Pearson Edexcel

Mark Scheme (Results)

November 2021

Pearson Edexcel International GCSE
In Computer Science (4CP0/2A)
Paper 02: 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

November 2021
Question Paper Log Number P69298A
Publications Code 4CP0_2A_2111_MS

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

Question	mp	Answer	Additional Guidance	Mark
1 (a)	A1	**Award 1 mark for any of:** • Comments (1) • Indentation (1) • Meaningful variable/constant/subprogram names (1) • White space (1)		(1)

Question	mp	Answer	Additional Guidance	Mark
1 (b)	B1 B2 B3	**Award** 1 mark for each column:		(3)

Error description	Logic	Syntax	Runtime
Divide by 0			✓
Use x instead of * multiply		✓	
Subtract 10 from 2 instead of 2 from 10	✓		

Question	mp	Answer	Additional Guidance	Mark
1 (c)	C1	**The only correct answer is D** *A is not correct because as it is a data structure* *B is not correct because as there is no arrow looping back to the condition* *C is not correct because as the flowchart does not include an operator*		(1)

<table>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>2 (b)</td><td>B1
B2
B3
B4</td><td>Award up to 4 marks for a linked explanation such as:

• The list is not sorted (1) Juan should come after Elija if the list was sorted / Elija is compared with Juan (1), so the bottom half of the list would be discarded (1) after the first pass through the loop (1)
• Binary search works on a sorted list (1) because it uses divide and conquer (1) half the list discarded each pass through (1) based on comparison of search item with middle item (1)</td><td></td><td>(4)</td></tr>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>2 (c)</td><td>C1
C2
C3
C4
C5</td><td>Award 1 mark for each of:
Split into two sets of four (1)
Split each set into two pairs (1)
Split each pair into single elements (1)
Merge elements into sorted pairs (1)
Merge pairs into sorted sets (1)

<table><tr><td>9 1 7 6 3 5 2 8</td></tr></table>
<table><tr><td>9, 1, 7, 6</td><td>3, 5, 2, 8</td></tr></table>
<table><tr><td>9, 1</td><td>7, 6</td><td>3, 5</td><td>2, 8</td></tr></table>
<table><tr><td>9</td><td>1</td><td>7</td><td>6</td><td>3</td><td>5</td><td>2</td><td>8</td></tr></table>
<table><tr><td>1, 9</td><td>6, 7</td><td>3, 5</td><td>2, 8</td></tr></table>
<table><tr><td>1 6, 7, 9</td><td>2, 3, 5, 8</td></tr></table>
<table><tr><td>1 2 3 5 6 7 8 9</td></tr></table>
</td><td>Allow step at C3 to be implied</td><td>(5)</td></tr>
</table>

<table>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>3 (a) (i)</td><td>A1</td><td>C# and Python monthNumber/monthsAndDays/count/found (1)
Java monthNumber/monthsAndDays/count/found/input</td><td></td><td>(1)</td></tr>
<tr><td>3 (a) (ii)</td><td>A2</td><td>monthsAndDays (1)</td><td></td><td>(1)</td></tr>
<tr><td>3 (a) (iii)</td><td>A3</td><td>C#
WriteLine, ToString, ToInt32, ReadLine
Java
Print, ParseInt, nextLine, ToString, Equals
Python
print/len/int/input (1)</td><td></td><td>(1)</td></tr>
<tr><td>3 (a) (iv)</td><td>A4</td><td>Display/showMonthNameAndDays (1)</td><td></td><td>(1)</td></tr>
<tr><td>3 (a) (v)</td><td>A5</td><td>pMonth (1)</td><td></td><td>(1)</td></tr>
<tr><td>3 (a) (vi)</td><td>A6
A7</td><td>Award 1 mark for each of:<table><tr><th>Type of test data</th><th>Test data</th></tr><tr><td>Boundary (1)</td><td>1</td></tr><tr><td>Erroneous (1)</td><td>15</td></tr></table></td><td></td><td>(2)</td></tr>
</table>

<table>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>3 (b)</td><td>B1
B2</td><td>Award up to 2 marks for a linked explanation such as:
A library program would not know the names of the calling program’s variables in advance (1) meaning it would not be a reusable solution (1)
(Using parameters) enables reusable solutions (1) using different data values/arguments (1)
Using parameters uses less memory / uses memory more efficiently (1) because it avoids the use of global variables / because the memory is freed after the subprogram is finished (1)</td><td></td><td>(2)</td></tr>
</table>

Question	mp	Answer	Additional Guidance	Mark
4 (b)	B1 B2	**Award up to 2 marks for a linked description such as:** • To check to see if a word is the same when it is reversed / is a palindrome (1) and output an appropriate message / repaper is the same when reversed (1)		(2)

Coding

Question	mp	Answer	Additional Guidance	Mark
1 (d)	**Award 1 mark for each of:**			
	D1	= changed to == (1)		
	D2	Condition changed to less than / final two print statements swapped (1)		
	D3	else changed to else (1)		(3)

Code examples

C#

```
if (number1 == number2)
{
    Console.WriteLine("The numbers are equal");
}
else if (number1 < number2)
{
    Console.WriteLine("The highest number is " + number2 + " and the lowest number is " + number1);
}
else
{
```

Java

```
if (number1 == number2)
{
    System.out.print("Numbers are equal");
}
else if (number1 < number2)
{
   System.out.print("The highest number is " + number2 + " and the lowest number is " +number1);
}
else
```

Python

```
if number1 == number2:
    print("Numbers are equal")
elif number1 < number2:
    print("The highest number is", number2, "and the lowest number is", number1)
else:
```

Question	mp	Answer	Additional Guidance	Mark
1 (e)	**Award 3 marks for:**			
	E1	Prompt for a number to be input (1)		
	E2	Input stored in an integer variable (1)		
	E3	Loop used (1)		
	E4	Loop iterates only for the values 1 – 12 (1)		
	E5	Display is correct when program executes (1)		(5)

Code examples

C#

```
// Print prompt and get number from user
Console.Write("Enter the number: ");
int num = Convert.ToInt32(Console.ReadLine());

// Create loop to display the table
for (int count = 1; count <= 12; count++)
{
    Console.WriteLine(count + " x " + num + " is " + count * num);
}
Console.ReadLine();
```

Java

```
// Print prompt and get number from user
Scanner input = new Scanner(System.in);
System.out.print("Enter the number ");
int num = input.nextInt();

// Create loop to display the table
for (int count = 1; count <=12; count++)
{
    System.out.println(count + " x " + num + " is " + count * num);
}
```

Python	`# Print prompt and get number from user` `num = int(input("Enter the number: "))` `# Create loop to display the table` `for count in range(1, 13):` `    print(count, "x" , num , "is" , count * num)`

Question	mp		Additional Guidance	Mark
2 (a)	**Award one mark for each of:**		Logic of algorithm must be followed as set out. Alternatives must address each point. Do not penalise candidates who attempt more than the stated requirements. Do not penalise spelling mistakes in the input message.	
	A1	Input as an integer (1)		
	A2	Loop with correct conditions start <= end and not found (1)		
	A3	Use of correct symbol for integer division (in conversion of DIV) or type coercion (1)		
	A4	Check to see if number in middle matches item (1)		
	A5	Set found to true (1)		
	A6	Check to see if item is less than number in middle (1)		
	A7	Set end to middle – 1 (1)		
	A8	else – set start to middle + 1 (1)		
	A9	count incremented by 1 (1)		
	A10	Program execution is fully correct (1)		(10)
Code examples				

C#	<pre>Console.WriteLine("What is the number to find? "); int item = Convert.ToInt32(Console.ReadLine()); while (start <= end && !found) { middle = (start + end) / 2; if (numberList[middle] == item) { found = true; } else { if (item < numberList[middle]) { end = middle - 1; } else { start = middle + 1; } } count++; }</pre>

Java

```java
Scanner input = new Scanner(System.in);
System.out.print("What is the number to find? ");
int item = Integer.parseInt(input.nextLine());

while (start <= end && !found)
{
    middle = (start + end) / 2;
    if (numberList[middle] == item)
    {
        found = true;
    }
    else
    {
      if (item < numberList[middle])
      {
          end = middle - 1;
      }
      else
      {
          start = middle + 1;
      }
    }
    count ++;
}
```

Python

```python
item = int(input("What is the number to find? "))
while start <= end and not found:
    middle = (start + end) // 2
    if numberList[middle] == item:
        found = True
    else:
        if item < numberList[middle]:
            end = middle - 1
        else:
            start = middle + 1
    count = count + 1
```

Question	mp	Answer	Additional Guidance	Mark
3 (c)	**Award 1 mark for each of:**			
	C1	Condition to check whether the year exactly divides by 4 or equivalent condition (1)		
	C2	Condition to check whether year exactly divides by 400 or equivalent condition (1)		
	C3	Condition to check whether year does not exactly divide by 100 or equivalent condition (1)		
	C4	Correct message for at least one of the conditions (1)		
	C5	Program executes and produces the correct output (1)		
	C6	Efficient solution (1)		
				(6)

Code examples

C#

```
if (year % 400 == 0)
 {
     Console.WriteLine(year + " is a leap year");
 }
else if (year % 4 == 0 && year % 100 !=0)
 {
     Console.WriteLine(year + " is a leap year");
 }
 else
 {
     Console.WriteLine(year + " is not a leap year");
 }
// End of main program
```

Java

```
if (year % 400 == 0)
{
    System.out.print(year + " is a leap year");
}
else if (year % 4 == 0 && year % 100 !=0)
{
    System.out.print(year + " is a leap year");
}
else
{
    System.out.print(year + " is not a leap year");
}
```

Python	<pre>if year % 400 == 0: print(year,"is a leap year") elif year % 4 == 0 and year % 100 !=0: print(year,"is a lear year") else: print(year,"is not a leap year")</pre>

Question	mp	Answer	Additional Guidance	Mark
4 (a)	**Award one mark for each of:**			
		Validation		
	A1	Suitable input prompt and storing it in binaryPattern (1)		
	A2	Length of binary pattern validated as 8 (1)		
	A3	Loop used to check the length of binary pattern (1)		
		Binary to denary		
	A4	Loop through each digit in the binary pattern (1)		
	A5	Correct conversion of at least one binary digit to relevant placeholder (1)		
	A6	Attempt at running total using denaryNumber (1)		
	A7	Correct running total using denaryNumber (1)		
	A8	Print statement includes the binary pattern and the denary number (1)		
	A9	Print statement outside of conversion loop (1)		
	A10	Comment explains how conversion works (1)		
	A11	Program executes correctly for any binary pattern that is exactly 8 characters long only (1)		(11)
Code examples				

C#	``` while (binaryPattern.Length != 8) { Console.WriteLine("Enter an 8 digit binary number to convert to denary "); binaryPattern = Console.ReadLine(); } foreach (char digit in binaryPattern) { if (digit == '1') { denaryNumber += denaryPlaceholders[count] * 1; } count++; } Console.WriteLine(binaryPattern + " converted to denary is " + denaryNumber); Console.ReadLine(); ```
Java	``` while (binaryPattern.length() != 8) { Scanner input = new Scanner(System.in); System.out.print("Enter an 8 digit binary number to convert to denary "); binaryPattern = input.next(); } for (char digit : binaryPattern.toCharArray()) { if (digit == '1') { denaryNumber += denaryPlaceholders[count] * 1; } count ++; } System.out.print(binaryPattern + " converted to denary is " + denaryNumber); ```

Python	<pre>while len(binaryPattern) != 8: binaryPattern = input("Enter an 8 digit binary number to convert to denary: ") for digit in binaryPattern: if digit == "1": denaryNumber = denaryNumber + denaryPlaceholders[count] * 1 count = count + 1 print(binaryPattern,"converted to denary is", denaryNumber)</pre>

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	A1	At least three variables initialised appropriately		(1)
	A2	Word input		(1)
	A3	Program repeats until 1 is input		(1)
	A4	Loop through each word in the array		(1)
		Words that begin with the same letter		
	A5	Check to see if the first letter of the input word matches the first letter of at least one word in the array		(1)
	A6	The number of words that begin with the same letter displayed		(1)
		Words that contain the input word		
	A7	Check to see if the word contains the input word		(1)
	A8	Number of letters in the word calculated		(1)
	A9	At least one word from the array that contains the input word displayed		(1)
	A10	The number of words that contain the input word displayed		(1)
	A11	Identification of the longest word or the shortest word		(1)

Band 1 (1-3 marks)	Band 2 (4-6 marks)	Band 3 (7-9 marks)	Mark
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	(9)

Code examples

C#

```
// Add your code here
string inputWord = "";

while (inputWord != "1")
{
    int shortest = 5000;
    int longest = 0;
    String shortestWord = "";
    String longestWord = "";

    Console.WriteLine("Enter a word or 1 to exit ");
    inputWord = Console.ReadLine();

    if (inputWord != "1")
    {
        int count = 0;

        foreach (String word in wordArray)
        {
            if(word[0] == inputWord[0])
            {
                Console.WriteLine(word);
                count ++;
            }
        } // End of Loop

        Console.WriteLine(count + " word(s) beginning with "+ inputWord[0]);
        Console.WriteLine("----------------------------------------------------------------------");

        count = 0;

        foreach (String word in wordArray)
        {
            if(word.Contains(inputWord))
            {
                count ++;
                Console.WriteLine(word);

                int length = word.Length;

                if (longest < length)
```

```
                        {
                            longest = length;
                            longestWord = word;
                        }

                        if (shortest > length)
                        {
                            shortest = length;
                            shortestWord = word;
                        }
                    }// end of checking word
                } // End of for loop

                // Print the number of words, the number of characters in the longest and
                // shortest word, the longest and the shortest word

                if (count > 0)
                {
                    Console.WriteLine(count + " word(s) with "+ inputWord + " in them");
                    Console.WriteLine("The longest word has " + longest + " characters");
                    Console.WriteLine("The shortest word has " + shortest + " characters");
                    Console.WriteLine("The longest word is " + longestWord + " characters");
                    Console.WriteLine("The shortest word is " + shortestWord + " characters");
                    Console.WriteLine("------------------------------------------------------------------------------");
                }
                else
                {
                    Console.WriteLine("There were 0 words that had all the letters from " + inputWord + " in them");
                    Console.WriteLine("------------------------------------------------------------------------------");
                } // End of printing words
            } // End of check
        } // End of while loop
```

Java

```
        // Add your code here
        Scanner input = new Scanner(System.in);
        String inputWord = "";

        // While there is a word run the program
        while (!inputWord.equals("1")) {

            int shortest = 5000;
            int longest = 0;
            String shortestWord = "";
            String longestWord = "";

            System.out.println("Enter a word or 1 to exit ");
            System.out.println("-----------------------------------------------------------------------------");
            inputWord = input.nextLine();
            if (!inputWord.equals("1")){
                // Find words that begin with the same letter as the input word
                int count = 0;

                // Get each word in the array
                for (String word: wordArray){

                    // If the first character of the word is the same as the first character of
                    // the input word then print the word
                    if (word.charAt(0) == inputWord.charAt(0)){
                        System.out.println(word);
                        count ++;
                    }
                } // End of for loop

                // Print the total number of words that begin with the same letter
                System.out.println(count + " word(s) beginning with "+ inputWord.charAt(0));
                System.out.println("-----------------------------------------------------------------------------");

                // Find words that contain the input word
                count = 0;

                // Get each word in the array
                for (String word: wordArray){
                    // Check to see if the word contains the inputWord
                    if (word.contains(inputWord)){
                        count ++;
                        System.out.println(word);

                        // Set the length of the word
                        int length = word.length();
```

```java
                // If the length is longer than the current longest replace it and the longestWord
                if (longest < length) {
                    longest = length;
                    longestWord = word;
                }

                // If the length is shorter than the shortest replace it and the shortestWord
                if (shortest > length){
                    shortest = length;
                    shortestWord = word;
                }
            } // End of checking word
        } // End of for loop

        // Print the number of words, the number of characters in the longest and
        // shortest word, the longest and the shortest word
        if( count > 0){
            System.out.println(count + " word(s) with "+ inputWord + " in them");
            System.out.println("The longest word has " + longest + " characters");
            System.out.println("The shortest word has " + shortest + " characters");
            System.out.println("The longest word is " + longestWord + " characters");
            System.out.println("The shortest word is " + shortestWord + " characters");
            System.out.println("------------------------------------------------------------------------------");
        } else {
            System.out.println("There were 0 words that had all the letters from " + inputWord + " in them");
            System.out.println("------------------------------------------------------------------------------");
        } // End of printing words
    }
} // End of while loop
```

Python

```python
# While there is a word run the program
while inputWord !="1":

    shortest = 5000
    longest = 0
    shortestWord = ""
    longestWord = ""

    inputWord = input("Enter a word or 1 to exit: ")
    print("----------------------------------------------------------------------")

    if inputWord !="1":
        # Find words that begin with the same letter as the input word
        count = 0
        # Get each word in the array
        for word in wordArray:
            # If the first character of the word is the same as the first character of
            # the input word then print the word
            if word[0] == inputWord[0]:
                print(word)
                count = count + 1

        # Print the total number of words that begin with the same letter
        print("\n",count,"word(s) beginning with",inputWord[0])
        print("----------------------------------------------------------------------")

        # Find words that contain the input word
        count = 0
        # Get each word in the array
        for word in wordArray:
            # Check to see if the word contains the inputWord
            if inputWord in word:
                count = count + 1
                print(word)
                # Set the length of the word
                length = len(word)

                # If the length is longer than the current longest replace it and the longestWord
                if longest < length:
                    longest = length
                    longestWord = word

                # If the length is shorter than the shortest replace it and the shortestWord
                if shortest > length:
                    shortest = length
                    shortestWord = word
```

```python

        # Print the number of words, the number of characters in the longest and
        # shortest word, the longest and the shortest word
        if count > 0:
            print("\n",count,"word(s) with",inputWord,"in them")
            print("The longest word has",longest,"letters")
            print("The shortest word has",shortest,"letters")
            print("The longest word is",longestWord)
            print("The shortest word is",shortestWord)
            print("-----------------------------------------------------------------------")
        else:
            print("There were 0 words with all the letters from",inputWord,"in them")
        # ------------------------------------------------------------------------------------------

# End the program when there are no more words
print("End of program")
```

Pearson Education Limited. Registered company number 872828
with its registered office at 80 Strand, London, WC2R 0RL, United Kingdom