

Please check the examination details below before entering your candidate information

Candidate surname

Other names

Centre Number

Candidate Number

--	--	--	--	--

--	--	--	--

Pearson Edexcel International GCSE (9–1)

Time 3 hours

Paper
reference

4CP0/02

Computer Science

PAPER 2: Application of Computational Thinking

You must have:

A computer workstation with appropriate programming language code editing software and tools, including a code interpreter/compiler, CODES folder containing code files, and pseudocode command set (enclosed)

Total Marks

--

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and candidate number.
- Answer **all** questions.
- Answer the questions **requiring a written answer** in the spaces provided – *there may be more space than you need.*
- Only **one** programming language (Python, C# or Java) must be used throughout the examination.
- Carry out practical tasks on the computer system and save new or amended code using the name given in the question with the appropriate file extension.
- Do **not** overwrite the original code and data files provided to you.
- You must **not** use the internet during the examination.

Information

- The total mark for this paper is 80.
- The marks for **each** question are shown in brackets – *use this as a guide as to how much time to spend on each question.*
- This paper covers Python, C# and Java.
- The CODES folder in your user area includes all the code files you need.
- The invigilator will tell you where to store your work.

Advice

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

Turn over ►

P69298A

©2021 Pearson Education Ltd.

E:1/1/1/



Pearson

Answer all questions.

Answer the questions requiring a written answer in the spaces provided.

Some questions must be answered with a cross in a box ☒. If you change your mind about an answer, put a line through the box ☒ and then mark your new answer with a cross ☒.

Carry out practical tasks on the computer system and save new or amended code using the name given with the appropriate file extension.

Use only ONE programming language throughout the examination.

Indicate the programming language that you are using with a cross in a box ☒.

C#	<input checked="" type="checkbox"/>	Java	<input checked="" type="checkbox"/>	Python	<input checked="" type="checkbox"/>
----	-------------------------------------	------	-------------------------------------	--------	-------------------------------------

1 Programmers write code to solve problems.

(a) Give **one** technique that programmers use to make their code easy to read and understand.

(1)

(b) Programs may contain logic, syntax or runtime errors.

A program has three errors.

These may be logic, syntax or runtime errors.

Complete the table by adding a tick (✓) for **each** error description to show whether the error is a logic, syntax or runtime error.

(3)

Error description	Logic	Syntax	Runtime
Divide by 0			
Use x instead of * to multiply			
Subtract 10 from 2 instead of 2 from 10			



(c) **Figure 1** shows a programming construct in a flowchart.

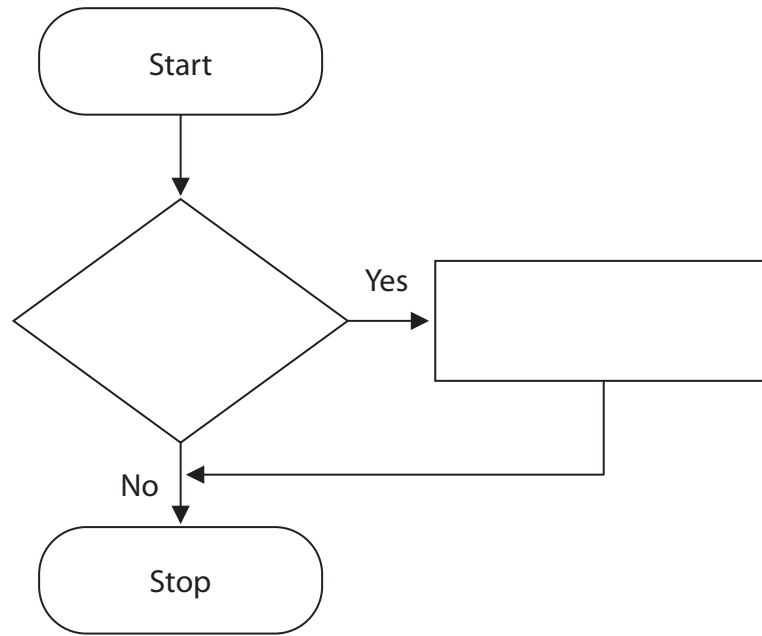


Figure 1

Identify the name of the programming construct.

(1)

- A** Array
- B** Iteration
- C** Operator
- D** Selection

(d) Open **Q01d** in the code editor.

The program inputs two numbers, compares them and outputs a suitable message.

There are **three** errors in the code.

Amend the code to correct the errors.

Save your amended code as **Q01dFINISHED** with the correct file extension for the programming language.

(3)



(e) A program is needed to calculate and print a multiplication table for a number that is input by the user.

Figure 2 shows the display when the number 2 is input.

```
Enter the number: 2
1 x 2 is 2
2 x 2 is 4
3 x 2 is 6
4 x 2 is 8
5 x 2 is 10
6 x 2 is 12
7 x 2 is 14
8 x 2 is 16
9 x 2 is 18
10 x 2 is 20
11 x 2 is 22
12 x 2 is 24
```

Figure 2

Open **Q01e** in the code editor.

Write the program.

You **must** use the structure given in the file **Q01e** to complete the program.

Do not add any further functionality.

Save your code as **Q01eFINISHED** with the correct file extension for the programming language.

(5)

(Total for Question 1 = 13 marks)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



2 Estelle is learning about sorting and searching algorithms.

(a) Estelle's friend has given her a partially completed program to carry out a binary search.

This pseudocode contains the logic required to complete the program.

```
11 RECEIVE item FROM (INTEGER) KEYBOARD
12 WHILE start <= end AND NOT found DO
13     SET middle TO (start + end) DIV 2
14     IF numberList[middle] = item THEN
15         SET found TO TRUE
16     ELSE
17         IF item < numberList[middle] THEN
18             end = middle - 1
19         ELSE
20             start = middle + 1
21         END IF
22     END IF
23     SET count TO count + 1
24 END WHILE
```

Open file **Q02a** in the code editor.

Amend the code to complete the program.

You **must** use the structure given in **Q02a** to write the program.

Do not add any further functionality.

Save your code as **Q02aFINISHED** with the correct file extension for the programming language.

(10)



P 6 9 2 9 8 A 0 5 1 6

(b) **Figure 3** shows an array of names.

0	1	2	3	4	5	6	7
Yekatrina	Juan	George	Elija	Christine	Sangeetha	Melanie	Shu-fen

Figure 3

Explain why a binary search algorithm would not find 'Juan' in the array.

(4)

.....

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



(c) A merge sort algorithm can be used to sort a list into ascending order.

The list 9, 1, 7, 6, 3, 5, 2, 8 needs to be sorted.

Complete the merge sort using the space provided.

(5)

9, 1, 7, 6, 3, 5, 2, 8

1, 2, 3, 5, 6, 7, 8, 9

(Total for Question 2 = 19 marks)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



3 Dates are often used in programs.

- (a) A program accepts the number of a month and outputs the number of days in the month.

Open **Q03a** in the code editor.

Use the code to answer these questions.

- (i) Identify the name of a local variable. (1)

- (ii) Identify the name of a data structure. (1)

- (iii) Identify the name of a built-in subprogram. (1)

- (iv) Identify the name of a user-written function. (1)

- (v) Identify the name of a parameter. (1)

- (vi) The program includes validation to ensure the month number is between 1 and 12 inclusive.

Complete the table by adding the type of test data that is being used to test the validation.

The first type of test has been completed for you. (2)

Type of test data	Test data
Normal	3
	1
	15

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



(b) Explain **one** reason why parameters are used to pass data into subprograms.

(2)

.....

.....

.....

.....

(c) A leap year is a year that has 366 days rather than 365 days. A leap year usually occurs every four years.

A year that is a leap year:

- can be exactly divided by 400

Or

- can be exactly divided by 4, but **not** exactly divided by 100.

These years are leap years: 2000, 2012 and 2020.

These years are not leap years: 1700, 1900 and 2021.

A program is needed that will accept the input of a year and output whether it is a leap year or not.

Open **Q03c** in the code editor.

Write the program.

You **must** use the structure given in **Q03c**.

Do not add any further functionality.

Save your code as **Q03cFINISHED** with the correct file extension for the programming language.

(6)

(Total for Question 3 = 15 marks)



P 6 9 2 9 8 A 0 9 1 6

4 Programs can be used to manipulate data.

(a) A program is needed to convert an 8-bit binary number into denary.

The program must:

- receive the input of a binary pattern and validate that it has exactly eight characters (there is no need to validate that it only contains 0s and 1s)
- request the input again if the binary pattern does not have exactly eight characters
- when a valid binary pattern is input:
 - convert the binary pattern into its denary equivalent
 - print a message that contains the binary pattern and its denary equivalent
- include a comment to explain how the conversion process works.

HINT

Denary placeholder	128	64	32	16	8	4	2	1
Binary number	1	0	1	1	0	1	0	0
Denary number	128	0	32	16	0	4	0	0

$$128 + 32 + 16 + 4 = 180$$

Open **Q04a** in the code editor.

Write a program to implement these requirements.

You must use the structure given in **Q04a** to write the program.

Do not add any further functionality.

Save your code as **Q04aFINISHED** with the correct file extension for the programming language.

(11)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



(b) Open **Q04b** in the code editor.

Describe the purpose of the program.

(2)

.....

.....

.....

.....

(Total for Question 4 = 13 marks)

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



5 Ann likes to create programs.

Open **Q05** in the code editor.

It contains an array of words.

Ann wants the program to:

- accept the input of a word, or the number 1 to exit. No validation of the input is required
- display all the words in the array that have the same first letter as the word that has been input
- calculate and display the number of words in the array that begin with the same letter as the word that has been input
- display all the words in the array that contain the word that has been input
- calculate and display
 - the number of words in the array that contain the word that has been input
 - the number of letters in the longest word that contains the word that has been input
 - the number of letters in the shortest word that contains the word that has been input
- display
 - the longest word that contains the word that has been input
 - the shortest word that contains the word that has been input
- repeat until the user inputs the number 1 to exit the program.

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA



Figure 4 shows the display when the word **win** is input.

```
Enter a word or 1 to exit: win
-----
wind
wire

2 word(s) begin with w
-----
wind
borrowing
swing

3 word(s) contain win
The longest word has 9 letters
The shortest word has 4 letters
The longest word is borrowing
The shortest word is wind
-----
Enter a word or 1 to exit:
```

Figure 4

Write the code for the program.

Your program should function correctly even if the number of words in the array is changed.

Save your code as **Q05FINISHED** with the correct file extension for the programming language.

The next page may be used for planning / design work.

(20)



You may use this space for planning / design work.

The content of this page will **not** be assessed.

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(Total for Question 5 = 20 marks)

TOTAL FOR PAPER = 80 MARKS



DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE



P 6 9 2 9 8 A 0 1 5 1 6

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE

