



Mark Scheme (Results)

Summer 2021

Pearson Edexcel International GCSE  
in Computer Science (4CP0\_2B)  
Paper 02: Application of Computational  
Thinking - C#

## **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](http://www.edexcel.com) or [www.btec.co.uk](http://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](http://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](http://www.pearson.com/uk)

Summer 2021

Question Paper Log Number P66490A

Publications Code 4CP0\_2B\_2106\_MS

All the material in this publication is copyright

© Pearson Education Ltd 2021

## 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.

## Theory Mark Scheme

Question	mp	Answer	Additional Guidance	Mark
1 (a)	A1	<p><b>1. The only correct answer is B</b></p> <p><i>A is not correct because as it is an arithmetic operator</i></p> <p><i>C is not correct because as it is a relational operator</i></p> <p><i>D is not correct because as it is a relational operator</i></p>		(1)

Question	mp	Answer	Additional Guidance	Mark
1 (b)	B1 B2	<p>Award up to 2 marks for a linked description such as:</p> <ul style="list-style-type: none"> <li>• 1D represents items as a list (1), 2D as a table / 2D as rows and columns (1)</li> <li>• 1D is a row/column (1), 2D is a table / 2D has rows and columns (1)</li> <li>• Each element in 1D is a single value (1), each element in 2D is a 1D array (1)</li> <li>• 1D array can only store one type of element (1). 2D array can store multiple elements in it (1)</li> </ul>	Accept an interpretation	(2)

Question	mp	Answer	Additional Guidance	Mark																									
2 (c)	Award 1 mark for each set of test data.			(3)																									
		<table border="1"> <thead> <tr> <th></th> <th>Test data</th> <th>Expected results</th> </tr> </thead> <tbody> <tr> <td>C1</td> <td rowspan="2">           Either of           <ul style="list-style-type: none"> <li>• booksSold = 4</li> <li>• profit = 4</li> </ul> </td> <td rowspan="2">Poor performances this week</td> </tr> <tr> <td>C2</td> <td>booksSold</td> </tr> <tr> <td rowspan="2">C3</td> <td>profit</td> <td></td> <td rowspan="2">Sales and profit are good this week</td> </tr> <tr> <td>booksSold</td> <td>5</td> </tr> <tr> <td></td> <td>profit</td> <td>10</td> <td></td> </tr> <tr> <td></td> <td>booksSold</td> <td>21</td> <td rowspan="2">Sales and profit are excellent this week</td> </tr> <tr> <td></td> <td>profit</td> <td>20</td> </tr> </tbody> </table>				Test data	Expected results	C1	Either of <ul style="list-style-type: none"> <li>• booksSold = 4</li> <li>• profit = 4</li> </ul>	Poor performances this week	C2	booksSold	C3	profit		Sales and profit are good this week	booksSold	5		profit	10			booksSold	21	Sales and profit are excellent this week		profit	20
		Test data			Expected results																								
	C1	Either of <ul style="list-style-type: none"> <li>• booksSold = 4</li> <li>• profit = 4</li> </ul>			Poor performances this week																								
	C2					booksSold																							
	C3	profit				Sales and profit are good this week																							
		booksSold			5																								
	profit	10																											
	booksSold	21	Sales and profit are excellent this week																										
	profit	20																											

Question	mp	Answer	Additional Guidance	Mark
3 (b)	B1	<p>Award up to 2 marks for a linked explanation such as:</p> <ul style="list-style-type: none"> <li>The number of keys are limited/only one shift used (1) making it easy to use brute force to decrypt (1)</li> <li>It can be easy to find commonly used letters (e.g. E) (1) and guess the key (1)</li> </ul>	Accept alternative similar wording.	(2)

Question	mp	Answer	Additional Guidance	Mark																																	
3 (c)		<p>Award 1 mark each up to a maximum of 4 for:</p> <table border="1"> <tbody> <tr> <td>Encrypted letter</td> <td>f</td> <td>l</td> <td>m</td> <td>k</td> <td>t</td> <td>r</td> <td>w</td> <td>h</td> <td>e</td> <td>e</td> </tr> <tr> <td>Keyword letter</td> <td>t</td> <td>h</td> <td>i</td> <td>r</td> <td>t</td> <td>y</td> <td>t</td> <td>h</td> <td>i</td> <td>r</td> </tr> <tr> <td>Decrypted letter</td> <td>m</td> <td>e</td> <td>e</td> <td>t</td> <td>a</td> <td>t</td> <td>d</td> <td>a</td> <td>w</td> <td>n</td> </tr> </tbody> </table>	Encrypted letter	f	l	m	k	t	r	w	h	e	e	Keyword letter	t	h	i	r	t	y	t	h	i	r	Decrypted letter	m	e	e	t	a	t	d	a	w	n		(4)
	Encrypted letter	f	l	m	k	t	r	w	h	e	e																										
	Keyword letter	t	h	i	r	t	y	t	h	i	r																										
	Decrypted letter	m	e	e	t	a	t	d	a	w	n																										
	C1	<p>Award 1 mark from:</p> <ul style="list-style-type: none"> <li>flmktrwhee mapped to thirtythir (1)</li> <li>Flmktr mapped to thirty (1)</li> </ul>																																			
C2	At least one letter decrypted correctly (1)																																				
C3	At least one word decrypted correctly (1)																																				
C4	Decrypted message 'meet at dawn' (1)																																				

Question	mp	Answer	Additional Guidance	Mark
3 (d)(i)	D1	Award 1 mark for: <ul style="list-style-type: none"> <li>• cipherLetter / a single encrypted letter (1)</li> </ul>	Do not accept word/message/text	(1)
3 (d)(ii)	D2	Award 1 mark for any of: <ul style="list-style-type: none"> <li>• keywordLetter</li> <li>• plaintextLetter</li> </ul>	Ignore case	(1)
3 (d)(iii)	D2	Award 1 mark for any of: <ul style="list-style-type: none"> <li>• subprogram that is already defined</li> <li>• subprogram that is already written</li> <li>• subprogram that is already compiled</li> <li>• subprogram that can be called without having to write code for it</li> </ul>	Must be clear they are referring to subprogram provided by the language itself	(1)

Question	mp	Answer	Additional Guidance	Mark																											
4 (b)(i)	B1	Award up to 2 marks for a linked explanation: <ul style="list-style-type: none"> <li>• binary search can be quicker / is more effective than a linear search / is more effective with larger lists (1) as it does not have to examine each item in the list (1)</li> <li>• binary search halves the list each time (1) so it can be faster to find an item (1)</li> <li>• binary search requires fewer comparisons than a linear search to establish an item is not in the list (1) because the linear list would need to compare each item before establishing this (1)</li> </ul>		(2)																											
4 (b)(ii)		Correct answer <table border="1" data-bbox="392 587 1258 954"> <thead> <tr> <th>Position in list</th> <th>Product code</th> <th>Order examined</th> </tr> </thead> <tbody> <tr><td>1</td><td>ark11</td><td></td></tr> <tr><td>2</td><td>asp11</td><td></td></tr> <tr><td>3</td><td>bar13</td><td></td></tr> <tr><td>4</td><td>dri15</td><td>1</td></tr> <tr><td>5</td><td>mil19</td><td></td></tr> <tr><td>6</td><td>rib10</td><td>2</td></tr> <tr><td>7</td><td>str15</td><td>3</td></tr> <tr><td>8</td><td>tor16</td><td></td></tr> </tbody> </table> Award one mark for each correct value in order column	Position in list	Product code	Order examined	1	ark11		2	asp11		3	bar13		4	dri15	1	5	mil19		6	rib10	2	7	str15	3	8	tor16			(4)
Position in list	Product code	Order examined																													
1	ark11																														
2	asp11																														
3	bar13																														
4	dri15	1																													
5	mil19																														
6	rib10	2																													
7	str15	3																													
8	tor16																														
	B1	Start of search correct	Accept 5 and 7 for B1 and B2 (2 marks)																												
	B2	Second search item correct	Follow through if start of search incorrect																												
	B3	Third search item correct	Follow through if start of search incorrect																												
	B4	All correct																													
4 (b)(iii)	B5	Award 1 mark for: <b>3 or <math>\log_2 n + 1</math></b>		(1)																											
4 (b)(iv)	B6	Award 1 mark for any of: <ul style="list-style-type: none"> <li>• bubble sort</li> <li>• merge sort</li> </ul>	Accept any known sorting algorithm	(1)																											

## C# Code Mark Scheme

Question	mp	Answer	Additional Guidance	Mark
1 (c)	C1	Change num_twenties == to num_twenties = (1)		(3)
	C2	The <b>left over</b> variable named the same in both places (1)		
	C3	Change , to +		

Question	mp	Answer	Additional Guidance	Mark
1 (d)(i)	D1	Award 1 mark for adding an appropriate comment at the end of the line where there is relational operator: <pre>20         if (letter == vowels[vowel]) // relational operator and selection</pre>	May be on different line numbers  The comment(s) added must clearly identify the component	(1)
1 (d)(ii)	D2	Award one mark for adding an appropriate comment at the end of a line where iteration starts: <pre>16         foreach (char letter in sentence) // iteration starts 17         { 18             for (vowel = 0; vowel &lt; vowels.GetLength(0); vowel++) // iteration starts 27             for (vowel = 0; vowel &lt; vowels.GetLength(0); vowel++) // iteration starts</pre>		(1)
1 (d)(iii)	D3	Award one mark for adding an appropriate comment at the end of the line where selection starts: <pre>20         if (letter == vowels[vowel]) // relational operator and selection</pre>		(1)
1 (d)(iv)	D4	Award one mark for adding an appropriate comment at the end of a line where a data structure is initialised: <pre>9         char[] vowels = {'a','e', 'i', 'o','u' }; // data structure initialised 10        int[] numVowels = { 0, 0, 0, 0, 0 }; // data structure initialised</pre>		(1)



Question	mp	Answer	Additional Guidance	Mark
2 (a)		Award one mark for each of:	<p>Logic of algorithm must be followed as set out.</p> <p>Alternatives must address each point.</p> <p>Do not penalise candidates who attempt more than the stated requirements.</p> <p>Do not penalise spelling mistakes in the assignment of username and/or password</p> <p>Do not penalise spelling mistakes and alternative wording of the output.</p>	(11)
	A1	At least one variable with a suitable variable name		
	A2	username = bard423		
	A3	password = nX2934? OR nX2934		
	A4	Loop used		
	A5	Username or password entered		
	A6	Username or password stored in variable(s)		
	A7	At least one suitable input message		
	A8	Checks username and password		
	A9	Appropriate error message(s) displays		
	A10	Welcome message displayed		
A11	Executing and producing correct output			
Code examples				
C#		<pre> // Initialise variables string username = "bard423"; string password = "nX2934?"; int count = 0; string inputUsername = ""; string inputPassword = "";  // Print prompts, take and check user input while (inputUsername != username   inputPassword != password) {     if (count &gt; 0)     {         Console.WriteLine("There is a problem with the login details. Try again");     }     count++;     Console.WriteLine("Enter your username");     inputUsername = Console.ReadLine();     Console.WriteLine("Enter your password");     inputPassword = Console.ReadLine(); } Console.WriteLine("Welcome"); Console.ReadKey(); </pre>		

Question	mp	Answer	Additional Guidance	Mark	
2 (b)		Award 1 mark for each correct condition.	Alternative alternatives e.g. Line 11 booksSold <=4 etc.		
		<b>Condition</b>			<b>Output message</b>
	B1	Number of books sold is under 5 or profit made is under 5			Poor performance this week
	B2	Number of books sold is over 20; profit made is at least 20			Sales and profit are excellent this week
	B3	Number of books sold is at least 5; profit made is at least 10			Sales and profit are good this week
	B4	All other inputs	Alert manager	(4)	

## Code examples

C#	<pre> if(booksSold &lt; 5    profit &lt; 5) {     Console.WriteLine("Poor performance this week"); } else if(booksSold &gt; 20 &amp;&amp; profit &gt;= 20) {     Console.WriteLine("Sales and profit are excellent this week"); } else if(booksSold &gt;=5 &amp;&amp; profit &gt;=10) {     Console.WriteLine("Sales and profit are good this week"); } else {     Console.WriteLine("Alert manager"); } </pre>
----	---

Question	mp	Answer	Additional Guidance	Mark
3 (a)	A1	Get plaintext and store in plaintext variable	Accept alternative wording Line numbers may be different compared to the examples shown  When testing the completed code use lowercase for the input	(1)
	A2	Get key and store in key variable		(1)
	A3	Validate key		(1)
	A4	Open file to write		(1)
	A5	Write cipher text		(1)
	A6	Close file		(1)
	A7	Displays ciphertext		(1)
	A7	Executing and producing correct output to file and screen (must have A3)		(1)

## Code examples

C#

```

18
19 // Add your code to get the plaintext and convert it to lowercase
20 Console.Write("Enter the plaintext using lowercase letters: ");
21 plaintext = Console.ReadLine().ToLower();
22

```

```

22
23 // Add your code to get the key and make sure the key is between 1 and 25
24 while (key < 1 || key > 25)
25 {
26     Console.Write("Enter the key - a number between 1 and 25 ");
27     key = Convert.ToInt32(Console.ReadLine());
28 }

```

```

56 // Add your code to write the ciphertext to a text file
57 System.IO.StreamWriter writer = new System.IO.StreamWriter("Cipher.txt");
58 writer.WriteLine(ciphertext);
59 writer.Close();

```

```

61 // Add your code to display the ciphertext
62 Console.WriteLine("Cipher text is " + ciphertext);

```

Question	mp	Answer	Additional Guidance	Mark
4 (a)	A1	At least 1 variable has a meaningful name	Ignore spelling mistakes in input message	(7)
	A2	Product name requested using a suitable input message		
	A3	Random number generated that would be at least 10 <b>or</b> no higher than 30		
	A4	Random number generated that would be in the correct range 10 to 30		
	A5	First 3 letters of product name generated		
	A6	First 3 letters of product name and random number concatenated to generate productCode		
	A7	productCode and productName output in the same print statement		

## Code examples

C#

```

9      // Get input
10     Console.WriteLine("Enter the product name:");
11     string productName = Console.ReadLine();
12
13     // Generate a random number between 10 and 30 inclusive
14     Random rand = new Random();
15     int randomNum = rand.Next(10,31);
16
17     // Generate the product code - first three letters of product name and the
18     string productCode = productName.Substring(0, 3) + randomNum;
19
20     // Display the product code and the product name
21     Console.WriteLine(productCode + " " + productName);
22     Console.ReadKey();

```

For Q5, the first 11 marks are for coding that matches requirements of task. The remaining 9 marks should be allocated on a best fit.

Question	mp	Answer	Additional Guidance	Mark
5	<b>addPlayerName()</b>			
	A1	Suitable prompt for player name and assigned to suitable variable		
	<b>guessCapital()</b>			
	A2	Ensure question can only be used once	Do not award if more than one question variable e.g. question1, question2 etc.	
	A3	Question includes suitable message and country name		
	A4	Check made to see if guess is correct		
	A5	If guess correct score incremented		
	A6	If guess is incorrect suitable message displayed		
	A7	If guess incorrect capital concatenated with message		
	A8	Repeated for a minimum of five questions	Do not award if questions are asked manually e.g. question1, question2, repeated code for each question etc.  Do not award if 5 unique questions are not asked while the program is running	
	<b>Main Program</b>			
A9	Player name or score displayed	Do not award if the return value from at least one function is not used		
A10	At least one menuChoice calls correct subprogram			
A11	Main program calls the two sub-programs correctly			
				(11)

<b>Band 1 (1-3 marks)</b>	<b>Band 2 (4-6 marks)</b>	<b>Band 3 (7-9 marks)</b>	<b>Mark</b>
Little attempt to decompose into component parts	Some attempt to decompose into component parts	The problem has been decomposed into component parts	
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	

## Code examples

C# **Add player name function**

```
string player = "";  
  
while (player == "")  
{  
    Console.Write("Enter your player name: ");  
    player = Console.ReadLine();  
    Console.WriteLine(player);  
}  
return player;
```

**Main program**

```
if (menuChoice == 1)  
{  
    playerName = AddPlayerName();  
}  
else if (menuChoice == 2)  
{  
    score = GuessCapital();  
}  
else  
{  
    Console.WriteLine("Well done " + playerName + ". The score is " + score);  
}  
}  
}  
}
```

## Guess capital city function

```
// Add your code here
int questionCount = 1;
int questionScore = 0;

// Ask 5 questions
while (questionCount <= 5)
{
    int questionChoice = -1;
    string questionNumbers = "";

    // Build a string containing the question numbers available
    foreach (int question in questions)
    {
        if (question != 0)
        {
            questionNumbers += question.ToString() + " ";
        }
    }

    // Ensure valid question number is chosen
    while (!questionNumbers.Contains(questionChoice.ToString()))
    {
        Console.Write("Pick a number from " + questionNumbers);
        questionChoice = Int32.Parse(Console.ReadLine());
    }
}
```

```
// Get the country and its capital
string country = countries[questionChoice - 1];
string capital = capitals[questionChoice - 1];

// Display the country and get the guess
Console.Write("What is the capital of " + country + "? ");
string guess = Console.ReadLine().ToLower();

// If the guess is correct display message and increment score
if (guess == capital.ToLower())
{
    Console.WriteLine("Well done you guessed correctly");
    questionScore ++;
}
else
// Otherwise display the country name and correct capital
{
    Console.WriteLine("You did not guess correctly. The capital of " + country + " is " + capital);
}

// Increment the number of questions asked
questionCount++;

// Set the question number to 0 so that it cannot be guessed again
questions[questionChoice - 1] = 0;
}

// Return the score to the main menu
return questionScore;
```





