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 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 ▶

P72406A

Q:1/1/1/e2/

P 7 2 4 0 6 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 code to solve problems.

(a) Identify the description of a variable in a computer program.

(1)

☒ **A** A value that cannot be used more than once

☒ **B** A value that must be input

☒ **C** A value that is always used in a calculation

☒ **D** A value that can change

(b) Identify the technique that improves the readability of code.

(1)

☒ **A** Using indents on every line

☒ **B** Using descriptive names for variables

☒ **C** Using the correct operators

☒ **D** Using suitable data structures

(c) Complete the table by giving an example value for **each** data type.

The first row has been completed for you.

(2)

Data type	Example value
integer	12
char	
real	

(d) Describe **one** difference between the data used in boundary testing and the data used in erroneous testing.

(2)

.....

.....

.....

.....

(e) A program should output the value 2

However, there is an error in the code and the actual output is 4

Name this type of error.

(1)

.....

(f) Open **Q01f** in the code editor.

The program should allow the input of the length of the side of a square and output the area of the square.

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

Amend the code to correct the errors.

Save your amended code as **Q01fFINISHED** with the correct file extension for the programming language.

(3)

(g) A program is needed that will accept the input of a letter and compare it with a stored letter.

It will check whether the letter comes earlier in the alphabet, later in the alphabet or is the same letter as the stored letter.

It will output the letter and the stored letter with an appropriate message.

Open **Q01g** in the code editor.

Amend the code to complete the if statement used to produce the output.

You must use the structure given in **Q01g** to complete the program.

Do not add any further functionality.

Save your code as **Q01gFINISHED** with the correct file extension for the programming language.

(4)

(Total for Question 1 = 14 marks)

2 Raza is writing a program to tell the user whether a number they input is a prime number.

A prime number is a whole number, larger than one, that can only be divided by one and itself with no remainder.

This pseudocode contains the logic required to complete the program.

```
 1 FUNCTION checkPrime(pNumber)
 2 BEGIN FUNCTION
 3    IF pNumber = 1 THEN
 4       check = False
 5    ELSE
 6       check = True
 7       FOR count FROM 2 TO pNumber DO
 8          IF pNumber MOD count = 0 THEN
 9             check = False
10          END IF
11       END FOR
12    END IF
13 RETURN check
14 END FUNCTION
15
16 SEND "Enter a number: " TO DISPLAY
17 RECEIVE number FROM (INTEGER) KEYBOARD
18 SET result TO checkPrime(number)
19
20 IF result = True THEN
21    SEND (number & " is a prime number") TO DISPLAY
22 ELSE
23    SEND (number & " is not a prime number") TO DISPLAY
24 END IF
```

(a) Answer these questions about the pseudocode.

(i) Identify a line number where **selection** starts.

(1)

.....

(ii) Identify the line number where **iteration** starts.

(1)

.....

(iii) Identify a line number where a **string** and a **number** are printed on the same line.

(1)

.....

(iv) Identify the name of a **local variable**.

(1)

.....

(v) Identify the name of a **parameter**.

(1)

.....

(b) Write a program to implement the logic in the pseudocode.

Open **Q02b** in the code editor.

Write the program.

You must use the structure given in **Q02b** to complete the program.

Do not add any further functionality.

Save your code as **Q02bFINISHED** with the correct file extension for the programming language.

(11)

(Total for Question 2 = 16 marks)

3 Manjit sells copies of a science textbook to schools.

She needs a program to process textbook orders. It must:

- accept the input of the number of textbooks required
- generate the price per textbook
- generate the total cost of the order
- display the price per textbook and the total cost of the order.

The price of one textbook depends on the number ordered:

Quantity	Price per textbook
1–5	£20.00
6–9	£15.00
10 or more	£12.00

Open **Q03** in the code editor.

Write the program.

You must use the structure given in **Q03** to complete the program.

Do not add any further functionality.

Save your code as **Q03FINISHED** with the correct file extension for the programming language.

(Total for Question 3 = 6 marks)

BLANK PAGE

QUESTION 4 BEGINS ON THE NEXT PAGE.

4 Several encryption algorithms have been developed.

(a) Identify what is meant by **encryption**.

(1)

- ☒ **A** Conversion of ciphertext into plaintext
- ☒ **B** Conversion of plaintext into ciphertext
- ☒ **C** Conversion of code into data
- ☒ **D** Conversion of data into information

(b) **Figure 1** shows a Pigpen cipher grid.

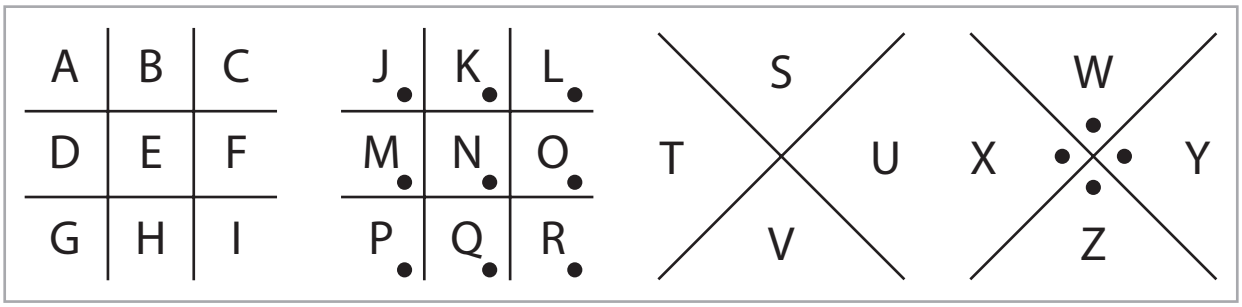

Figure 1

Complete the table by adding the symbol for each letter in the word **MAY**.

(3)

Letter	M	A	Y
Symbol			

(c) The message THE ENEMY IS NEAR is encoded using a different encryption algorithm.

Figure 2 shows the first stage of the encryption process.

T								I					
	H						Y		S				
		E				M				N			
			E		E						E		R
				N								A	

Figure 2

(i) Give the key used in the first stage of the encryption process.

(1)

.....

(ii) Give the ciphertext that is produced by the encryption algorithm.

(1)

.....

(iii) Give the name of the encryption algorithm that has been used to produce the ciphertext.

(1)

.....

(Total for Question 4 = 7 marks)

5 Julia runs a computer gaming club.

(a) She wants a program to check passwords stored in a file.

The file **passwords.txt** contains the list of passwords.

The program must:

- check each password to ensure that:
  - the first character is an uppercase letter
  - if it is, that it also includes at least one digit (0–9)
- if a password does not meet these requirements:
  - display the password
  - increment the number of incorrect passwords
- display the total number of incorrect passwords after all the passwords have been checked.

Open **Q05a** in the code editor.

Write the program.

You must use the structure given in **Q05a** to write the program.

Do not add any further functionality.

Save your code as **Q05aFINISHED** with the correct file extension for the programming language.

(9)

(b) **Figure 3** shows an array that stores player scores after a game.

3	2	10	8	1	9

Figure 3

(i) Julia uses a bubble sort algorithm to sort the scores.

Complete the table to show how the bubble sort algorithm will sort the scores.

You may not need to use all the rows.

(3)

3	2	10	8	1	9

(ii) Explain **one** reason why a bubble sort is very efficient in terms of memory usage.

(2)

.....

.....

.....

.....

(c) Describe the steps a linear search algorithm takes to find a search item.

(3)

.....

.....

.....

.....

.....

.....

(Total for Question 5 = 17 marks)

BLANK PAGE

QUESTION 6 BEGINS ON THE NEXT PAGE.

6 Carlos wants you to create a **guess the animal** game.

Open **Q06** in the code editor.

The code contains an array of animals.

It also contains a function that randomly selects an animal from the array. This is the secret word the user needs to guess.

Carlos wants the program to:

- generate the number of attempts the user has to guess the secret word.
  The maximum number of attempts is the length of the secret word +3. For example, the user has 8 attempts to guess when the secret word is **tiger**
- keep track of letters from incorrect attempts that are in the secret word and those that are not. There should be no duplicated letters
- display a message telling the user:
  - the number of letters in the secret word
  - how many attempts they have left
- force the user to input a word that is the same length as the secret word
- check whether the input word matches the secret word:
  - if the words match then a message that includes the secret word and the number of attempts taken to guess it is displayed
  - if the words do not match then:
    - letters from the attempt that appear in the secret word should be added to the correct letters store
    - letters from the attempt that do not appear in the secret word should be added to the wrong letters store
    - the contents of the correct and wrong letter stores are displayed
- allow the user another attempt until they have guessed the word or have run out of attempts
- display a message telling the user the game is over including the random word if the maximum attempts have been taken and the word has not been guessed.

Figure 4 shows the contents of the correct and wrong stores after two attempts to guess the secret word **cow**.

Secret word cow			
First attempt		**Second attempt**	
input	dog	**input**	cat
correct store	o	**correct store**	o c
wrong store	d g	**wrong store**	d g a t

Figure 4

Your program should include at least two subprograms that you have written yourself.

You must include comments in the code to explain the logic of your solution.

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

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

BLANK PAGE

Pearson Edexcel International GCSE (9–1)

Paper reference **4CP0/02**

Computer Science

Component 2
Pseudocode command set

Resource Booklet

Do not return this Booklet with the question paper.

Turn over ▶

P72406A

Q:1/1/1/e2/

P 7 2 4 0 6 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