Mark Scheme

Summer 2024

Pearson Edexcel International GCSE

In Computer Science (4CP0/02)

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

Summer 2024
Question Paper Log Number P75737A
Publications Code 4CP0_02_2406_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.

Question	mp	Answer	Additional Guidance	Mark
1(a)	A1	**The only correct answer is D** ***A*** *is not correct because it is the definition of decomposition* ***B*** *is not correct because it is a task undertaken during evaluation* ***C*** *is not correct because it describes a method for expressing algorithms*		**1**

Question	mp	Answer	Additional guidance	Mark
1b		Award **one** mark for each of:		
	B1	Initialisation of myName to "" (1)	• `String myName = "";` • `String myName = "";` • `myName = ""` Allow null Allow single space between quotes " "	
	B2	Add a line to create myAge and initialise it to 0 (1)	• `int myAge = 0;` • `int myAge = 0;` • `myAge = 0` Allow any method	
	B3	Complete the test by inserting < (1)	• `if (myAge < 30)` • `if (myAge < 30)` • `if (myAge < 30):` Allow <= Can award reverse test only if welcome message contained within an 'else' part	
	B4	Add a line to display "Welcome" (1)	• `Console.WriteLine ("Welcome");` • `System.out.println ("Welcome");` • `print ("Welcome")` ignore indentation Allow any method	**4**

C#

```
1   // Q01bFINISHED
2
3   // Complete the line to initialise myName to an empty string
4   String myName = "";
5
6   // Add a line to create an integer variable called myAge and
7   //     initialise myAge to 0
8   int myAge = 0;
9
10  Console.WriteLine ("Enter your name: ");
11  myName = Console.ReadLine();
12  Console.WriteLine ("Enter your age: ");
13  myAge = Convert.ToInt32 (Console.ReadLine());
14
15  // Complete the test to check if the age is less than 30
16  if (myAge < 30)
17  {
18      // Add a line to display a message that says "Welcome"
19      Console.WriteLine ("Welcome");
20  }
21
```

Java

```java
// Q01bFINISHED

import java.util.Scanner;
public class Q01bFINISHED
{
    public static void main (String[] args)
    {
        Scanner myKeyboard = new Scanner (System.in);

        // Complete the line to initialise myName to an empty string
        String myName = "";

        // Add a line to create an integer variable called myAge and
        //     initialise myAge to 0
        int myAge = 0;

        System.out.println ("Enter your name: ");
        myName = myKeyboard.nextLine();
        System.out.println ("Enter your age: ");
        myAge = Integer.parseInt (myKeyboard.nextLine());

        // Complete the test to check if the age is less than 30
        if (myAge < 30)
        {
            // Add a line to display a message that says "Welcome"
            System.out.println ("Welcome");
        }
    }
}
```

Python

```python
# Q01bFINISHED

# Complete the line to initialise myName to an empty string
myName = ""

# Add a line to create an integer variable called myAge and
#     initialise myAge to 0
myAge = 0

myName = input ("Enter your name: ")
myAge = int (input ("Enter your age: "))

# Complete the test to check if the age is less than 30
if (myAge < 30):

    # Add a line to display a message that says "Welcome"
    print ("Welcome")

```

Question	mp	Answer	Additional guidance	Mark
1c(i)		Award **one** mark for any of the following:		
	C1	C#: • // Q01c (1) • Q01c (1) • // Loop for 50 times (1) • Loop for 50 times (1) Java: • // Q01c (1) • Q01c (1) • // Loop for 50 times (1) • Loop for 50 times (1) Python: • # Q01c (1) • Q01c (1) • # Loop for 50 times (1) Loop for 50 times (1)	Ignore transcription errors	**1**

Question	mp	Answer	Additional guidance	Mark
1c(ii)		Award **one** mark for any of the following:		
	C2	C#: • int count = 0; (1) • int total = 0; (1) • int subtotal = 0; (1) Java: • int count = 0; (1) • int total = 0; (1) • int subtotal = 0; (1) Python: • count = 0 (1) • total = 0 (1) • subtotal = 0 (1)	Ignore transcription errors	**1**
Question	**mp**	**Answer**	**Additional guidance**	**Mark**
1c(iii)		Award **one** mark for:		
	C3	• if (1)	Applies to all programming languages Ignore capitalisation	**1**

Question	mp	Answer	Additional guidance	Mark
1c(iv)		Award **one** mark for:		
	C4	C#: • && (1) Java: • && (1) Python: • and (1)	Ignore capitalisation	**1**

Question	mp	Answer	Additional Guidance	Mark
1(d)	D1	**The only correct answer is B** ***A*** *is not correct because the data can be stored in an array as it is all the same data type (string)* ***C*** *is not correct because the data can be stored in an array as it is all the same data type (integer)* ***D*** *is not correct because the data can be stored in an array as it is all the same data type (real)*		**1**

<table>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(e)</td><td>E1
E2</td><td>One mark for each correct cell:

<table>
<tr><th>Item</th><th>Data type</th></tr>
<tr><td>45.82</td><td>All programming languages:
real/float
C#:
decimal/double
Java:
Double/double/BigDecimal
Python:
decimal</td></tr>
<tr><td>True</td><td>All programming languages:
Boolean/bool</td></tr>
</table>
</td><td>Ignore capitalisation

Do not accept string for True</td><td>2</td></tr>
</table>

Question	mp	Answer	Additional Guidance	Mark
2(a)(i)	A1	**The only correct answer is A** ***B*** *is not correct because* ***computational thinking*** *are skills used in creating algorithms and programs* ***C*** *is not correct because* ***decomposition*** *is breaking down a problem into smaller parts* ***D*** *is not correct because* ***pseudocode*** *is a notation for expressing an algorithm*		**1**

Question	mp	Answer	Additional Guidance	Mark
2(a)(ii)	A2	Award **one** mark for: • Variable (1)	Allow parameter Ignore local/global	**1**

Question	mp	Answer	Additional Guidance	Mark
2(b)(i)	B1	Award **one** mark for any of the following: • An instruction is impossible to execute (1) • The program stops executing **unexpectedly** / **crashes** (1)	Do not award an example, as that is the next question Don't award repeat of question e.g. "Stopping during execution" without clarifying reason or that stop was unexpected	**1**

Question	mp	Answer	Additional Guidance	Mark
2(b)(ii)	B2	Award **one** mark for any of the following: • Division by zero (1) • Accessing memory outside the range of an array (1) • Attempting an operation on incompatible types (1) • File not found (1) • An exception is raised that cannot be/is not handled (1)	Award other examples, if they can be mapped to a bullet	**1**

Question	mp	Answer	Additional guidance	Mark
2(c)		Award **one** mark for each of:		
	C1	Add or remove a semicolon on initialisation of the length variable (1)	• `int length = 4;` • `int length = 4;` • `length = 4`	**3**
	C2	Add a comma between the arguments to the subprogram call to generate a random number (1)	• `width = randGenerator.Next (1, 6);` • `width = randGenerator.nextInt(1, 6);` • `width = random.randint (1, 5)`	
	C3	Replace modulus operator (%) with multiplication operator (*) (1)	• `perimeter = (2 * width) + (2 * length);` • `perimeter = (2 * width) + (2 * length);` • `perimeter = (2 * width) + (2 * length)`	

C#

```
// Q02cFINISHED

int width = 0;
int length = 4;
int perimeter = 0;
Random randGenerator = new Random();

width = randGenerator.Next (1, 6);
perimeter = (2 * width) + (2 * length);

Console.WriteLine (width.ToString() + " " +
                   length.ToString() + " " +
                   perimeter.ToString());
```

Java

```java
// Q02cFINISHED

import java.util.Random;

public class Q02cFINISHED
{
    public static void main (String[] args)
    {
        int width = 0;
        int length = 4;
        int perimeter = 0;
        Random randGenerator = new Random();

        width = randGenerator.nextInt(1, 6);
        perimeter = (2 * width) + (2 * length);

        System.out.println (String.valueOf(width) + " " +
                            String.valueOf(length) + " " +
                            String.valueOf(perimeter));
    }
}
```

Python

```python
# Q02cFINISHED

import random

width = 0
length = 4
perimeter = 0

width = random.randint (1, 5)
perimeter = (2 * width) + (2 * length)

print (width, length, perimeter)

```

<table>
<tr><th>Question</th><th>mp</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr>
<td>2(d)</td>
<td>D1
D2
D3
D4</td>
<td>
<table>
<tr><td>Erroneous</td><td>Award one mark for any of the following up to a maximum of two marks:
Contains incorrect data type (1)
<2 letters (1)
9+ letters (1)</td></tr>
<tr><td>Normal</td><td>A string between 2 and 8 in length that includes only letters (1)</td></tr>
<tr><td>Boundary</td><td>Award one mark for any of the following:
A string of exactly 8 letters (1)
A string of exactly 2 letters (1)</td></tr>
</table>
</td>
<td>Award any correct example (not description)

Do not award an empty cell as equivalent to empty string

Do not accept same test twice for erroneous</td>
<td>4</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)</td>
<td>A1
A2
A3
A4</td>
<td>Award one mark for correct column (A OR B)
Award one mark for correct column (NOT B)
Award one mark for first set of two cells in third column
Award one mark for second set of two cells in third column

<table>
<tr><th>A</th><th>B</th><th>A OR B</th><th>NOT B</th><th>(A OR B) AND (NOT B)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>
</td>
<td>Award follow through on last column only

5th column can be judged

On columns 1 and 2 for MP3 and MP4
or
as FT from columns 3 and 4 for MP3 and MP4
award highest mark

</td>
<td>4</td>
</tr>
</table>

Question	mp	Answer	Additional guidance	Mark
3(b)		Award **one** mark for each of:		
	B1	Correct output taken from array (1)	Must be from array i.e. Indexing used	
	B2	Use of iteration to loop over every item in the array (1)	Award foreach loops Accept any array	
	B3	Use of appropriate method to step accurately, i.e. every other item starting at position 0 (1)	Any appropriate step method Not hard coded index	**3**

C#

```
// Q03bFINISHED

String[] starNames = {"Alasia", "Beid", "Castor", "Denebola",
                      "Electra", "Fafnir", "Gudja", "Haedus",
                        "Izar", "Jishui", "Kang", "Lich",
                        "Maia", "Nahn", "Ogma", "Peacock"};


// Write your code here
int start = 0;
int stop = starNames.Length;
int step = 2;

for (int index = start; index < stop; index = index + step)
{
    Console.WriteLine (starNames[index]);
}

// Alternative
for (int i=0; i<starNames.Length; i=i+2)
{
    Console.WriteLine (starNames[i]);
}

```

Java

```java
// Q03bFINISHED


public class Q03bFINISHED
{
    public static void main (String[] args)
    {
        String starNames[] = {"Alasia", "Beid", "Castor", "Denebola",
                              "Electra", "Fafnir", "Gudja", "Haedus",
                              "Izar", "Jishui", "Kang", "Lich",
                              "Maia", "Nahn", "Ogma", "Peacock"};

        // Write your code here
        int start = 0;
        int stop = starNames.length;
        int step = 2;

        for (int index = start; index < stop; index = index + step)
        {
            System.out.println (starNames[index]);
        }

        // Alternative
        for (int i=0; i<starNames.length; i=i+2)
        {
            System.out.println (starNames[i]);
        }

    }
}
```

Python

```python
# Q03bFINISHED

starNames = ["Alasia", "Beid", "Castor", "Denebola",
             "Electra", "Fafnir", "Gudja", "Haedus",
             "Izar", "Jishui", "Kang", "Lich",
             "Maia", "Nahn", "Ogma", "Peacock"]

# Write your code here

start = 0
stop = len (starNames)
step = 2

for index in range (start, stop, step):
    print (starNames[index])

# Alternative
for index in range (0, len (starNames), 2):
    print (starNames[index])
```

Question	mp	Answer	Additional guidance	Mark
3(c)		Award **one** mark for each:		
	C1	Take a string input (1)	• `aString = Console.ReadLine ();` • `aString = myKeyboard.nextLine();` • `aString = input ("Enter a number: ")`	**6**
	C2	Complete the selection with 'if' (1)		
	C3	Convert string to integer (1)	• `aNum = Convert.ToInt32 (aString);` • `aNum =  Integer.parseInt (aString);` • `aNum = int (aString)`	
	C4	Any one correct set of relational operators (1) ignore logical operator first two tests ignore test dealing with >= 1	Python • `(aNum <= 20) or (aNum >= 60)` • `(aNum < 21) or (aNum > 59)` • `(aNum > 30) and (aNum < 40)` • `(aNum >= 31) and (aNum <= 39)` C# and Java • `((aNum <= 20) || (aNum >= 60))` • `(aNum < 21) || (aNum > 59))` • `((aNum > 30) && (aNum < 40))` • `((aNum >= 31) && (aNum <= 39))`	
	C5	Both logical operators used correctly in each test (1) Ignore relational operators First test: **or** (must attempt to address 20 and 60 with relational, even if C4 not awarded Second test: **and**		
	C6	Complete the third test (1)	All Languages: `(aNum == 30)`	

C#

```
// Q03cFINISHED

String aString = "";
int aNum = 0;

Console.WriteLine ("Enter a number:");

// Complete the line to take a string input
aString = Console.ReadLine ();

// Complete the selection statement
if (aString != "")
{
    // Convert the string to an integer
    aNum = Convert.ToInt32 (aString);
    if (aNum > 0)
    {
        // Complete the test
        if ((aNum <= 20) || (aNum >= 60))
        {
            Console.WriteLine ("Acceptable");
        }
        // Complete the test
        else if ((aNum >30) && (aNum < 40))
        {
            Console.WriteLine ("Centre");
        }
        // Complete the test
        else if (aNum == 30)
        {
            Console.WriteLine ("Perfect");
        }
        else
        {
            Console.WriteLine ("No message");
        }
    }
    else
    {
        Console.WriteLine ("The number must be greater than zero");
    }
}
else
{
```

Java

```java
// Q03cFINISHED

import java.util.Scanner;

public class Q03cFINISHED
{
    public static void main (String[] args)
    {
        String aString = "";
        int aNum = 0;
        Scanner myKeyboard = new Scanner (System.in);

        System.out.println ("Enter a number:");

        // Complete the line to take a string input
        aString = myKeyboard.nextLine();

        // Complete the selection statement
        if (aString != "")
        {
            // Convert the string to an integer
            aNum =  Integer.parseInt (aString);
            if (aNum > 0)
            {
                // Complete the test
                if ((aNum <= 20) || (aNum >= 60))
                {
                    System.out.println ("Acceptable");
                }
                // Complete the test
                else if ((aNum >30) && (aNum < 40))
                {
                    System.out.println ("Centre");
                }
                // Complete the test
                else if (aNum == 30)
                {
                    System.out.println ("Perfect");
                }
                else
                {
                    System.out.println ("No message");
                }
            }
            else
            {
                System.out.println ("The number must be greater than zero");
            }
        }
```

Python

```python
# Q03cFINISHED

# Complete the line to take a string input
aString = input ("Enter a number: ")

# Complete the selection statement
if (aString != ""):

    # Convert the string to an integer
    aNum = int (aString)

    if (aNum > 0):

        # Complete the test
        if (aNum <= 20) or (aNum >= 60):
            print ("Acceptable")
        # Complete the test
        elif (aNum > 30) and (aNum < 40):
            print ("Centre")
        # Complete the test
        elif (aNum == 30):
            print ("Perfect")
        else:
            print ("No message")
    else:
        print ("The number must be greater than zero")
else:
    print ("You must provide a number")
```

Question	mp	Answer	Additional Guidance	Mark
4(a)(i)	A1	Award **one** mark for: • (At the highest level) in the main scope/program/file (1)	Java and C#: allow answers that indicate inside namespace e.g. 'before main class'	**1**

Question	mp	Answer	Additional Guidance	Mark
4(a)(ii)	A2	Award **one** mark for: • Only from within the scope/subprogram/code block in which it is created/declared (1)		**1**

Question	mp	Answer	Additional guidance	Mark
4(b)	B1 B2	Award **two** marks for a linked explanation such as: • **Figure 4** executes fewer comparisons/tests (1), because (the use of nested ifs/else ifs/ means that) as soon as a test evaluates to true, all the remaining tests are skipped (1) • **Figure 5** executes more comparison/tests (1), because every if statement must be executed, even if any test before has already evaluated to true (1)		**2**

Question	mp	Answer	Additional guidance	Mark
4(c)		Award **one** mark for each:		
	C1	Subprogram must attempt to use both pString and pNum to generate Key (1)	Award attempted use, even if not correct	
	C2	1D indexing used to access chars in string (1)	Can be anywhere in code (main or subprogram) Any method e.g. • pString[0]	
	C3	2 pairs of separated chars obtained (1)	First 2 **and** last 2 characters obtained from pString Any method Can be anywhere in code • pString[0] + pString[1] • pString[2] + pString[3] • pString[0:2] • pString[2:4]	
	C4	Key concatenated correctly (1)	Key must be correct Can be anywhere in code	
	C5	Subprogram must return their key to the calling line (1)	Do not award return without a value • Value must be variable/value intended to be key	
	C6	Complete line to call new subprogram (1)	Must be fully correct `myNewKey = genNewKey(myString, myNumber)`	**6**

C#

```
// Q04cFINISHED

namespace Q04cFINISHED
{
    class Q04cFINISHED
    {
        static String genNewKey (String pString, int pNum)
        {
            String newKey;

            // Write your code below this line
            newKey = pString.Substring (0, 2);            // First two chars
            newKey = newKey + pNum.ToString();            // Number
            newKey = newKey + pString.Substring (2, 2);   // Last two chars

            return (newKey);
        }

        static void Main(string[] args)
        {
            String myNewKey = "";
            int myNumber = 0;
            String myString = "";

            Console.WriteLine ("Enter a string:");
            myString = Console.ReadLine();
            Console.WriteLine ("Enter a whole number: ");
            myNumber = Convert.ToInt32(Console.ReadLine());

            if (myString.Length != 4)
            {
                Console.WriteLine ("String must be four characters");
            }
            else
            {
                Console.WriteLine ("Original: " + myString +
                                   " " + myNumber.ToString());

                // Complete the call to the subprogram
                myNewKey = genNewKey (myString, myNumber);
                Console.WriteLine ("New: " + myNewKey);
            }
        }
    }
}

```

Java

```java
// Q04cFINISHED

import java.util.Scanner;

public class Q04cFINISHED
{
    static String genNewKey (String pString, int pNum)
    {
        String newKey;

        // Write your code below this line
        newKey = pString.substring(0,2);  // First two chars
        newKey = newKey + String.valueOf (pNum);              // Number
        newKey = newKey + pString.substring (2, 4);   // Last two chars

        return (newKey);
    }

    public static void main(String[] args)
    {
        String myNewKey = "";
        int myNumber = 0;
        String myString = "";
        Scanner myKeyboard = new Scanner (System.in);

        System.out.println ("Enter a string:");
        myString = myKeyboard.nextLine();
        System.out.println ("Enter a whole number: ");
        myNumber = myKeyboard.nextInt();

        if (myString.length() != 4)
        {
            System.out.println ("String must be four characters");
        }
        else
        {
            System.out.println ("Original: " + myString +
                    " " + String.valueOf(myNumber));

            // Complete the call to the subprogram
            myNewKey = genNewKey (myString, myNumber);
            System.out.println ("New: " + myNewKey);
        }
    }
}

```

Python

```python
# Q04cFINISHED

def genNewKey (pString, pNum):
    newKey = ""         # Make new word here

    # Write your code below this line
    newKey = pString[0:2]               # First two chars
    newKey = newKey + str (pNum)       # Number
    newKey = newKey + pString[2:4]     # Last two chars
    return (newKey)

myString = input ("Enter a string: ")
myNumber = int (input ("Enter a whole number: "))

if (len (myString) != 4):
    print ("String must be four characters")
else:
    print ("Original: ", myString, myNumber)

    # Complete the call to the subprogram
    myNewKey = genNewKey (myString, myNumber)
    print ("New word:", myNewKey)

```

Question	mp	Answer	Additional guidance	Mark
5(a)	A1 A2	Award **two** marks for a linked explanation such as: • Replacing individual variables with an array (using a single name) (1) means that comparisons could be done in a loop (1) • The animals could be processed in a loop (1) because they could be stored in an array (under a single name) (1)		**2**

Question	mp	Answer	Additional guidance	Mark
5(b)		Award **one** mark for each of:	Ignore contents of fourth row if given.	
	B1	First row high index and low index (1) • 9, 0	Accept • 20, 10	
	B2	First row mid point (1) • 4	Accept • 15	
	B3	Second row (1) • 3, 0, 1	Accept • 14, 10, 12	
	B4	Third row (1) • 0, 0, 0	Accept • 10, 10, 10	**4**

Example: Using index

highIndex	lowIndex	midPoint
9	0	4
3	0	1
0	0	0

Example: Using Values

highIndex	lowIndex	midPoint
20	10	15
14	10	12
10	10	10

Question	mp	Answer	Additional guidance	Mark
5(c)		Award **one** mark for each of:		
	C1	Opens the file for reading (1)		
	C2	Reads each line from the file (1)	Requires both read lines (prime, inside loop) for C# and Java. Requires a foreach for Python. Allow FT from C1	
	C3	Splits each line into individual strings (1)	Syntactically correct <anyString>.split(',') Does not need to be splitting correct line	
	C4	Individual values cast to int for processing (1)	Any method (index or foreach) allow float no FT allowed from C1 to C3 – must be full values between commas in file	
	C5	SubTotal += value **and** grandTotal += subTotal (1)	Any reasonable attempt at calculating both totals	
	C6	All given variable and constant names used throughout (1)	INPUT_FILE COMMA subTotal grandTotal	
	C7	Subtotals displayed / Grand total displayed (1)	Must be accurate as shown in QP Fig 8 Not hard-coded in any way	
	C8	The input file is closed, before exiting (1)	Syntactically correct Must be file name originally opened – FT from C1	**8**

Q	mp	Code Examples
5(c)		
	C1	`StreamReader theFile = new StreamReader (inputFile);`
		`BufferedReader theFile = new BufferedReader (new FileReader (inputFile));`
		`theFile = open (INPUT_FILE, "r")`

	C2	`line = theFile.ReadLine ();  // Prime loop`
		`… …`
		`line = theFile.ReadLine ();  // Inside loop`

		`line = theFile.ReadLine ();  // Prime loop`
		`… …`
		`line = theFile.ReadLine ();  // Inside loop`

		`for line in theFile:`
	C3	`string[] stringSales = line.Split (comma);`
		`String[] stringSales = line.split (comma);`
		`stringSales = line.split (COMMA)`
	C4	`Convert.ToInt32 (sale);`
		`Integer.parseInt (stringSales[i]);`
		`int (sale)`
	C5	`subTotal = subTotal + sale`
		`subTotal = subTotal + sale`

		subTotal = subTotal + sale
		grandTotal = grandTotal + subTotal;
		grandTotal = grandTotal + subTotal;
		grandTotal = grandTotal + subTotal
	C6	INPUT FILE COMMA subtotal grandTotal

	C7	1000 2000 3000 4000 5000 Grand total: 15000
	C8	`theFile.Close ();`
		`theFile.close ();`
		`theFile.close ()`

C#

```
// Q05cFINISHED

using System.IO;

String inputFile = "Sales.txt";     // Output file name
String comma = ",";                 // Use as a constant

int subTotal = 0;               // Subtotal for each line
int grandTotal = 0;             // Running total
String line;                    // Line read from file

// Complete the code to open the file for reading
String fullPath = "C:\\Q05c";
inputFile = fullPath + "\\" + inputFile;
StreamReader theFile = new StreamReader (inputFile);

// Complete the code to read the first line of the input file
line = theFile.ReadLine ();

// Loop as long as the line is not null
while (line != null)
{
    // Complete the code to split the line into a set of five strings
    string[] stringSales = line.Split (comma);

    subTotal = 0;
    // Add code to sum up each value in the set of five strings
    foreach (string sale in stringSales)
    {
        subTotal = subTotal + Convert.ToInt32 (sale);
    }

    // Add code to display the subtotal for the line
    Console.WriteLine (Convert.ToString (subTotal));

    // Add code to calculate the running total
    grandTotal = grandTotal + subTotal;

    // Complete the code to read the next line of the file
    line = theFile.ReadLine ();
}

// Add code to display the total of all lines in the file
Console.WriteLine ("Grand total: " + Convert.ToString (grandTotal));

// Complete the code to close the opened file
theFile.Close ();
```

Java

```java
// Q05cFINISHED

import java.io.FileReader;
import java.io.BufferedReader;

public class Q05cFINISHED
{
    public static void main(String[] args) throws Exception
    {
        String inputFile = "Sales.txt";    // Output file name
        String comma = ",";                 // Use as a constant

        int subTotal = 0;               // Subtotal for each line
        int grandTotal = 0;             // Running total
        String line;                    // Line read from file

        // Complete the code to open the file for reading
        String full_path = "C:\\Q05c";
        inputFile = full_path + "\\" + inputFile;
        BufferedReader theFile = new BufferedReader (new FileReader (inputFile));

        // Complete the code to read the first line of the input file
        line = theFile.readLine();

        // Loop as long as the line is not null
        while (line != null)
        {
            // Complete the code to split the line into a set of five strings
            String[] stringSales = line.split (comma);

            subTotal = 0;
            // Add code to sum up each value in the set of five strings
            for (int i = 0; i < stringSales.length; i++)
            {
                subTotal = subTotal + Integer.parseInt (stringSales[i]);
            }

            // Add code to display the subtotal for the line
            System.out.println (String.valueOf (subTotal));

            // Add code to calculate the running total
            grandTotal = grandTotal + subTotal;

            // Complete the code to read the next line of the file
            line = theFile.readLine();
        }

        // Add code to display the total of all lines in the file
        System.out.println ("Grand total: " + String.valueOf (grandTotal));

        // Complete the code to close the opened file
        theFile.close ();
    }
}
```

Python

```python
# Q05cFINISHED

INPUT_FILE = "Sales.txt"     # Output file name
COMMA = ","                  # Use as a constant

subTotal = 0         # Subtotal for each line
grandTotal = 0       # Running total

# Complete the code to open the file for reading
theFile = open (INPUT_FILE, "r")

# Complete the code to read each line of the input file
for line in theFile:

    # Complete the code to split the line into a set of five strings
    stringSales = line.split (COMMA)

    subTotal = 0
    # Add code to sum up each value in the set of five strings
    for sale in stringSales:
        subTotal = subTotal + int (sale)

    # Add code to display the subtotal for the line
    print (str (subTotal))

    # Add code to calculate the running total
    grandTotal = grandTotal + subTotal

# Add code to display the total of all lines in the file
print ( "Grand total: " + str (grandTotal))

# Complete the code to close the opened file
theFile.close ()

```

Question	mp	Answer	Additional guidance	Mark
6		Award **one** mark for each of:		
	A1	Put new group of words (newspaper, magazine) into last blank position in array (1)	Any method Do not award hard-coded changes to table Note: *if* the solution will not work for any length of array. This will impact LBMS functionality.	
	A2	Loop for processing every group of words in the array (1)	Allow hard-coded length	
	A3	Loop controlled by the length of the array (1)	Method/function to determine the number of groups in the array	
	A4	Mechanism for tracking group number to appear in output (1)	Any method	
	A5	Output includes row number and original group of words (1)	Ignore 'punctuation' Requires both a line number and each word pair	
	A6	Mechanism for finding longest word in each group (1)	Method/function to determine the length of each string	
	A7	Mechanism for determining if the words in the group are out of order (1)	Any method	
	A8	Mechanism for putting the words in the group into alphabetical order (1)	Any method	
	A9	Output includes longest word in each group and any group found to be out of order is displayed in order (1)		
	A10	Evidence of two-dimensional indexing (1)	Any method	**11**

	A11	At least one instance of an informative comment (1)		

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	
	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)**

C#

```
// Q06FINISHED

String[,] tblWords = {{"apple", "banana"},
                      {"wrist", "leg"},
                      {"blue", "yellow"},
                      {"speaker", "keyboard"},
                      {"lavender", "tulip"},
                      {"pencil", "chalk"},
                      {"apartment", "house"},
                      {"bottom", "top"},
                      {"snow", "fog"},
                      {"beach", "mountain"},
                      {"", ""}};

String word1 = "newspaper";
String word2 = "book";

// ---------------------------------------------
// Write your code below this line
int numRows = tblWords.GetLength(0);
String outString = "";              // For output
int indexMaxLength = 0;             // For checking lengths
String temp = "";                    // For swapping

// Insert a new group of words into the blank row
tblWords[numRows - 1, 0] = word1;
tblWords[numRows - 1, 1] = word2;

```

```
// Loop for each individual group of words in the table
for (int i = 0; i < numRows; i++)
{
    // Pick up an entire row
    String[] words;
    words = new String[2];
    words[0] = tblWords[i, 0];
    words[1] = tblWords[i, 1];

    // Display the row number and the original content
    outString = Convert.ToString (1 + i);
    outString = outString + " " + words[0] + " " + words[1];
    Console.WriteLine (outString);

    // Find the length of the longest word and display the word
    indexMaxLength = 0;             // Longest is first word
    if (words[1].Length > words[indexMaxLength].Length)
    {
        indexMaxLength = 1;
    }
    // Display indented
    Console.WriteLine ("\t" + words[indexMaxLength]);

    // Check if the words in this group are in order
    if (words[0][0] > words[1][0])
    {
        // This group is out of order so needs ordering
        temp = words[0];
        words[0] = words[1];
        words[1] = temp;

        Console.WriteLine ("\t" + words[0] + " " + words[1]);
    }
}
```

Java

```java
// Q06FINISHED

public class Q06FINISHED
{
    public static void main(String[] args)
    {
        String[][] tblWords = {{"apple", "banana"},
                               {"wrist", "leg"},
                               {"blue", "yellow"},
                               {"speaker", "keyboard"},
                               {"lavender", "tulip"},
                               {"pencil", "chalk"},
                               {"apartment", "house"},
                               {"bottom", "top"},
                               {"snow", "fog"},
                               {"beach", "mountain"},
                               {"", ""}};

        String word1 = "newspaper";
        String word2 = "book";

        // ---------------------------------------------
        // Write your code below this line
        int numRows = tblWords.length;
        String outString = "";              // For output
        int indexMaxLength = 0;             // For checking lengths
        String temp = "";                    // For swapping

        // Insert a new group of words into the blank row
        tblWords[numRows - 1][0] = word1;
        tblWords[numRows - 1][1] = word2;

```

```
        // Loop for each individual group of words in the table
        for (int i = 0; i < numRows; i++)
        {
            // Pick up an entire row
            String[] words = tblWords[i];

            // Display the row number and the original content
            outString = String.valueOf(1 + i) + " " +  words[0] + " " + words[1];
            System.out.println (outString);

            // Find the length of the longest word and display the word
            indexMaxLength = 0;             // Longest is first word
            if (words[1].length() > words[indexMaxLength].length())
            {
                indexMaxLength = 1;
            }

            // Display indented
            System.out.println ("\t" + words[indexMaxLength]);

            // Check if the words in this group are in order
            if (words[0].charAt (0) > words[1].charAt(0))
            {
                // This group is out of order so needs ordering
                temp = words[0];
                words[0] = words[1];
                words[1] = temp;

                System.out.println ("\t" + words[0] + " " + words[1]);

            }
        }
    }
}

```

Python

```python
# Q06FINISHED

tblWords = [["apple", "banana"],
            ["wrist", "leg"],
            ["blue", "yellow"],
            ["speaker", "keyboard"],
            ["lavender", "tulip"],
            ["pencil", "chalk"],
            ["apartment", "house"],
            ["bottom", "top"],
            ["snow", "fog"],
            ["beach", "mountain"],
            ["", ""]]

word1 = "newspaper"
word2 = "book"

# ---------------------------------------------
# Write your code below this line
row = 1                 # Row number
numRows = len (tblWords)

# Insert a new group of words into the blank row
tblWords[numRows - 1][0] = word1
tblWords[numRows - 1][1] = word2

# Loop for each individual group of words in the table
for words in tblWords:
    # Display the row number and the original content
    print (str (row) + " " + words[0] + " " + words[1])

    # Find the length of the longest word and display the word
    indexMaxLength = 0          # Longest is first word
    if (len (words[1]) > len (words[indexMaxLength])):
        indexMaxLength = 1

    # Display indented
    print (" "*5 + words[indexMaxLength])

    # Check if the words in this group are in order
    if (words[0] > words[1]):
        # This group is out of order so needs sorting
        temp = words[0]
        words[0] = words[1]
        words[1] = temp
```

Output

```
1 apple banana
     banana
2 wrist leg
     wrist
     leg wrist
3 blue yellow
     yellow
4 speaker keyboard
     keyboard
     keyboard speaker
5 lavender tulip
     lavender
6 pencil chalk
     pencil
     chalk pencil
7 apartment house
     apartment
8 bottom top
     bottom
9 snow fog
     snow
     fog snow
10 beach mountain
```

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