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)

Monday 10 – Wednesday 12 June 2024

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 and data 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 and data 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 ▶

P75737A

F:1/1/1/1/1/

P 7 5 7 3 7 A 0 1 1 6

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#	Java	Python
☒	☒	☒

1 Programmers write programs to solve problems.

(a) Programmers use many different techniques and tools to solve problems.

Identify what is meant by the term **abstraction**.

(1)

☒ **A** Breaking a problem down into smaller more manageable parts

☒ **B** Deciding if a program is efficient in terms of execution time

☒ **C** Drawing flowcharts with symbols used in the computer industry

☒ **D** Removing unnecessary detail to highlight the important points

(b) Open **Q01b** in the code editor.

A user enters a name and an age when the program executes.

The program should display a welcome message when the user is less than 30 years of age.

Amend the code to complete the program.

Save your amended code as **Q01bFINISHED** with the correct file extension for the programming language.

(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) Open **Q01c** in the code editor.

Use the code to answer these questions.

(i) Give the text of a comment used in the code.

(1)

.....

(ii) Give the text of a line that creates and initialises a variable.

(1)

.....

(iii) Give the keyword that starts the selection used in the code.

(1)

.....

(iv) Give the logical operator used in the code.

(1)

.....

(d) Arrays and records are data structures.

Identify the data that **must be** stored in a record rather than an array.

(1)

- ☒ **A** "King Lear", "Macbeth", "Romeo and Juliet", "The Tempest"
- ☒ **B** "King Lear", 1983, 78.32, 99
- ☒ **C** 1978, 1985, 1990, 2001
- ☒ **D** 13.99, 12.75, 10.58, 11.43

(e) Complete the table to show the data type for **each** item.

(2)

Item	Data type
45.82	
True	

(Total for Question 1 = 12 marks)

2 Programmers design, correct and test programs.

(a) Programmers design programs.

(i) Identify the term for a step-by-step description to complete a task.

(1)

☒ **A** Algorithm

☒ **B** Computational thinking

☒ **C** Decomposition

☒ **D** Pseudocode

(ii) A constant is a memory location whose value does not change during program execution.

Give the name for a memory location whose value can change during program execution.

(1)

.....

.....

(b) Programs can have runtime errors.

(i) State what is meant by the term **runtime error**.

(1)

.....

.....

(ii) Give **one** example of a runtime error.

(1)

.....

.....

(c) Open **Q02c** in the code editor.

The program generates a random number for the width of a rectangle.

The length of the rectangle is always 4

The program should calculate the perimeter of the rectangle.

The program should display the width, the length and the perimeter of the rectangle.

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

Amend the code to correct the errors.

Save your amended code as **Q02cFINISHED** with the correct file extension for the programming language.

(3)

(d) **Figure 1** shows part of an online form displayed in a browser.

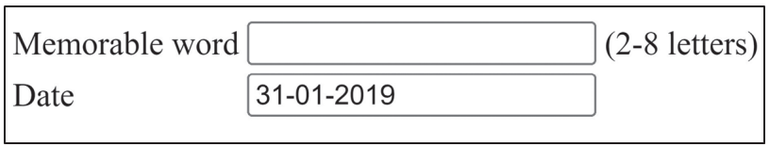

Figure 1

Complete the table to show **two** examples of erroneous, **one** example of normal and **one** example of boundary test data for the **memorable word** field.

Each example must be different.

(4)

Erroneous	
Erroneous	
Normal	
Boundary	

(Total for Question 2 = 11 marks)

3 Programs use logic, display data and validate input.

(a) The truth table shows the initial values for A and B.

Complete the truth table to show the results of each operation.

(4)

A	B	A OR B	NOT B	(A OR B) AND (NOT B)
0	0			
0	1			
1	0			
1	1			

(b) **Figure 2** shows the intended output from a program that displays every other name in an array of star names.

```
Alasia
Castor
Electra
Gudja
Izar
Kang
Maia
Ogma
```

Figure 2

Open **Q03b** in the code editor.

Amend the code to display every other name in the array of star names.

Do not add any further functionality.

Save your code as **Q03bFINISHED** with the correct file extension for the programming language.

(3)

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 program validates a number input by the user.

Figure 3 shows the output messages based on the inputted number.

Input	Output
<empty>	You must provide a number
Any negative number	The number must be greater than zero
0	The number must be greater than zero
1 to 20	Acceptable
60 or more	Acceptable
31 to 39	Centre
30	Perfect
Any other number	No message

Figure 3

Open **Q03c** in the code editor.

Amend the code to ensure the messages are generated correctly.

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 = 13 marks)

4 Programmers write code that keeps track of values, is efficient and uses subprograms.

(a) Global variables are different from local variables.

(i) State where a **global** variable is created.

(1)

.....

.....

(ii) State where a **local** variable is accessible.

(1)

.....

.....

(b) An algorithm reports the range that an integer falls within.

Integers from 0 to 39 are in the low range.

Integers from 40 to 69 are in the middle range.

Integers from 70 to 100 are in the high range.

Figure 4 and **Figure 5** show two algorithms that solve this problem.

```
1  SEND 'Enter an integer: ' TO DISPLAY
2  RECEIVE theNumber FROM (INTEGER) KEYBOARD
3  IF (theNumber < 40) THEN
4      SEND 'Low range' TO DISPLAY
5  ELSE
6      IF (theNumber < 70) THEN
7          SEND 'Middle range' TO DISPLAY
8      ELSE
9          SEND 'High range' TO DISPLAY
10     END IF
11 END IF
```

Figure 4

```
1  SEND 'Enter an integer: ' TO DISPLAY
2  RECEIVE theNumber FROM (INTEGER) KEYBOARD
3  IF (theNumber < 40) THEN
4      SEND 'Low range' TO DISPLAY
5  END IF
6  IF (theNumber > 39) AND (theNumber < 70) THEN
7      SEND 'Middle range' TO DISPLAY
8  END IF
9  IF (theNumber > 69) THEN
10     SEND 'High range' TO DISPLAY
11 END IF
```

Figure 5

Explain the reason that the algorithm in **Figure 4** is more efficient than the algorithm in **Figure 5**.

(2)

.....

.....

.....

.....

(c) A program is required to create a new key.

The program takes two inputs.

The first input is a four-character string.

The second input is a whole number.

The key is constructed by joining the first two characters from the string, the number and the final two characters from the string.

When the user enters the four-character string **abcd** and the integer **123**, the program must construct and display the new key **ab123cd**

Open **Q04c** in the code editor.

Amend the code to:

- complete the subprogram to construct the new key
- complete the call to the subprogram.

Do not add any further functionality.

Save your code as **Q04cFINISHED** with the correct file extension for the programming language.

(6)

(Total for Question 4 = 10 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

BLANK PAGE

5 Programmers refine algorithms, use searches and read data from files.

(a) **Figure 6** shows an algorithm that determines whether the animal name inputted by the user is found.

```
1  SET a1 TO 'alpaca'
2  SET a2 TO 'bear'
3  SET a3 TO 'camel'
4  SET a4 TO 'deer'
5
6  SEND ('Enter an animal: ') TO DISPLAY
7  RECEIVE target FROM (STRING) KEYBOARD
8
9  IF (a1 = target) THEN
10     SEND ('Found') TO DISPLAY
11 ELSE
12     IF (a2 = target) THEN
13         SEND ('Found') TO DISPLAY
14     ELSE
15         IF (a3 = target) THEN
16             SEND ('Found') TO DISPLAY
17         ELSE
18             IF (a4 = target) THEN
19                 SEND ('Found') TO DISPLAY
20             END IF
21         END IF
22     END IF
23 END IF
```

Figure 6

Explain **one** improvement to this algorithm that will reduce the number of variables required and will enable it to work with any number of animals.

(2)

.....

.....

.....

.....

(b) Binary search is a divide and conquer algorithm.

Here is a list of data.

10	12	13	14	15	16	17	18	19	20

A binary search is used to check whether the number 11 is in the list.

The midpoint is calculated by adding the high index to the low index and **integer dividing** by 2

Complete the table to show the values for the high index, the low index and the midpoint on each pass of a binary search.

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

(4)

high index	low index	midpoint

(c) **Figure 7** shows the **Sales.txt** file. It stores sales information.

```
264,140,120,284,192
420,377,435,376,392
619,589,606,586,600
799,811,788,814,788
982,1007,1013,989,1009
```

Figure 7

A program is required to calculate and display:

- a subtotal for each line of sales
- a grand total for all the sales in the file.

Figure 8 shows the intended output from the program.

```
1000
2000
3000
4000
5000
Grand total: 15000
```

Figure 8

Open **Q05c** in the code editor.

Amend the code to produce the intended output.

You must use the structure and variables given in **Q05c** to complete the program.

Do not add any further functionality.

Save your code as **Q05cFINISHED** with the correct file extension for the programming language.

(8)

(Total for Question 5 = 14 marks)

6 A program stores pairs of words in a two-dimensional array.

Each word in a pair starts with a different letter.

Each pair of words should be in alphabetical order, but some are not.

The last pair in the array is an empty pair.

The program must:

- replace the final blank pair of words with the variables **word1** and **word2**
- display the pair number of each pair, followed by each word in the pair, without punctuation
- display the longer word in the pair, indented
- display any pair found to be not in alphabetical order, in alphabetical order, indented, without punctuation.

Figure 9 shows part of the intended output from a functional program.

```
1 apple banana
        banana
2 wrist leg
        wrist
        leg wrist
3 blue yellow
        yellow
4 speaker keyboard
        keyboard
        keyboard speaker
```

Figure 9

Open the file **Q06** in the code editor.

Write a program to produce the intended output.

You must use the structure and variables given in **Q06** to complete the program.

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

You should use techniques to make your code easy to read.

Save your amended code as **Q06FINISHED** with the correct file extension for the programming language.

(20)

You may use this space for planning/design work.

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

(Total for Question 6 = 20 marks)

TOTAL FOR PAPER = 80 MARKS

Pearson Edexcel International GCSE (9–1)

Monday 10 – Wednesday 12 June 2024

Paper reference 4CP0/02

Computer Science

Component 2
Pseudocode command set

Resource Booklet

Do not return this Booklet with the question paper.

Turn over ▶

P75737A

F:1/1/1/1/1/

P 7 5 7 3 7 A

Pearson

Pseudocode command set

Questions in the written examination that involve code will use this pseudocode for clarity and consistency. However, students may answer questions using any valid method.

Data types

INTEGER

REAL

BOOLEAN

CHARACTER

Type coercion

Type coercion is automatic if indicated by context. For example 3 + 8.25 = 11.25 (integer + real = real)

Mixed mode arithmetic is coerced like this:

	INTEGER	REAL
INTEGER	INTEGER	REAL
REAL	REAL	REAL

Coercion can be made explicit. For example, RECEIVE age FROM (INTEGER) KEYBOARD assumes that the input from the keyboard is interpreted as an INTEGER, not a STRING.

Constants

The value of constants can only ever be set once. They are identified by the keyword CONST. Two examples of using a constant are shown.

CONST REAL PI

SET PI TO 3.14159

SET circumference TO radius * PI * 2

Data structures

ARRAY

STRING

Indices start at zero (0) for all data structures.

All data structures have an append operator, indicated by &.

Using & with a STRING and a non-STRING will coerce to STRING. For example, SEND 'Fred' & age TO DISPLAY, will display a single STRING of 'Fred18'.

Identifiers

Identifiers are sequences of letters, digits and '_', starting with a letter, for example:
MyValue, myValue, My_Value, Counter2

Functions

LENGTH()

For data structures consisting of an array or string.

RANDOM(n)

This generates a random number from 0 to n.

Comments

Comments are indicated by the # symbol, followed by any text.

A comment can be on a line by itself or at the end of a line.

Devices

Use of KEYBOARD and DISPLAY are suitable for input and output.

Additional devices may be required, but their function will be obvious from the context.
For example, CARD_READER and MOTOR are two such devices.

Notes

In the following pseudocode, the < > indicates where expressions or values need to be supplied. The < > symbols are not part of the pseudocode.

Variables and arrays		
Syntax	**Explanation of syntax**	**Example**
SET Variable TO <value>	Assigns a value to a variable.	SET Counter TO 0 SET MyString TO 'Hello world'
SET Variable TO <expression>	Computes the value of an expression and assigns to a variable.	SET Sum TO Score + 10 SET Size to LENGTH(Word)
SET Array[index] TO <value>	Assigns a value to an element of a one-dimensional array.	SET ArrayClass[1] TO 'Ann' SET ArrayMarks[3]TO 56
SET Array TO [<value>, ...]	Initialises a one-dimensional array with a set of values.	SET ArrayValues TO [1, 2, 3, 4, 5]
SET Array [RowIndex, ColumnIndex] TO <value>	Assigns a value to an element of a two dimensional array.	SET ArrayClassMarks[2,4] TO 92

Selection		
Syntax	**Explanation of syntax**	**Example**

IF <expression> THEN
  <command>
END IF
	If <expression> is true then command is executed.	IF Answer = 10 THEN
		  SET Score TO Score + 1
		END IF

IF <expression> THEN
  <command>
ELSE
  <command>
END IF
	If <expression> is true then first <command> is executed, otherwise second <command> is executed.	IF Answer = 'correct' THEN
		  SEND 'Well done' TO DISPLAY
		ELSE
		  SEND 'Try again' TO DISPLAY
		END IF

Repetition		
Syntax	**Explanation of syntax**	**Example**

WHILE <condition> DO
  <command>
END WHILE
	Pre-conditioned loop. Executes <command> whilst <condition> is true.	WHILE Flag = 0 DO
		  SEND 'All well' TO DISPLAY
		END WHILE

REPEAT <command> UNTIL <expression>	Post-conditioned loop. Executes <command> until <condition> is true. The loop must execute at least once.	REPEAT SET Go TO Go + 1 UNTIL Go = 10
REPEAT <expression> TIMES <command> END REPEAT	Count controlled loop. The number of times <command> is executed is determined by the expression.	REPEAT 100-Number TIMES SEND '*' TO DISPLAY END REPEAT

FOR <id> FROM <expression> TO
<expression> DO
  <command>
END FOR
	Count controlled loop. Executes <command> a fixed number of times.	FOR Index FROM 1 TO 10 DO
		  SEND ArrayNumbers[Index] TO DISPLAY
		END FOR

FOR <id> FROM <expression> TO
<expression> STEP
<expression> DO
  <command>
END FOR
	Count controlled loop using a step.	FOR Index FROM 1 TO 500 STEP 25 DO
		  SEND Index TO DISPLAY
		END FOR

FOR EACH <id> FROM <expression> DO <command> END FOREACH	Count controlled loop. Executes for each element of an array.	SET WordsArray TO ['The', 'Sky', 'is', 'grey'] SET Sentence to '' FOR EACH Word FROM WordsUArray DO SET Sentence TO Sentence & Word & ' ' END FOREACH

Input/output		
Syntax	**Explanation of syntax**	**Example**
SEND <expression> TO DISPLAY	Sends output to the screen.	SEND 'Have a good day.' TO DISPLAY
RECEIVE <identifier> FROM (type) <device>	Reads input of specified type.	RECEIVE Name FROM (STRING) KEYBOARD RECEIVE LengthOfJourney FROM (INTEGER) CARD_READER RECEIVE YesNo FROM (CHARACTER) CARD_READER

File handling		
Syntax	**Explanation of syntax**	**Example**
READ <File> <record>	Reads in a record from a <file> and assigns to a <variable>. Each READ statement reads a record from the file.	READ MyFile.doc Record
WRITE <File> <record>	Writes a record to a file. Each WRITE statement writes a record to the file.	WRITE MyFile.doc Answer1, Answer2, 'xyz 01'

Subprograms		
Syntax	**Explanation of syntax**	**Example**
PROCEDURE <id> (<parameter>, …) BEGIN PROCEDURE <command> END PROCEDURE	Defines a procedure.	PROCEDURE CalculateAverage (Mark1, Mark2, Mark3) BEGIN PROCEDURE SET Avg to (Mark1 + Mark2 + Mark3)/3 END PROCEDURE

FUNCTION <id>
(<parameter>, …)
BEGIN FUNCTION
  <command>
RETURN <expression>
END FUNCTION
	Defines a function.	FUNCTION AddMarks (Mark1, Mark2, Mark3)
		BEGIN FUNCTION
		  SET Total to (Mark1 + Mark2 + Mark3)/3
		  RETURN Total
		END FUNCTION

<id> (<parameter>, …)	Calls a procedure or a function.	Add (FirstMark, SecondMark)

Arithmetic operators	
Symbol	**Description**
+	Add
-	Subtract
/	Divide
*	Multiply
^	Exponent
MOD	Modulo
DIV	Integer division

Relational operators	
Symbol	**Description**
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Logical operators	
Symbol	**Description**
AND	Returns true if both conditions are true.
OR	Returns true if any of the conditions are true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

BLANK PAGE