



Mark Scheme (Results)

Summer 2025

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 2025

Question Paper Log Number P78639A

Publications Code 4CP0_2A_2506_MS

All the material in this publication is copyright

© Pearson Education Ltd 2025

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.

Mark Scheme - Python

Question	mp	Answer	Additional Guidance	Mark
1(a)	A1	The only correct answer is B <i>A is not correct because for is used in iteration</i> <i>C is not correct because new is used to declare an object</i> <i>D is not correct because while is used in repetition</i>		1

Question	mp	Answer	Additional Guidance	Mark																
1(b)(i)	B1	Award one mark for each correct cell:																		
	B2																			
	B3																			
		<table><thead><tr><th>Description</th><th>Syntax error</th><th>Logic error</th><th>Runtime error</th></tr></thead><tbody><tr><td>The program does not produce the expected output.</td><td></td><td>✓</td><td></td></tr><tr><td>The program does not translate.</td><td>✓</td><td></td><td></td></tr><tr><td>The program crashes during execution.</td><td></td><td></td><td>✓</td></tr></tbody></table>	Description	Syntax error	Logic error	Runtime error	The program does not produce the expected output.		✓		The program does not translate.	✓			The program crashes during execution.			✓		3
Description	Syntax error	Logic error	Runtime error																	
The program does not produce the expected output.		✓																		
The program does not translate.	✓																			
The program crashes during execution.			✓																	

Question	mp	Answer	Additional guidance	Mark
1(b)(ii)	B4	Line 15 - : added to end of line		3
	B5	Line 20 – 0 changed to 1		
	B6	Line 22 – missing comma added between string and greenCount		

```

1 # Q01bii
2
3 # initialise counters
4 ballCount = 0
5 greenCount = 0
6 redCount = 0
7
8 ballArray = ["green", "red", "green", "green", "red", "green", "green", "green",
9             "green", "green", "green", "red", "green", "red", "green", "red",
10            "red", "green", "green", "red", "red", "green", "green", "red",
11            "green", "red", "red", "green", "green", "red", "green", "green",
12            "red", "green", "green", "green", "red", "green", "red", "green"]
13
14 # iterate over ball array
15 for ball in ballArray:
16     ballCount = ballCount + 1
17     if ball == "green":
18         greenCount = greenCount + 1
19     else:
20         redCount = redCount + 1
21
22 print(" Green: ", greenCount, "balls \n", "Red: ", redCount,
23       "balls \n", "Total: ", ballCount, "balls")

```

Question	mp	Answer	Additional guidance	Mark
1(c)(i)	C1	Award one mark for: - To store a value that must not be changed/prevent accidental change of value - To simplify code maintenance/so that if the value must be updated the change only has to made in a single place - To avoid inconsistency error of specifying same value multiple times - To make code more readable/give meaningful names to values	Accidental change cannot refer to 'user'	1

Question	mp	Answer	Additional guidance	Mark
1(c)(ii)	C2 C3	Award two marks for a linked description such as: A variable stores a single value (1) whereas an array stores multiple values (1)		2

Question	mp	Answer	Additional guidance	Mark
1(d)(i)	D1	Award one mark for: getWages (1)	Accept 'main' Accept: <pre>def getWages def getWages (inSales, inAge) line 7 (Python)</pre>	1

Question	mp	Answer	Additional guidance	Mark
1(d)(ii)	D2	Award one mark for: and (1)	Do not accept line numbers Accept: &&	1

Question	mp	Answer	Additional guidance	Mark
1(d)(iii)	D3	Award one mark for any of the following: - sales (1) - wages (1) - inSales (1)	Do not accept line numbers	1

Question	mp	Answer	Additional guidance	Mark
1(d)(iv)	D4	Award one mark for any of the following: - inSales (1) - inAge (1)	Do not accept line numbers	1

Question	mp	Answer	Additional Guidance	Mark
2(a)	A1	The only correct answer is D <i>A is not correct because abstraction is a technique used in computational thinking</i> <i>B is not correct because encryption is a security measure</i> <i>C is not correct because iteration is a programming construct</i>		1

Question	mp	Answer	Additional guidance	Mark
2(b)(i)		Award one mark for each correct cell:		
	B1 B2	real/double/float: (1) • pay rate/18.75 (1) • total pay/356.25 (1) • tax/0.2/71.25 (1) • income/285.00 (1) int/integer: (1) • hours/19 (1)	Example must match data type for B2 Award suitable example even if not taken from payslip Accept string and matching example e.g. Hours	2

Question	mp	Answer	Additional guidance	Mark
2(b)(ii)		Award one mark for each correct cell:		3
	B3	Input: - Pay Rate/18.75 (1) - Hours/19 (1)		
	B4	Process: - Pay Rate * Hours/18.75 * 19 (1) - Total Pay * Tax/356.25 * 0.2 (1) - Total Pay – Tax/356.25 – 71.25 (1)	Don't need output e.g. total pay, just the <u>calculation part</u> e.g. pay rate * hours: Total pay = <u>pay rate * hours</u> Accept 0.2 or 20% as part of calculation for tax Ok to mix types e.g. Total Pay * 20% Accept descriptions such as <i>multiply the pay rate by hours</i> <i>(hours etc could be taken from input box/ B3 mark)</i>	
	B5	Output: - Total Pay/356.25 (1) - Tax/71.25 (1) - Income/285 (1)		

Question	mp	Answer	Additional Guidance	Mark
2(c)(i)	C1 C2	Award one mark for one correct cell: Award two marks for all three correct cells: • Normal: any value between 0-255 e.g. 125 • Boundary: 0/255 • Erroneous: any negative value/any value 256 or above	Accept -1/256 for boundary data Don't accept non-int types	2

Question	mp	Answer	Additional Guidance	Mark
2(c)(ii)		Award one mark for any of the following up to a maximum of four marks:		4
	C3	Input changed to int before comparison (1)		
	C4	Relational operator used correctly to check for valid input (both ≥ 0 and < 256 or equivalent) (1)	Allow: > 0 and/or < 255	
	C5	'and' logical operator used to combine relational tests (1)	or operator acceptable if candidate rearranges the selection statement and test works for correct code option – check if C4 is now awardable	
	C6	Calling of <code>getHex (userInput)</code> (1)	Call must be inside <code>print ()</code> Subprogram and argument must be functionally correct Ignore additional variables/strings in print statement	

```
37 # complete the line to convert the user input to integer
38 userInput = int(input("please enter a number between 0 and 255 \n"))
39
40 # complete the test to check the input is between 0 and 255
41 if (userInput >=0) and (userInput <256):
42     # complete the line to call getHex subprogram
43     print(getHex(userInput))
44 else:
45     print("number out of range")
```

Question	mp	Answer	Additional Guidance	Mark																																																																																																																
3(a)(i)	A1 A2	Award one mark for each: • Pass 1 correct (1) • Pass 2 correct (1) <table><tr><td>9</td><td>11</td><td>12</td><td>19</td><td>14</td><td>18</td><td>15</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>18</td><td>15</td><td>17</td><td>19</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>15</td><td>17</td><td>18</td><td>19</td></tr></table> (1) (1) Must show ONLY the 19 moving in first pass and ONLY the 18 moving in second pass	9	11	12	19	14	18	15	17	9	11	12	14	18	15	17	19	9	11	12	14	15	17	18	19	Ignore lines with individual swaps and only award correct completed passes <table><tr><td>9</td><td>11</td><td>12</td><td>19</td><td>14</td><td>18</td><td>15</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>19</td><td>18</td><td>15</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>18</td><td>19</td><td>15</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>18</td><td>15</td><td>19</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>18</td><td>15</td><td>17</td><td>19</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>15</td><td>17</td><td>18</td><td>19</td></tr></table> (1) (1) Accept accurate right-to-left bubble sort as shown below <table><tr><td>9</td><td>11</td><td>12</td><td>19</td><td>14</td><td>18</td><td>15</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>19</td><td>15</td><td>18</td><td>17</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>15</td><td>19</td><td>17</td><td>18</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>15</td><td>17</td><td>19</td><td>18</td></tr><tr><td>9</td><td>11</td><td>12</td><td>14</td><td>18</td><td>15</td><td>17</td><td>19</td></tr></table> (1) (1) Complete sort cannot be the 1st /only line in response – 0 marks as this will be taken as both 18 and 19 moving in first pass	9	11	12	19	14	18	15	17	9	11	12	14	19	18	15	17	9	11	12	14	18	19	15	17	9	11	12	14	18	15	19	17	9	11	12	14	18	15	17	19	9	11	12	14	15	17	18	19	9	11	12	19	14	18	15	17	9	11	12	14	19	15	18	17	9	11	12	14	15	19	17	18	9	11	12	14	15	17	19	18	9	11	12	14	18	15	17	19	
9	11	12	19	14	18	15	17																																																																																																													
9	11	12	14	18	15	17	19																																																																																																													
9	11	12	14	15	17	18	19																																																																																																													
9	11	12	19	14	18	15	17																																																																																																													
9	11	12	14	19	18	15	17																																																																																																													
9	11	12	14	18	19	15	17																																																																																																													
9	11	12	14	18	15	19	17																																																																																																													
9	11	12	14	18	15	17	19																																																																																																													
9	11	12	14	15	17	18	19																																																																																																													
9	11	12	19	14	18	15	17																																																																																																													
9	11	12	14	19	15	18	17																																																																																																													
9	11	12	14	15	19	17	18																																																																																																													
9	11	12	14	15	17	19	18																																																																																																													
9	11	12	14	18	15	17	19																																																																																																													

2

Question	mp	Answer	Additional Guidance	Mark
3(a)(ii)	A3	Award one mark for: 4	If right-to-left pass completed in Q3ai, can only accept 2 swaps	1

Question	mp	Answer	Additional Guidance	Mark
3(a)(iii)	A4	Award one mark for: 3	Do not accept 2 – 3 rd pass required for algorithm to finish If right-to-left pass completed in Q3ai, can only accept 5 passes	1

Question	mp	Answer	Additional guidance	Mark
3(b)	B1	Award one mark for: Binary search	Accept any other divide and conquer search algorithm (google if necessary) Accept 'binary' on its own as search in question	2
	B2	Award one mark for any of the following: - Divide and conquer / list halves each time(1) - Fewer searches needed (1)	Do not accept points about being 'already sorted' as this is in question	

Question	mp	Answer	Additional guidance	Mark
3(c)		Award one mark for each of:		5
	C1	(while) Loop ends only when 0 entered (1)	Any method e.g. <code>while age != 0:</code>	
	C2	Avoiding divide by zero error (1)	By coding 2 nd selection box from flowchart – no output required Must come after some attempt to process age or count – award even if inside loop. ignore 'other' divide by zero errors introduced by candidate Must follow logic of flowchart e.g.: <code>if count == 0: (or count != 0)</code> ... <code>else</code> ...	
	C3	Average value rounded to whole number (1)	Must come after some attempt to process age or count – award even if inside loop. Must target calculated average (even if formula not correct) Any method to get int Accept casting to int (<i>round not in spec, but string manipulation is</i>) Accept integer division	

	C4	Correct value calculated for average and count (1)	Allow even if 0 used to initialise count in B1 (this must be only error in calculation) – set count to -1 to test	
	C5	Count and average included in a syntactically correct final print statement (1)	Award even if never runs Must include both variables (may also include string)	

Correct

```

1 # Q03c
2
3 # program variables
4 age = 1
5 count = -1
6 total = 0
7
8 # -----
9 # Write your code below this line
10
11 while age != 0:
12
13     age = int(input("Please enter age in years: "))
14
15     total = total + age
16     count = count + 1
17
18 if count == 0:
19     print("No ages input")
20 else:
21     average = round(total / count)
22     print("You entered", count, "ages; the average age is", average)

```

Typical

```

1 # Q03c
2
3 # program variables
4 age = 0
5 count = 0
6 total = 0
7
8 # -----
9 # Write your code below this line
10 age = int(input("Please enter age in years: "))
11 while age != 0:
12
13     total = total + age
14     count = count + 1
15
16     age = int(input("Please enter age in years: "))
17
18 if count == 0:
19     print("No ages input")
20 else:
21     average = int(total / count)
22     print("You entered", count, "ages; the average age is", average)

```

Question	mp	Answer	Additional Guidance	Mark
4(a)	A1	Breaking problem into small parts		1

Question	mp	Answer	Additional guidance	Mark
4(b)		Award two marks for a linked explanation such as:		
	B1 B2	- Programmer can ignore unnecessary detail (1) because it doesn't matter how library generates the number (1) - Eli can focus on just his own code (1) because library hides the code that generates the random numbers (1) - It doesn't matter how the subprogram generates the random number (1) because Eli only needs the number itself (1)	First mark for meaning of abstraction (focus on necessary parts / hiding unnecessary detail) Second mark for linking use of random library Do not accept answers relating to modelling real-world scenarios as this is in the question	2

Question	mp	Answer	Additional guidance	Mark
4(c)		Award one mark for each of:		
	C1	Booster variable initialised to 1 (1)	booster = 1	
	C2	At least one given constant used correctly (1)		
	C3	Loop used to make three throws (1)	Either type	
	C4	Use of random.randint() to assign roll a value between 1 and 6 (1)	roll = random.randint(1,SIDES) roll = random.randint(1,6) Any accurate method do not accept: random.randint(6) random.randint(SIDES)	
	C5	Check for both 3 or 6 for at least one roll (1)	any method e.g. if (roll == 3 or roll == 6) if roll % 3 == 0	
	C6	booster incremented (1)	Must be indented under condition in C5. Can award even if C5 not functioning correctly, but must be idea of only boosting on 3 and 6	
	C7	Correct final score calculated from 3 random rolls (1)	Even if not displayed Does not need to use given variable (finalScore) to get this mark	8

	C8	Correct output strings including each roll score and final score with appropriate strings (1)	All three throws and final score as shown in figure 6. <code>roll</code> and <code>finalScore</code> variables should match candidate solution Allow sensible alternate strings e.g. Roll for Rolled or Total for Score – Ignore : (colon)	
--	----	---	---	--

```
1 # Q04c
2
3 # import libraries
4 import random
5
6 # program constants
7 SIDES = 6
8 THROWS = 3
9
10 # complete the lines to initialise the variables
11 roll = 0
12 subtotal = 0
13 booster = 1
14 finalScore = 0
15
16 # roll dice 3 times
17 for throw in range(THROWS):
18     # display number rolled
19     roll = random.randint(1,SIDES)
20     print("Rolled:",roll)
21
22     # update subtotal
23     subtotal = subtotal + roll
24
25     # check for roll of 3 or 6 and increment booster if needed
26     if roll == 3 or roll == 6:
27         booster = booster + 1
28
29 # calculate and output final score
30 finalScore = subtotal * booster
31 print("Final score:", finalScore)
```

Question	mp	Answer	Additional guidance	Mark
5(a)(i)	A1 A2	<u>final</u> score not accurate (1) because score is resetting/score keeps going back to zero (in each iteration of the loop) (1)	'Score not accurate' is not enough on its own 'Score on line 20' or 'Score output to DISPLAY' would count as <u>final</u> score.	2

Question	mp	Answer	Additional guidance	Mark
5(a)(ii)	A3 A4	Award up to two marks for a linked description such as: - Asking 10 addition questions (1) and calculating final score (1) - Asking randomised adding questions (1) and checking user answer against computer answer (1) - Creating adding quiz with random numbers up to 99 (1) and outputting number of correct answers (1)	Processes that can be included Concept of this being an 'Adding Quiz' required - Getting random numbers for question - Calculating sum of two random numbers for correct answers - comparing user answer to correct answer - incrementing /tracking score - asking 10 questions (loop) - outputting final score	2

Question	mp	Answer	Additional guidance	Mark
5(b)				
	B1	Open file for writing (1)	"w" mode only Accept "w+"	8
	B2	Selection used to get grade (1)	Must address all 4 grades Does not have to be accurate on boundaries e.g. >80 is ok for distinction etc.1	
	B3	Output suitable message to say file created (1)		
	B4	Evidence of 2D indexing (1)	Even if element not used in rest of program 2D indexing or 1D index within <i>for-each</i> structure or sum(1Darray) or min(1Darray) etc. At least 1 inner value accessed	
	B5	Lowest score must be found and omitted/subtracted from calculated scores (all arrays) (1)	Any method e.g.: - List methods - Selection statements - iterable functions must be done programmatically	
	B6	At least 1 line written to file (1)	Must contain something generated by program (total/result), does not have to be correct total or grade	
	B7	Add line feed to write line after both score and grade (1)		
	B8	Output file all correct (1)		

```

1 # Q05b
2
3 Tbl_scores = [[45,42,43],[19,36,41],[48,48,48],[35,32,31],
4               [18,22,26],[41,35,31],[15,19,12],[41,39,40],
5               [27,25,22],[32,36,35],[37,36,37],[40,35,41]]
6
7 # -----
8 # Write your code below this line
9 total = 0
10 grade = ""
11 outString = ""
12
13 with open ("grades.txt","w") as theFile:
14
15     # calculate each students score
16     for student in Tbl_scores:
17         student.sort()
18         total = student[1] + student[2]
19
20     # decide student grade
21     if total >= 80:
22         grade = "Distinction"
23     elif total >= 65:
24         grade = "Merit"
25     elif total >= 50:
26         grade = "Pass"
27     else:
28         grade = "Fail"
29
30     # write score and grade to file
31     outString = str(total) + " " + grade + "\n"
32     theFile.write(outString)
33
34 print ("Grade file has been created")

```

```

1 # Q05b
2
3 Tbl_scores = [[45,42,43],[19,36,41],[48,48,48],[35,32,31],
4               [18,22,26],[41,35,31],[15,19,12],[41,39,40],
5               [27,25,22],[32,36,35],[37,36,37],[40,35,41]]
6
7 # -----
8 # Write your code below this line
9 total = 0
10 grade = ""
11 outString = ""
12
13 theFile = open ("grades.txt","w")
14
15 # calculate each students score
16 for student in Tbl_scores:
17     total = sum(student) - min(student)
18
19 # decide student grade
20 if total >= 80:
21     grade = "Distinction"
22 elif total >= 65:
23     grade = "Merit"
24 elif total >= 50:
25     grade = "Pass"
26 else:
27     grade = "Fail"
28
29 # write score and grade to file
30 outString = str(total) + " " + grade + "\n"
31 theFile.write(outString)
32
33 theFile.close()
34
35 print ("Grade file has been created")

```

```

1 # Q05b
2
3 Tbl_scores = [[45,42,43],[19,36,41],[48,48,48],[35,32,31],
4               [18,22,26],[41,35,31],[15,19,12],[41,39,40],
5               [27,25,22],[32,36,35],[37,36,37],[40,35,41]]
6
7 # -----
8 # Write your code below this line
9 grade = ""
10 outString = ""
11
12 theFile = open ("grades.txt","w")
13
14 # calculate each students score
15 for student in Tbl_scores:
16     total = 0
17     lowest = 51
18     for score in student:
19         if score < lowest:
20             lowest = score
21     total = total + score
22     total = total - lowest
23
24 # decide student grade
25 if total >= 80:
26     grade = "Distinction"
27 elif total >= 65:
28     grade = "Merit"
29 elif total >= 50:
30     grade = "Pass"
31 else:
32     grade = "Fail"
33
34 # write score and grade to file
35 outString = str(total) + " " + grade + "\n"
36 theFile.write(outString)
37
38 theFile.close()
39
40 print ("Grade file has been created")

```

Question	mp	Answer	Additional guidance	Mark
6		Award one mark for each of:		11
	A1	Both City and Temp taken as input (1)		
	A2	Both correct conversion calculation coded (1)	Does not have to be used	
	A3	Any relevant subprogram has been declared (1)	At least one relevant subprogram declared. def name(param) : def name() : and one line of code indented	
	A4	Temperature in correct parallel position is updated when existing city is input (1)		
	A5	Loop for accessing all items in either array (1)	For each acceptable for this mark	
	A6	Table includes header and dividing line (1)	<code>print("city Celsius fahrenheit")</code> <code>print("-----")</code>	
	A7	Mechanism for getting temp from string and completing conversion (1)	Include <code>int()</code> and passed to formula for at least one temperature value	
	A8	Mechanism for getting scale C/F (1)		
	A9	All rows output to table with corresponding city and (at least one) temperature values (1)	(same index value used) do not accept hard coded	
	A10	Table (for cities) all correct (1)	Ignore header (A6) and average (A11) Must have city, C value and F value (all accurate) Ignore float values if calculation correct	
	A11	Overall average correctly calculated and displayed in both scales (1)	Average may change depending on whether whole number cast or rounded – award accurate formula, not result.	

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), 2.4.1(2), 2.5.1(2), 2.6.1(2)

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	


```

1 # Q06
2
3 # city and temperature arrays
4 Tbl_city = ["Reggane", "Cairo", "New Delhi", "Muscat", "Athens",
5             "Barcelona", "Havana", "Phoenix", "Brisbane", "Darwin"]
6 Tbl_temps = ["F103", "F82", "C34", "F95", "C29", "F79", "F84",
7             "C35", "F77", "F81"]
8
9 # -----
10 # write subprogram(s) below this line
11 def CtoF(pTemp):
12     fhTemp = pTemp * 1.8 + 32
13     fhTemp = int(round(fhTemp,0))
14     return fhTemp
15
16 def FtoC(pTemp):
17     ctTemp = (pTemp - 32) / 1.8
18     ctTemp = int(round(ctTemp,0))
19     return ctTemp
20
21 # -----
22 # Write your code below this line
23 totalF = 0
24 totalC = 0
25 cTemp = 0
26 fTemp = 0
27
28 cityCount = len(Tbl_city)
29
30 # update city temp
31 thisCity = input("Please enter City: ")
32 for x in range(cityCount):
33     if thisCity == Tbl_city[x]:
34         thisTemp = input("Please enter temp in correct format: ")
35         Tbl_temps[x] = thisTemp
36     else:
37         print("City does not exist")
38

```

```

39 # display table header
40 print("City          Celsius          Fahrenheit")
41 print("-----")
42
43 # take each city and temp
44 for i in range(cityCount):
45
46     # get each scale and temp
47     nextTemp = Tbl_temps[i]
48     scale = nextTemp[0]
49     theTemp = int(nextTemp[1:])
50
51     # convert between temp scales
52     if scale == "C":
53         cTemp = theTemp
54         totalC = totalC + cTemp
55         fTemp = CtoF(theTemp)
56         totalF = totalF + fTemp
57     else:
58         fTemp = theTemp
59         totalF = totalF + fTemp
60         cTemp = FtoC(theTemp)
61         totalC = totalC + cTemp
62
63     # display each row
64     print(Tbl_city[i], "          ", cTemp, "          ", fTemp)
65
66 # calculate averages
67 averageF = int(round(totalF / cityCount,0))
68 averageC = int(round(totalC / cityCount,0))
69
70 # display table footer
71 print("-"*40)
72 print("Overall Average          ", averageC, "          ", averageF)
73

```

```

1  # Q06
2
3  # city and temperature arrays
4  Tbl_city = ["Reggane", "Cairo", "New Delhi", "Muscat", "Athens",
5             "Barcelona", "Havana", "Phoenix", "Brisbane", "Darwin"]
6  Tbl_temps = ["F103", "F82", "C34", "F95", "C29", "F79", "F84",
7             "C35", "F77", "F81"]
8
9  # -----
10 # write subprogram(s) below this line
11 def CtoF(pTemp):
12     fhTemp = pTemp * 1.8 + 32
13     fhTemp = int(round(fhTemp,0))
14     return fhTemp
15
16 def FtoC(pTemp):
17     ctTemp = (pTemp - 32) / 1.8
18     ctTemp = int(round(ctTemp,0))
19     return ctTemp
20
21 # -----
22 # Write your code below this line
23 totalF = 0
24 totalC = 0
25 cTemp = 0
26 fTemp = 0
27
28 # update city temp
29 thisCity = input("Please enter City: ")
30 if thisCity in Tbl_city:
31     thisIndex = Tbl_city.index(thisCity)
32     thisTemp = input("Please enter temp in correct format: ")
33     Tbl_temps[thisIndex] = thisTemp
34 else:
35     print("City does not exist")
36
37 # display table header
38 layout = "{:16} {:^10} {:^10}"
39 print(layout.format("City", "Celsius", "Fahrenheit"))
40 print("-"*40)
41
42 cityCount = len(Tbl_city)
43
44 # take each city and temp
45 for i in range(cityCount):
46
47     # get each scale and temp
48     nextTemp = Tbl_temps[i]
49     scale = nextTemp[0]
50     theTemp = int(nextTemp[1:])
51
52     # convert between temp scales
53     if scale == "C":
54         cTemp = theTemp
55         totalC = totalC + cTemp
56         fTemp = CtoF(theTemp)
57         totalF = totalF + fTemp
58     else:
59         fTemp = theTemp
60         totalF = totalF + fTemp
61         cTemp = FtoC(theTemp)
62         totalC = totalC + cTemp
63
64     # display each row
65     print(layout.format(Tbl_city[i], cTemp, fTemp))
66
67 # calculate averages
68 averageF = int(round(totalF / cityCount,0))
69 averageC = int(round(totalC / cityCount,0))
70
71 # display table footer
72 print("-"*40)
73 print(layout.format("Overall Average", averageC, averageF))

```

