



Mark Scheme (Results)

Summer 2026

Pearson Edexcel International GCSE
In Computer Science (4CP0)
Paper 2A: Application of Computational Thinking

Edexcel and BTEC Qualifications

Edexcel and BTEC qualifications are awarded by Pearson, the UK's largest awarding body. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.edexcel.com or www.btec.co.uk. Alternatively, you can get in touch with us using the details on our contact us page at www.edexcel.com/contactus

Pearson: helping people progress, everywhere

Pearson aspires to be the world's leading learning company. Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your students at:

www.pearson.com/uk

Summer 2026

Question Paper Log Number P79638A

Publication Code 4CPO_2A_2606_MS

All the material in this publication is copyright

© Pearson

General Marking Guidance

- All candidates must receive the same treatment. Examiners must mark the first candidate in exactly the same way as they mark the last.
- Mark schemes should be applied positively. Candidates must be rewarded for what they have shown they can do rather than penalised for omissions.
- Examiners should mark according to the mark scheme not according to their perception of where the grade boundaries may lie.
- There is no ceiling on achievement. All marks on the mark scheme should be used appropriately.
- All the marks on the mark scheme are designed to be awarded. Examiners should always award full marks if deserved, i.e. if the answer matches the mark scheme. Examiners should also be prepared to award zero marks if the candidate's response is not worthy of credit according to the mark scheme.
- Where some judgement is required, mark schemes will provide the principles by which marks will be awarded and exemplification may be limited.
- When examiners are in doubt regarding the application of the mark scheme to a candidate's response, the team leader must be consulted.
- Crossed out work should be marked UNLESS the candidate has replaced it with an alternative response.

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional Guidance	Mark
1(a)(i)	A1	The only correct answer is C <i>A is not correct because it stores true/false</i> <i>B is not correct because it stores whole numbers</i> <i>D is not correct because it stores text</i>	The correct answer is: Real	1

Question	mp	Answer	Additional Guidance	Mark	
1(a)(ii)	A2	Award one mark for each correct cell:		Accept reasonable alternative terms e.g. 'acceptable'	2
	A3				

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
1(b)(i)	B1	Line 6		1
1(b)(ii)	B2	weight	Accept weight initialisation e.g. <ul style="list-style-type: none"> <code>weight = 0</code> do not accept line number	1
1(b)(iii)	B3	Award one mark for any of the following <ul style="list-style-type: none"> get height calculate bmi 	do not accept line number ignore case and comment indicators	1
1(b)(iv)	B4	Line 6	accept range of lines e.g. lines 6 to 7 (must start with line that contains <code>while</code>)	1
1(b)(v)	B5	Line 12		1

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
1(c)		Award one mark for each of:		
	C1	Correct spelling of input	Line 15	
	C2	Change multiply to plus minsWeek = minsWeek + minsToday	Line 16	
	C3	Change if level == "e" and minsWeek > 300 to if level == "e" and minsWeek < 300	Line 18 Accept <=	
	C4	Change elif level == "c" or minsWeek < 200: to elif level == "c" and minsWeek < 200:	Line 21 Ignore <=	
				4

Paper 2 – Application of Computational Thinking

```
1 # Q01c
2
3 # Program variables
4 days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
5 minsWeek = 0
6 level = "m"
7 minsToday = 0
8
9 # Correct the following code
10
11 level = input("Please enter e (elite) or c (club): ") # get level
12
13 # get total minutes exercise for week
14 for day in days:
15     minsToday = int(input("Please enter total minutes exercise on " + day + " >"))
16     minsWeek = minsWeek + minsToday
17
18 if level == "e" and minsWeek < 300: # check if elite runner has completed enough exercise
19     print("Weekly exercise:", minsWeek, "minutes")
20     print("You have not met your exercise target of 300 minutes.")
21 elif level == "c" and minsWeek < 200: # check if club runner has completed enough exercise
22     print("Weekly exercise:", minsWeek, "minutes")
23     print("You have not met your exercise target of 200 minutes.")
24 else:
25     print("Weekly exercise:", minsWeek, "minutes")
26     print("Well done, you have met your weekly exercise target.")
```

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional Guidance	Mark
2(a)	A1	80: Just Right (1)	Ignore quote marks, capital letters and minor spelling errors ignore an additional attempt to receive new number (line 18)	1
	A2	-2: End (1)		1
	A3	16: Low Evens (1)		1

Question	mp	Answer	Additional Guidance	Mark
2(b)	B1	<p>The only correct answer is D</p> <p><i>A is not correct because encryption is making data meaningless</i> <i>B is not correct because a function is a type of subprogram</i> <i>C is not correct because repetition is a type of loop</i></p>	The correct answer is Selection	1

Question	mp	Answer	Additional guidance	Mark
2(c)	C1 C2	<p>Award up to two marks for a linked explanation such as:</p> <ul style="list-style-type: none"> • The program will crash (1) because the wrong data type is entered (1) • A type error occurs (1) because this string cannot be used as an int (1) • A runtime error occurs (1) because program expected an integer (1) • Value error is given (1) as this string cannot be converted to an int (1) 	<p>Error on its own is not enough.</p> <p>Can mix and match as long as link makes sense.</p>	2

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
2(d)	D1	Award up to two marks for a linked description such as:	allow answers relating to Figure 2 e.g: Divides number by 2 (1) and outputs 1 or 0 / determines if number is odd or even (1)	2
	D2	<ul style="list-style-type: none"> Divides (1) (one number by another) and returns the remainder (1) 		

Question	mp	Answer	Additional guidance	Mark
2(e)		Award one mark for each of:		4
	E1	Add <code>int()</code> to height and radius input	award mark for either height or radius input (don't need both for this mark)	
	E2	Correct volume formula <code>PI * radius * radius * height</code> <code>height * radius**2 * math.pi</code> <code>PI * math.pow(radius, 2) * height</code>	Award any correct formula using given variables, in any order Ignore typos Allow FT from E1 (incorrect data types of variables)	
	E3	Use of constant <code>PI</code> in the formula	Must use exact constant name Allow consistent change of constant name	
	E4	Output message combining str and int <code>print(volume, "cm3")</code>	Must include <code>volume</code> and any suitable string Suitable string includes "volume" or "cm3"	

```
1 # Q02e
2
3 # import math library to access PI constant
4 import math
5
6 # program variables
7 radius = 0
8 height = 0
9 volume = 0.0
10
11 # constant PI taken from math library
12 PI = math.pi
13
14 # complete the code to get integer values for radius and height
15 radius = int(input("Enter cylinder radius in cm: "))
16 height = int(input("Enter cylinder height in cm: "))
17
18 # complete the code to calculate the area of the cylinder
19 volume = (PI * radius ** 2) * height
20
21 # rounding volume to whole number before output message
22 volume = round(volume)
23
24 # add a line of code to output volume within a suitable message
25 print("Cylinder volume: ", volume, "cm3")
```

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
3(a)		Award one mark for each of:		
	A1	deposit and period taken as input	must contain appropriate prompt	
	A2	deposit and period as correct data types (deposit: float, and period: int)	Allow FT from A1 if period and year taken as input	
	A3	While loop used and terminates correctly using period this mark requires both: while period > 0: (condition) and period - 1 (decrement)	Allow FT from A2 (period not int type)	
	A4	Pseudocode line 17 formula coded correctly <div>17 SET value TO value + value * INTEREST</div>	value = value + value * INTEREST value = value * INTEREST + value value = value * (1 + INTEREST) value = (1 + INTEREST) * value value += value * INTEREST	
	A5	Counter implemented for year		
	A6	Appropriate output as indicated for each year	year values must be correct (1, 2, 3 etc..) Allow infinite loop where year is incrementing and displayed Any value for savings – must be present Message should appear as given in pseudocode – allow missing currency Do not accept hard coded	
	A7	All correct output	Allow other currencies for MP6 and MP7 Allow use of round to get int for earned	
				7

Test data for examiners

deposit	period	earned (output)
2500	5	<pre> Please enter the amount of money you wish to invest: £2500 Please enter the number of years you wish to invest for: 5 Year 1 Savings value: £ 2612 Year 2 Savings value: £ 2730 Year 3 Savings value: £ 2852 Year 4 Savings value: £ 2981 Year 5 Savings value: £ 3115 Total earned: £ 615 </pre>
1000	10	<pre> please enter the amount of money you wish to invest: £1000 Please enter the number of years you wish to invest for: 10 Year 1 Savings value: £ 1045 Year 2 Savings value: £ 1092 Year 3 Savings value: £ 1141 Year 4 Savings value: £ 1192 Year 5 Savings value: £ 1246 Year 6 Savings value: £ 1302 Year 7 Savings value: £ 1360 Year 8 Savings value: £ 1422 Year 9 Savings value: £ 1486 Year 10 Savings value: £ 1552 Total earned: £ 552 </pre>

Paper 2 – Application of Computational Thinking

```
1 # Q03a
2
3 # program variables
4 deposit = 0.0
5 value = 0.0
6 earned = 0.0
7 period = 1
8 year = 0
9
10 INTEREST = 0.045 # 4.5% fixed annual interest rate
11
12 # write your code below this line
13 # -----
14 deposit = float(input("Please enter the amount of money you wish to invest: £"))
15 period = int(input("Please enter the number of years you wish to invest for: "))
16
17 value = deposit
18 while period > 0:
19     value = value + value * INTEREST
20     year = year + 1
21     period = period - 1
22     print("Year", year, "Savings value: £", int(value))
23
24 earned = value - deposit
25 print("Total earned: £", int(earned))
```

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional Guidance	Mark
3(b)(i)	B1 B2	<p>Award one mark for any of the following up to a maximum of two marks:</p> <ul style="list-style-type: none"> Immutable/value cannot be changed (while running) (1) Declared once / declared at beginning (1) Value set by programmer / set at compile time (1) Unique identifier (1) Has a data type (1) 	<p>'named location in RAM' acceptable for BP4</p> <p>Accept name in UPPERCASE (convention rather than characteristic)</p>	2

Question	mp	Answer	Additional guidance	Mark
3(b)(ii)	B1 B2	<p>Award up to two marks for a linked description such as:</p> <ul style="list-style-type: none"> Global variables are accessible from anywhere in a program (1), whereas local variables can only be accessed from within the scope/subprogram/subroutine in which they are declared (1) Local variables are declared within subprograms/subroutines (1), whereas global variables are declared in the main program / outside of subprograms/subroutines (1) Local variables are created when a subroutine/subprogram starts and are deleted when it ends (1), whereas global variables are in RAM for the entire running/execution of the program (1) 		2

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional Guidance	Mark
4(a)	A1	<p>The only correct answer is C</p> <p><i>A is not correct because a library is a collection of subprograms</i> <i>B is not correct because a parameter is a local variable</i> <i>D is not correct because a string stores a sequence of characters</i></p>	The correct answer is record	1

Question	mp	Answer	Additional guidance	Mark
4(b)	B1 B2	<p>Award one mark for data structure:</p> <ul style="list-style-type: none"> • Array/list (1) <p>Award one mark for any of the following Justifications:</p> <ul style="list-style-type: none"> • The values are all the same data type (1) • Multiple values are stored (under a single variable name) (1) • Because a loop can iterate across the values in the array (1) • Because individual values can be indexed (1) • Array size is known in advance (max 31) 		2

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
4(c)	C1 C2	<p>Award up to two marks for a linked explanation such as:</p> <ul style="list-style-type: none"> • It is easier to see what the code is intended to do (1) because the implementation is hidden (1) • There is less code to read through (1) because lots of the code is contained in the subprograms / separate files (libraries) (1) • What the code is doing is clear / easier to understand (1) because subprograms names reflect their purpose (1) • The amount of code (in main) is reduced (1) as unnecessary/repetitive code is removed (1) 		2

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
4(d)		Award one mark for each of:		
	D1	Display last 2 numbers to user	Syntactically correct output statement with "string" and 2 int variables Not hard coded	
	D2	Get int input from user and Output suitable message for each correctly entered value e.g. correct	<code>Var = int(input("..."))</code>	
	D3	Accurate comparison of user input to calculated value. Must be syntactically correct if/while statement e.g. <code>if userInput == nextNumber:</code> <code>while userInput == nextNumber:</code>	Ignore incorrect calculation of nextNumber. userInput must be the variable taken for D2	
	D4	Calculate next number in sequence and accurately update sequence values	<code>lastNumber</code> and <code>secondLastNumber</code> must hold correct values throughout. Candidate may rename or have own version of these variables	
	D5	End program only when incorrect number entered – must follow Fibonacci sequence	Accept while True / break Do not award this mark as FT	
	D6	Program output matches Fibonacci sequence for: <ul style="list-style-type: none"> 1,2,3,5.. 9 to exit loop Include suitable final message and accurate number of values added (4 correct values added)	including appropriate messages	6

```
1 #Q04d
2
3 # program variables
4 lastNumber = 1
5 secondLastNumber = 0
6 countCorrect = 0
7 userInput = 0
8
9 # write your code below this line
10 # -----
11
12 # loop until incorrect sequence number entered
13 endProgram = False
14 while endProgram != True:
15
16     # output last two numbers with a suitable message
17     print("Last two numbers in the sequence are", secondLastNumber, "and", lastNumber )
18
19     # calculate next number
20     nextFibonacci = lastNumber + secondLastNumber
21
22     # get user input
23     userInput = int(input("Enter the next number in the sequence: "))
24
25     # check if user has input correct number and take appropriate steps
26     if userInput == nextFibonacci:
27         print("Correct \n")
28         secondLastNumber = lastNumber
29         lastNumber = nextFibonacci
30         countCorrect = countCorrect + 1
31     # deal with incorrect user input
32     else:
33         endProgram = True
34
35 # tell user how many numbers they added to the fibonacci array
36 print("Incorrect. Numbers you added to sequence:", countCorrect)
```

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
5(a)	A1	7 / seven (1)		1

Question	mp	Answer			Additional guidance	Mark
5(b)	B1 B2 B3 B4		Number	Order compared	All FT on 1 st value being in 7 th row only. 2 FT marks for this response	4
			2328			
			3369			
			3928			
			4091	2		
			5017	4		
			5909	3		
			6266			
			6499	1		
			7148			
			7789			
			8027			
			8412			
			9027			
			9031			
			9256			

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
5(c)				
	C1	Counter controlled accurately	Both; increment and reset after each run	
	C2	Method for checking if run has ended and only then adding LN to string e.g. <ul style="list-style-type: none"> if <string>[i] == <string>[i-1]: if char_current == char_previous: 	Any method Must work e.g. Correct IF and <string> += LN inside ELSE Correct While and <string> += LN outside loop	
	C3	Loop to process all items in instruction string	Loop that targets string/string length	
	C4	Counter cast to string/char	Any method Must target an int that candidate reasonably expects is a run count	
	C5	Construct an accurate compressed string	Ignore missed E3 at end, missing N3 at start or missing N1 in middle Allow swapping order of N and L Do not award <ul style="list-style-type: none"> N11W19S13 N1N2N3W4W5....E49 	
	C6	Construct fully accurate compressed string	Including N3 at start, N1 in middle and E3 at end	
	C7	Open (for writing) and close file	Must include 'w' (accept w+) Constant FILENAME not necessary for this mark Accept file handling methods: with open(FILE, 'w') as filename	
	C8	String written to file	Even if compressed string not correct If required, examiner can add line to close text file to award this mark	
	C9	Appropriate use of FILENAME constant and final message output		
				9

Paper 2 – Application of Computational Thinking

```
1 # Q05c
2
3 # program variables
4 instructionString = "NNNWWWSSSSSEENWWWWWWWWWSSSSWWWWSSSSWNNNNNNNEEE"
5 compressedString = ""
6 count = 0
7 previousChar = instructionString[0]
8 FILENAME = "instructions.txt"
9
10 # -----
11 # write your code below this line
12
13 # iterate over the string
14 for thisChar in instructionString:
15
16     # count number of occurrences for each direction
17     if thisChar == previousChar:
18         count = count + 1
19     # update compressed string
20     else:
21         compressedString = compressedString + previousChar + str(count)
22         count = 1
23     # update previous char at end of each run
24     previousChar = thisChar
25
26 # deal with final character
27 compressedString = compressedString + previousChar + str(count)
28
29 # write string to file
30 myFile = open(FILENAME, "w")
31 myFile.write(compressedString)
32 myFile.close()
33
34 # output confirmation message
35 print("Instruction file has been created")
```

Paper 2 – Application of Computational Thinking

Question	mp	Answer	Additional guidance	Mark
6		Award one mark for each of:		
	A1	Integer input taken for player choice of category	Does not have to be validated	
	A2	Selection used to compare category values		
	A3	Correct use of 2D indexing	Must get at least 1 inner item	
	A4	Appropriate winner/lose/draw string displayed to console		
	A5	Mechanism for accurately tracking player/computer scores. Must include all 3: <ul style="list-style-type: none"> • <code>playerScore += 1</code> (when player wins) • <code>computerScore = computerScore + 1</code> (when cpu wins) • draw has no increment (when both elements are ==) 	must be within selection statement	
	A6	User input validated accurately. Must work in all rounds	does not need additional error message	
	A7	Loop controlled by half length of the deck array or length of accurately generated player/cpu array	do not award hard coded must play 5 rounds do not award full deck array unless step of 2	
	A8	Mechanism for dealing unique cards to player and computer and no card repeated	e.g. alternate cards or first half and second half or <code>.pop()</code> etc.	
	A9	Player card accurately displayed each round	must be within loop and appear at least once	
	A10	Accurate tracking and display of (win/lose/draw) scores		11
	A11	All five rounds played with all correct output including computer cards and final scores	Ignore missed validation for this mark	

LBMS (AOs: 3xAO3.2a, 3xAO3.2b, 3xAO3.3c) (Tgt: 3, 3, 3); Spec: 1.1.6(2), 1.2.1(2), 1.2.2(2), 2.1.1, 2.1.2, 2.1.5

PBMS: 2.1.1, 2.3.2(2), 2.3.3, 2.4.1, 2.5.1(2), 2.6.1(3), 2.6.3

Paper 2 – Application of Computational Thinking

Award up to a maximum of nine marks using the levels-based mark scheme below.				Mark
Band 0	Band 1 (1-3 marks)	Band 2 (4-6 marks)	Band 3 (7-9 marks)	
No rewardable content	Little attempt to decompose the problem into component parts	Some attempt to decompose the problem into component parts	The problem has been decomposed into component parts	(9)
	Some parts of the logic are clear and appropriate to the problem	Most parts of the logic are clear and mostly appropriate to the problem	The logic is clear and appropriate to the problem	
	Some appropriate use and manipulation of data types, variables, data structures and program constructs	The use and manipulation of data types, variables and data structures and program constructs is mostly appropriate	The use and manipulation of data types, variables and data structures and program constructs is appropriate	
	Parts of the code are clear and readable	Code is mostly clear and readable	Code is clear and readable	
	Finished program will not be flexible enough with other data sets or input	Finished program will function with some but not all other data sets or input	Finished program could be used with other data sets or input	
	The program meets some of the given requirements	The program meets most of the given requirements	The program fully meets the given requirements	

Paper 2 – Application of Computational Thinking

```

1  # Q06
2
3  # program variables
4  pack = [[3,4,8],[4,2,16],[3,6,16],[2,10,8],
5          [3,2,16],[2,8,16],[4,4,4],[4,4,32],
6          [4,6,16],[3,4,8]]
7
8  # -----
9  # Write your code below this line
10 categories = ["Storage TB", "Cores", "RAM GB"]
11
12 playerCards = []
13 computerCards = []
14
15 playerScore = 0
16 computerScore = 0
17
18 playerSelection = 0
19
20 # deal cards
21 for i in range(len(pack)):
22     if i % 2 == 0:
23         playerCards.append(pack[i])
24     else:
25         computerCards.append(pack[i])
26
27 # play game
28 for card in range(len(playerCards)):
29     print("\nYour card:")
30     for category in range(0,3):
31         print(categories[category] + ":", playerCards[card][category])
32     playerSelection = 0
33
34     # get player choice of category
35     while playerSelection not in range(1,4): # allow only valid input 1-3
36         playerSelection = int(input("Please enter 1 for Storage, 2 for Cores or 3 for RAM\n>"))
37     playerSelection = playerSelection - 1
38
39 # show computer card
40 print("\nComputer card:")
41 for category in range(0,3):
42     print(categories[category] + ":", computerCards[card][category])
43

```

```
44 # decide who wins
45 if playerCards[card][playerSelection] == computerCards[card][playerSelection]:
46     print("\nDraw \n")
47 elif playerCards[card][playerSelection] > computerCards[card][playerSelection]:
48     print("\nYou win!\n")
49     playerScore = playerScore + 1
50 else:
51     print("\nYou lose!\n")
52     computerScore = computerScore + 1
53
54 # print final Score
55 print("Final score \nPlayer", playerScore, "-", computerScore, "Computer")
```