and understand.

Save your amended code file as **Q05FINISHED.py**

(Total for Question 5 = 13 marks)

Suggested time: 25 minutes

6 A program is under development to keep track of a club's members.

The records for current members are stored as structured data implemented as a list. Each record contains a member number, last name, first name, month of birth, and year of birth.

The list is unsorted. The member number is calculated by this formula, using the modulus operator.

member number = (year of birth % month of birth) + year of birth

It will generate duplicates.

Open file **Q06.py**

Write a program to meet the following requirements:

Inputs

- Prompt for and accept a last name and a first name, no validation required.
- Prompt for and accept a month of birth, in the range of 1 to 12. You can assume only numbers will be entered.
- Prompt for and accept a year of birth, in the range of 1900 to 9999. You can assume only numbers will be entered.

Process

- Complete the subprogram to generate a member number.
- Complete the subprogram to use a linear search to ensure the member number is not already assigned.
- When the member number is not already assigned, create a new member record and append it to the member list.
- The program only needs to process a single member.

Outputs

- Display an invalid month message, if the month is out of range.
- Display an invalid year message, if the year is out of range.
- Display an informative message, if the member number is already assigned.
- Display the new member record, if a new member is added.

Do **not** add any additional functionality.

Use comments, white space and layout to make the program easier to read and understand.

Save your amended code as **Q06FINISHED.py**

(Total for Question 6 = 15 marks)

TOTAL FOR PAPER = 75 MARKS

BLANK PAGE

BLANK PAGE

BLANK PAGE