Mark Scheme (Results)

Summer 2019

Pearson Edexcel International GCSE
In Computer Science (4CP0)

Paper 02: Application of Computational Thinking

Summer 2019
Publications Code 4CP0_02_1906_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.

This mark scheme is presented so that questions requiring a handwritten response are grouped at the start.
This is followed by the mark scheme for the questions which require submission of coding.
Examples of coding that would gain marks is grouped at the end in order of coding language.

Question	Answer	Additional Guidance	Mark
1(a)	**The only correct answer is C** A is not correct because Boolean only has values True and False B is not correct because it only allows for one character D is not correct because it is used for mixed text and numeric data		(1)
1(b)	(an error that) produces incorrect /unintended output (1)	Accept an example of an explained logic error	(1)
1(d)	Any one from: 1. explain how the program works (1) 2. improve readability of the program (1) 3. make code more understandable (1)		(1)
1(e)	1. Global variables created in the main program / can be accessed from any part of the program / are in scope throughout the program (1) 2. Local variables are declared/created inside a function/blocked code (indented, brackets) and are only accessible there / are only in scope within the function in which they're declared (1)		(2)

<table>
<tr><th>Question</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(g)</td><td>For up to <b>two</b> tests
Award 1 mark for an appropriate validation test and 1 mark for an item of test data that would fail <b>only</b> the given test
<table>
<tr><th>Validation test</th><th>Erroneous data</th></tr>
<tr><td>Are the first two characters letters/upper case letters?</td><td>1234CCC</td></tr>
<tr><td>Are the third- and fourth- characters numerical digits?</td><td>CCCCBBB</td></tr>
<tr><td>Are the last three characters letters/upper case letters?</td><td>CC12345</td></tr>
</table>
</td><td>Each test should target a different feature of one of the requirements
• Exactly 7 characters (in the stem)
• First two characters are letters / upper case letters
• Characters 3 & 4 are numeric
• Last three characters are letters / upper case letters</td><td>(4)</td></tr>
</table>

Question	Answer	Additional Guidance	Mark
2(b)(i)	24.0 (1)	Accept 24	(1)
2(b)(ii)	One of: By the rules of precedence multiplication is done before addition. (1) Without the brackets the addition would be done to the total of the multiplication. (1) By the rules of the BODMAS/BIDMAS additions happen after multiplication. (1) newSpeed and intialSpeed have to be added together before multiplying (by time). (1)		(1)

Question	Answer	Additional Guidance	Mark
2(b)(iii)	Real (1) Because real x integer = real (1)	Accept double / float Accept 'because it could be a number with decimal places'	(2)
2(b)(iv)	27.375 (1)		(1)
Q2(c)(i)	C#: 16 or 20 (1) Java: 16 or 21 (1) Python: 8 or 10 (1)		(1)
Q2(c)(ii)	C#: 29 or 31 (1) Java: 24 or 26 (1) Python: 14 or 15 (1)		(1)
Q2(c)(iii)	Any of: 1. Message (1) 2. startPos (Python) / pos (Java or C#) (1) 3. characters (1)		(1)
Q2(c)(iv)	newMessage (1)	Do not accept outMessage	(1)
Q2(c)(v)	Response starts with 'c' (1) Response has six letters in sequence from 'educational' (1)	Marks can be awarded from Q2(c)(vi)	(2)
Q2(c)(vi)	1. Returns substring / crops a string (1) 2. Located between the two input values / with number of characters identified in the last parameter (characters) and starting at the second parameter (startPos) (1)	Award two marks for: It produces a substring of the message starting at position startPos with length characters Award up to 2 marks for a correct interpretation of the code (e.g 'obtain a newMessage' – is worth 1) Accept a correct generic explanation of what a subprogram does for 1 mark	(2)

Question	Answer	Additional Guidance	Mark
Q3(b)	More memory/storage required (1) Because not sorted in place / not efficient / many steps / division of data set / is recursive (1)		(2)

Question	Answer	Additional Guidance	Mark
4(a)	**The only correct answer is D** A is not correct because Plaintext is the original unencrypted text B is not correct because there is a consistency in the coding of the text C is not correct because a shift encryption replaces letters with other letters a given distance from the original letter (as in a Caesar cipher)		1
4(b)	Award 3 marks for all rows correct Award 2 marks for 4 rows correct Award 1 mark for 2 rows correct **G** **Q** **I** **X** **N** **U**		(3)
Q4(c)	One mark for correct decision: Yes (1) Up to two marks for the correct justifications: Each/every letter has a symbol / is represented (1) The symbol is a single/unique representation (1)	Award marks independently	(3)

Question	Answer	Additional Guidance	Mark
1(c)	**C#** 1. Semicolon at the end of line 9 [maxValue=5;] (1) 2. Consistent capitalisation of ‘count’ on lines 10, 11, 13 and 14 (1) 3. constantValue needs to be defined and given the value 7 / replaced by value 7 (1) **Java** 1. Semicolon at the end of line 8 [maxValue = 5;] (1) 2. Consistent capitalisation of ‘count’ on lines 7, 9, 11 and 12 (1) 3. constantValue needs to be defined and given the value 7 / replaced by 7 (1) **Python** 1. Colon needed at end of line 7 [count, maxValue):] (1) 2. Consistent capitalisation of ‘count’ on line 3, 7, 8 and 9 (1) 3. constantValue needs to be defined and given the value 7 / replaced by 7 (1)	Candidates are required to open the file Q01c in the code editor. Amended code should be saved as Q01cFINISHED. Line numbers may have been changed by corrections to code (e.g. insertion of constantValue definition)	(3)

<table>
<tr><th>Question</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(f)</td><td>Award 1 mark for <b>each</b> correct pair of relational operators and 1 mark for correct Boolean operator for <b>each</b> condition.
<table>
<tr><td>Condition</td><td>Coding</td></tr>
<tr><td rowspan="2">Shop income more than £5000 or assistant sold at least 10 pairs</td><td>shopIncome > 5000; assistantSales >= 10 (1)</td></tr>
<tr><td>correct <b>OR</b> operator (1)</td></tr>
<tr><td rowspan="2">Shop income at least £2000 and assistant sold at least 5 pairs</td><td>shopIncome >= 2000; assistantSales >= 5 (1)</td></tr>
<tr><td>correct <b>AND</b> operator (1)</td></tr>
</table>
Accept >9, >1999 and >4 for respective >= comparisons</td><td>Candidates are required to open the file Q01f in the code editor.
Amended code should be saved as Q01fFINISHED.
Do not penalise candidates who attempt more than the stated requirements.
Do not award Boolean operator mark for single | or & as these are not correct.</td><td>(4)</td></tr>
</table>

Question	Answer	Additional Guidance	Mark
2(a)	Award 1 mark for each of: 1. set counter to 1 and guess to 0 (1) 2. set answer to random value between 1 and 10 (1) 3. request input of and accept an integer value for guess (1) 4. create a while loop with a correct condition (1) 5. increment the counter (1) 6. use if else selection to determine and display appropriate output message for incorrect guesses (1) 7. request input of and accept another integer value for a guess inside the loop (1) 8. display correct guess and count of guesses (1) 9. compiling without syntax errors (1) 10. coding meets all requirements of question (1)	Candidates are required to open the file Q02a in the code editor. Amended code should be saved as Q02aFINISHED. 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.	(10)

Question	Answer	Additional Guidance	Mark
Q3(a)	Award 1 mark each for: 1. File opened for input (1) 2. File opened for output (1) 3. Loop for processing data (1) 4. Check for '@' symbol (1) 5. Write invalid address to output file (1) 6. Close files (1)	Amended code should be saved as Q03aFINISHED. Do not penalise candidates who attempt more than the stated requirements.	(6)

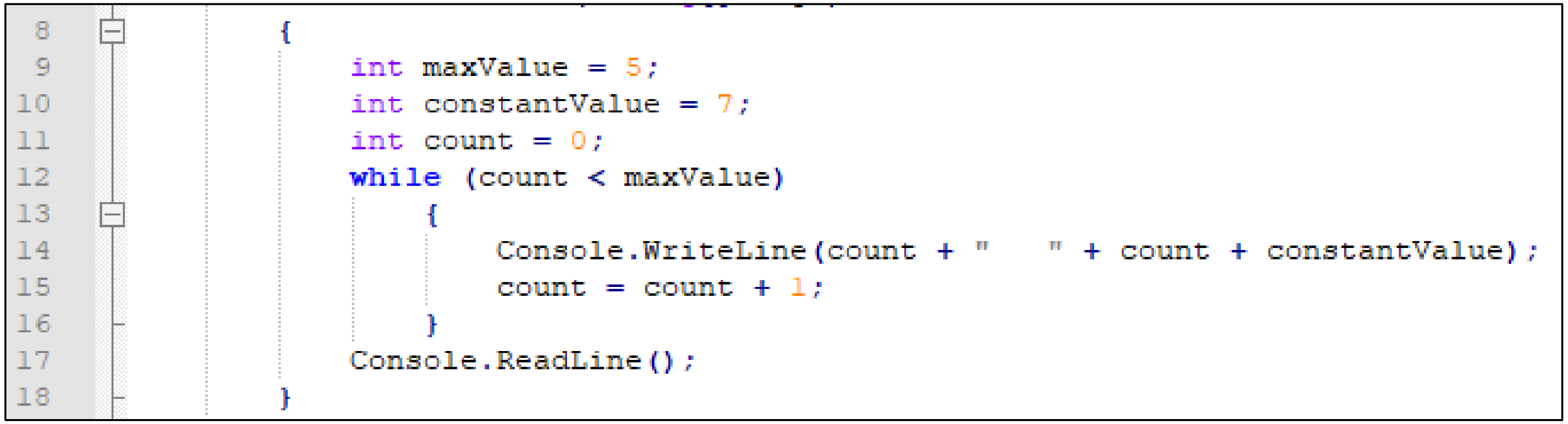
Question	Answer	Additional Guidance	Mark
Q3(c)	Award 1 mark for each section of code that: 1. Asks for input of a number and stores it in a variable with a meaningful name (1) 2. Checks number is between 1 and 50 (1) 3. Calculates square and cube of number (1) 4. Displays suitable labels e.g. ‘Number’, ‘Square’ and ‘Cube’ and relevant values (1) For the whole code: 5. Compiles and runs without logical or syntax errors (1) 6. Efficient in terms of computation, storage and selection of programming constructs (1)	Amended code should be saved as Q03cFINISHED. Allow follow through for mark point 4 when square and cube are incorrectly calculated. Credit alternative coding which produces same results. Accept use of exponent operators: C# e.g. Math.Pow (2, 3) = 8 Java e.g. Math.pow (2, 3) = 8 Python e.g. 2**3 = 8.	(6)

Question	Designated marks:	Additional Guidance	Mark
Q5	Designated marks: 1. Initialise variables for at least two of total number of books, count of pupils and average 2. Initialise variables for gold, silver and bronze / sort array in order of books read 3. Create loop for identifying (and printing) those borrowing fewer than 10 books 4. Print out appropriate details (must include pupil ID) 5. Print out total of books borrowed 6. Print out average number of books borrowed 7. Identification of gold medal winner 8. Identification of silver medal winner 9. Identification of bronze medal winner 10. Details of at least one of gold, silver and bronze medal winner printed (minimum last name or first name) 11. Details of all three medal winners (first name and last name)	Award marks independently Award marks for reasonable attempt to meet requirement.	(11)

Question 5 Overview marks:				
Mark Band 0	**Mark Band 1**	**Mark Band 2**	**Mark Band 3**	**Marks**
0	1 – 3	4 – 6	7 – 9	
No rewardable content	Little attempt to decompose the problem into component parts Some parts of the logic are clear and appropriate to the problem Some appropriate use and manipulation of data types, variables, data structures and program constructs Parts of the code are clear and readable Finished program will not be flexible enough with other data sets or input The program meets some of the given requirements	Some attempt to decompose the problem into component parts Most parts of the logic are clear and mostly appropriate to the problem The use and manipulation of data types, variables and data structures and program constructs is mostly appropriate Code is mostly clear and readable Finished program will function with some but not all other data sets or input The program meets most of the given requirements	The problem has been decomposed into component parts The logic is clear and appropriate to the problem The use and manipulation of data types, variables and data structures and program constructs is appropriate Code is clear and readable Finished program could be used with other data sets or input The program fully meets the given requirements	(9)

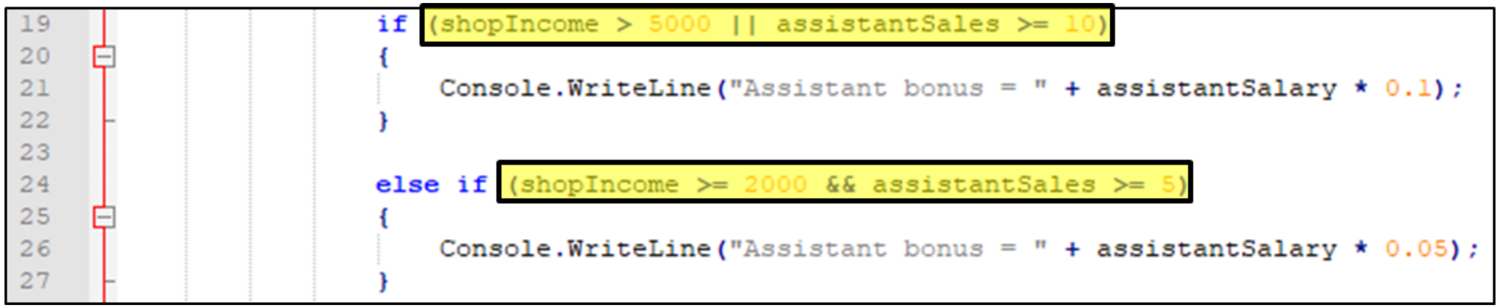
C# Example code

1(c)



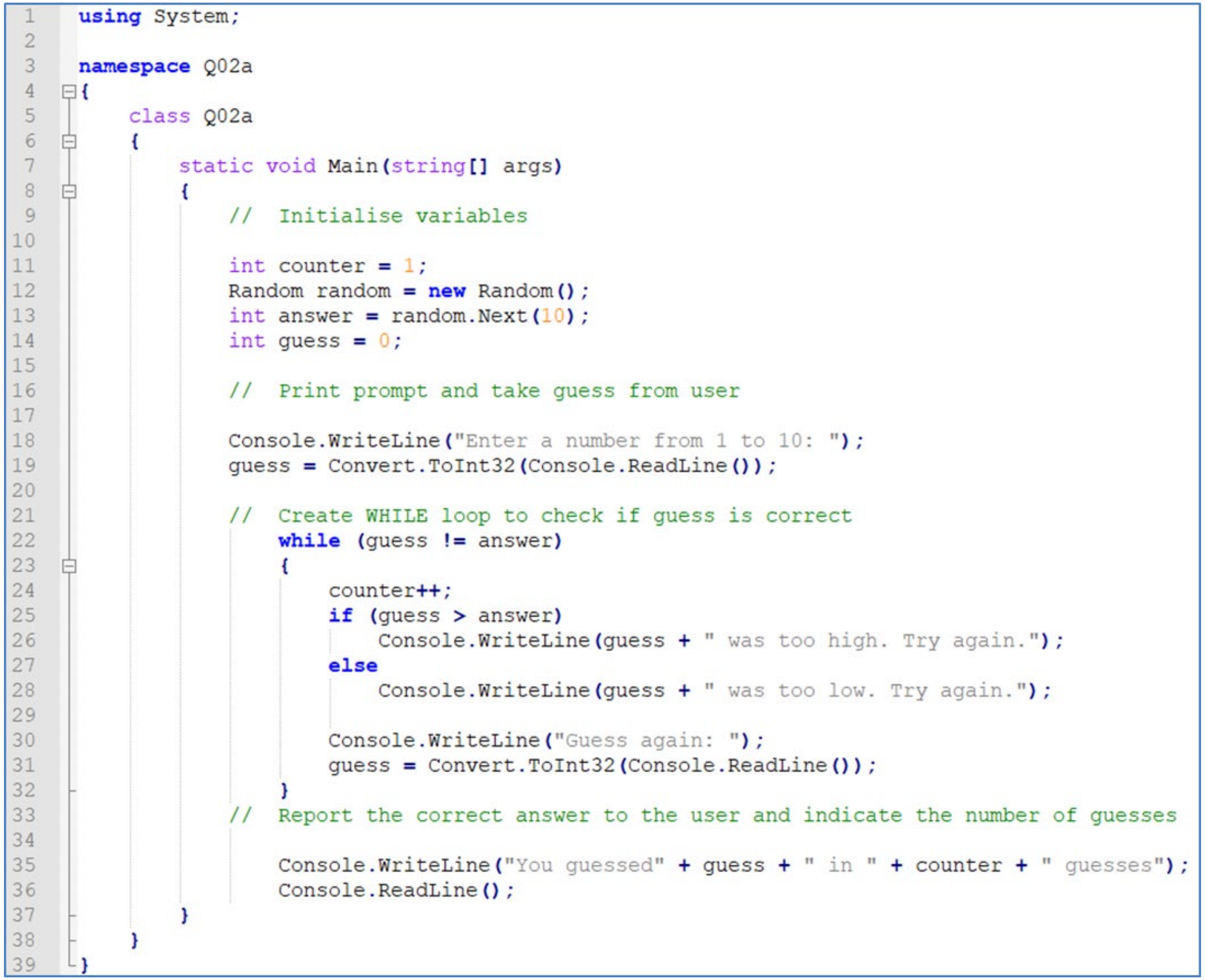
```
{
    int maxValue = 5;
    int constantValue = 7;
    int count = 0;
    while (count < maxValue)
    {
        Console.WriteLine(count + "   " + count + constantValue);
        count = count + 1;
    }
    Console.ReadLine();
}
```

1(f)



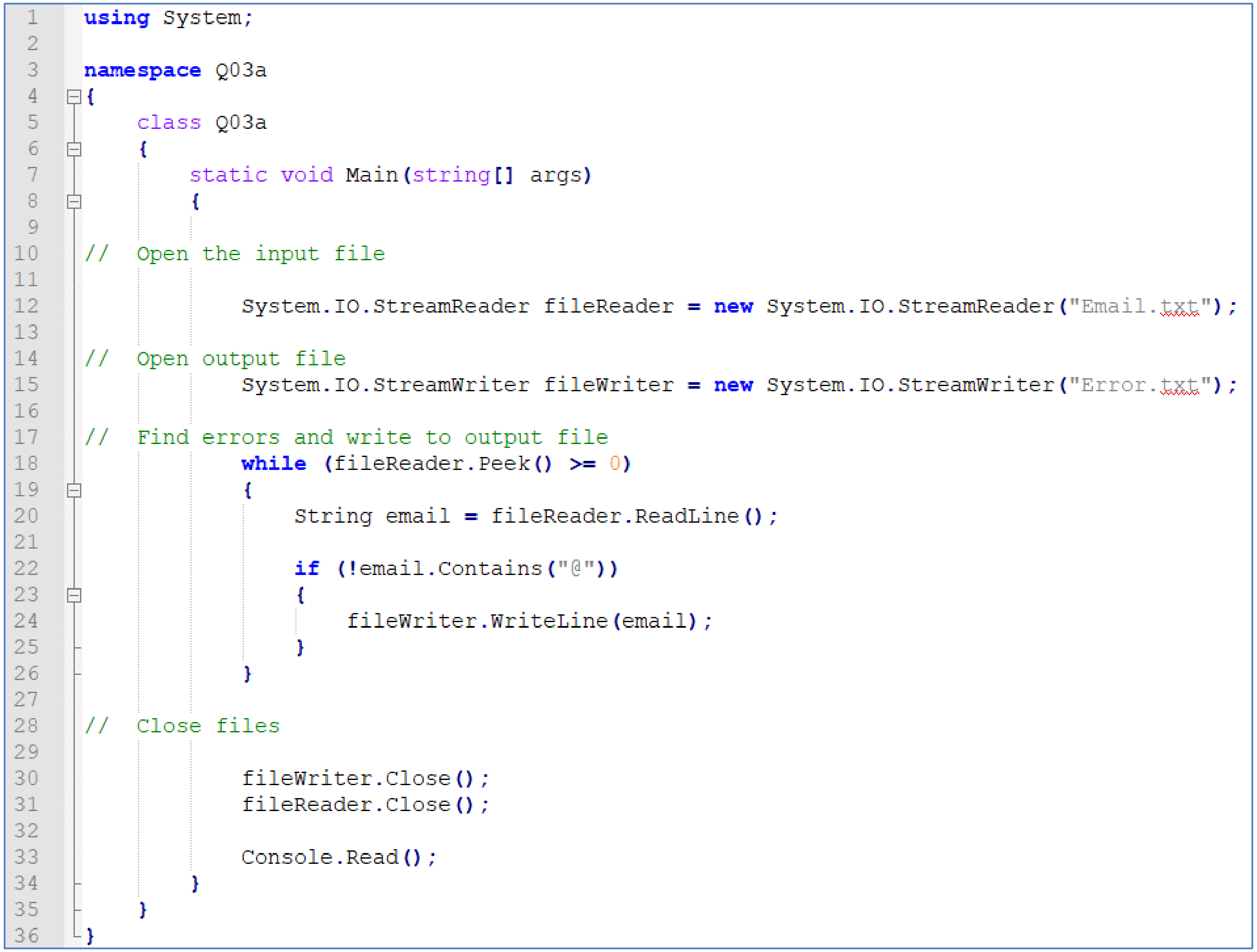
```
if (shopIncome > 5000 || assistantSales >= 10)
{
    Console.WriteLine("Assistant bonus = " + assistantSalary * 0.1);
}

else if (shopIncome >= 2000 && assistantSales >= 5)
{
    Console.WriteLine("Assistant bonus = " + assistantSalary * 0.05);
}
```

2(a)



```
using System;

namespace Q02a
{
    class Q02a
    {
        static void Main(string[] args)
        {
            //  Initialise variables

            int counter = 1;
            Random random = new Random();
            int answer = random.Next(10);
            int guess = 0;

            //  Print prompt and take guess from user

            Console.WriteLine("Enter a number from 1 to 10: ");
            guess = Convert.ToInt32(Console.ReadLine());

            //  Create WHILE loop to check if guess is correct
                while (guess != answer)
                {
                    counter++;
                    if (guess > answer)
                        Console.WriteLine(guess + " was too high. Try again.");
                    else
                        Console.WriteLine(guess + " was too low. Try again.");

                    Console.WriteLine("Guess again: ");
                    guess = Convert.ToInt32(Console.ReadLine());
                }
            //  Report the correct answer to the user and indicate the number of guesses

                Console.WriteLine("You guessed" + guess + " in " + counter + " guesses");
                Console.ReadLine();
        }
    }
}
```

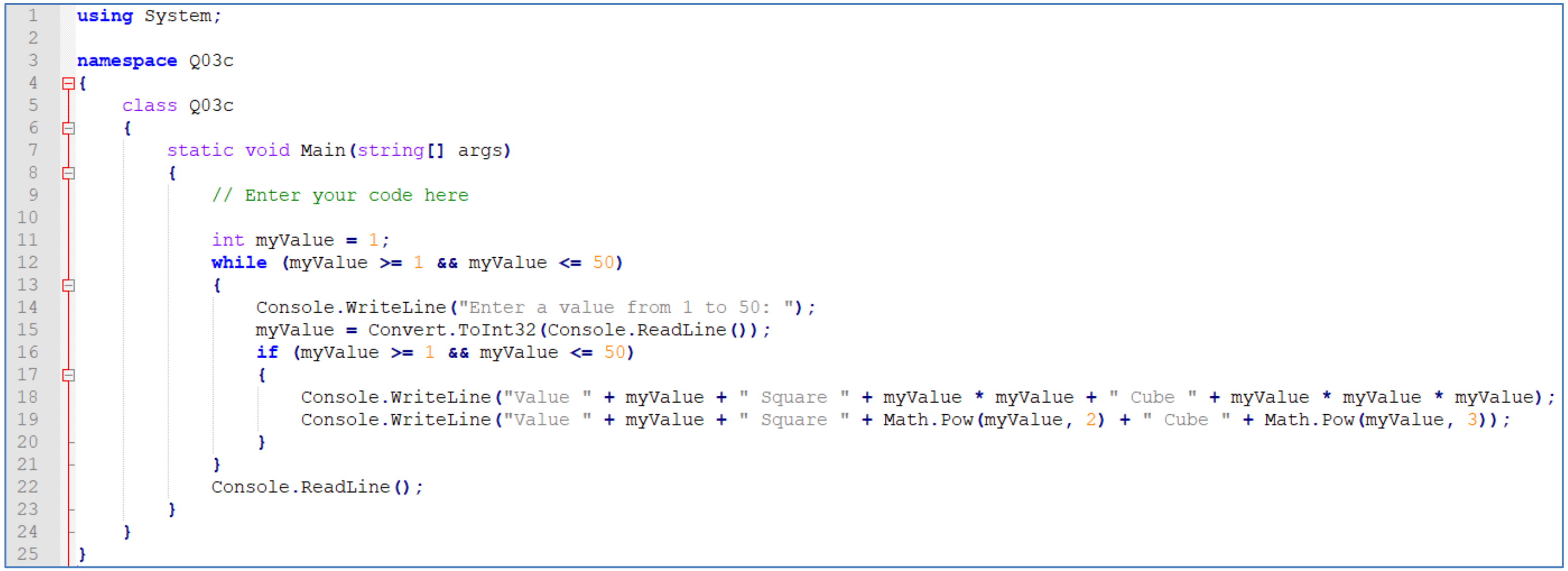
3(a)



```
using System;

namespace Q03a
{
    class Q03a
    {
        static void Main(string[] args)
        {

//  Open the input file

            System.IO.StreamReader fileReader = new System.IO.StreamReader("Email.txt");

//  Open output file
            System.IO.StreamWriter fileWriter = new System.IO.StreamWriter("Error.txt");

//  Find errors and write to output file
            while (fileReader.Peek() >= 0)
            {
                String email = fileReader.ReadLine();

                if (!email.Contains("@"))
                {
                    fileWriter.WriteLine(email);
                }
            }

//  Close files

            fileWriter.Close();
            fileReader.Close();

            Console.Read();
        }
    }
}
```

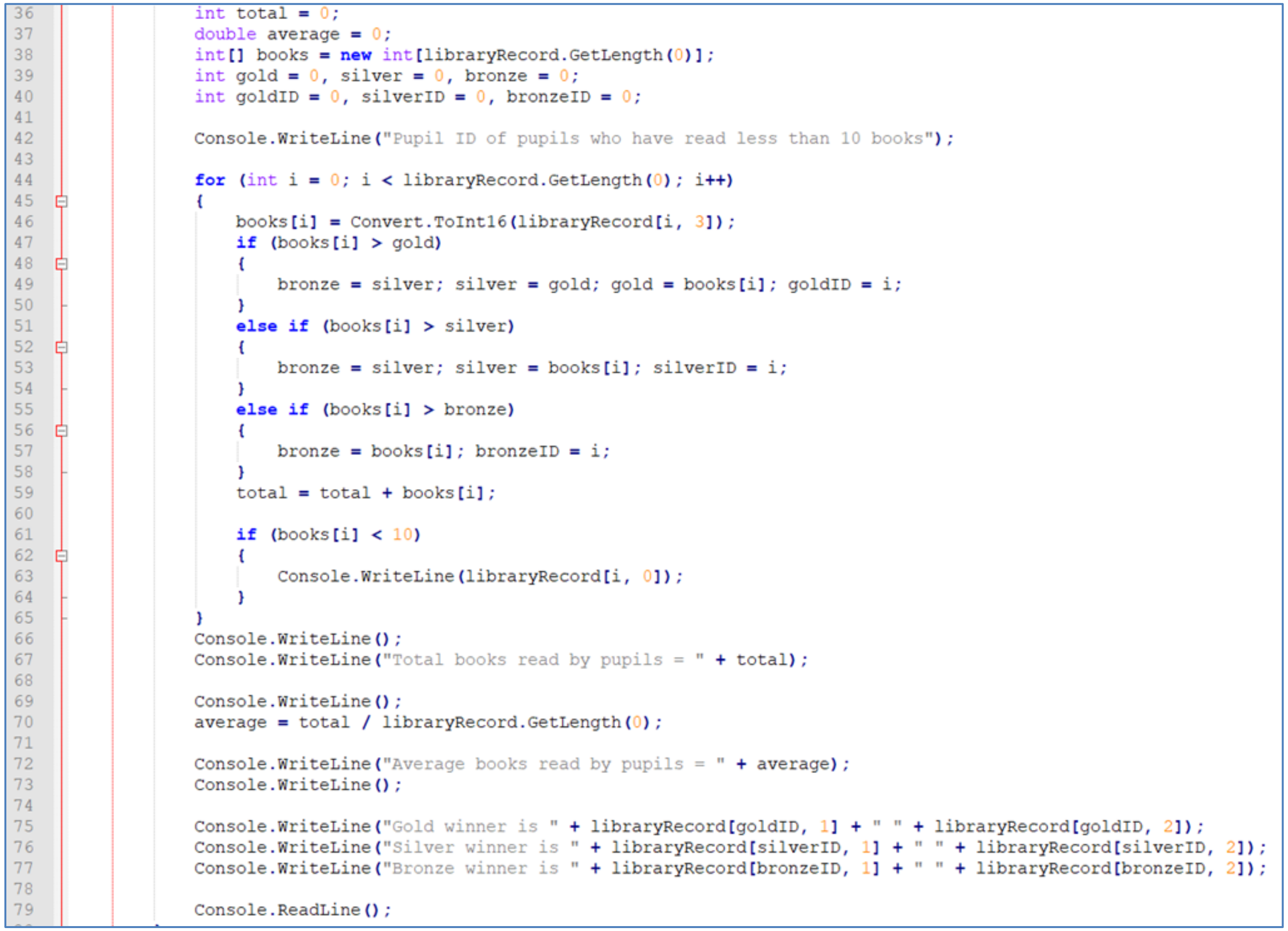
3(c)

Note: Line 19 is an alternative response to line 18



```
1   using System;
2
3   namespace Q03c
4   {
5       class Q03c
6       {
7           static void Main(string[] args)
8           {
9               // Enter your code here
10
11              int myValue = 1;
12              while (myValue >= 1 && myValue <= 50)
13              {
14                  Console.WriteLine("Enter a value from 1 to 50: ");
15                  myValue = Convert.ToInt32(Console.ReadLine());
16                  if (myValue >= 1 && myValue <= 50)
17                  {
18                      Console.WriteLine("Value " + myValue + " Square " + myValue * myValue + " Cube " + myValue * myValue * myValue);
19                      Console.WriteLine("Value " + myValue + " Square " + Math.Pow(myValue, 2) + " Cube " + Math.Pow(myValue, 3));
20                  }
21              }
22              Console.ReadLine();
23          }
24      }
25  }
```

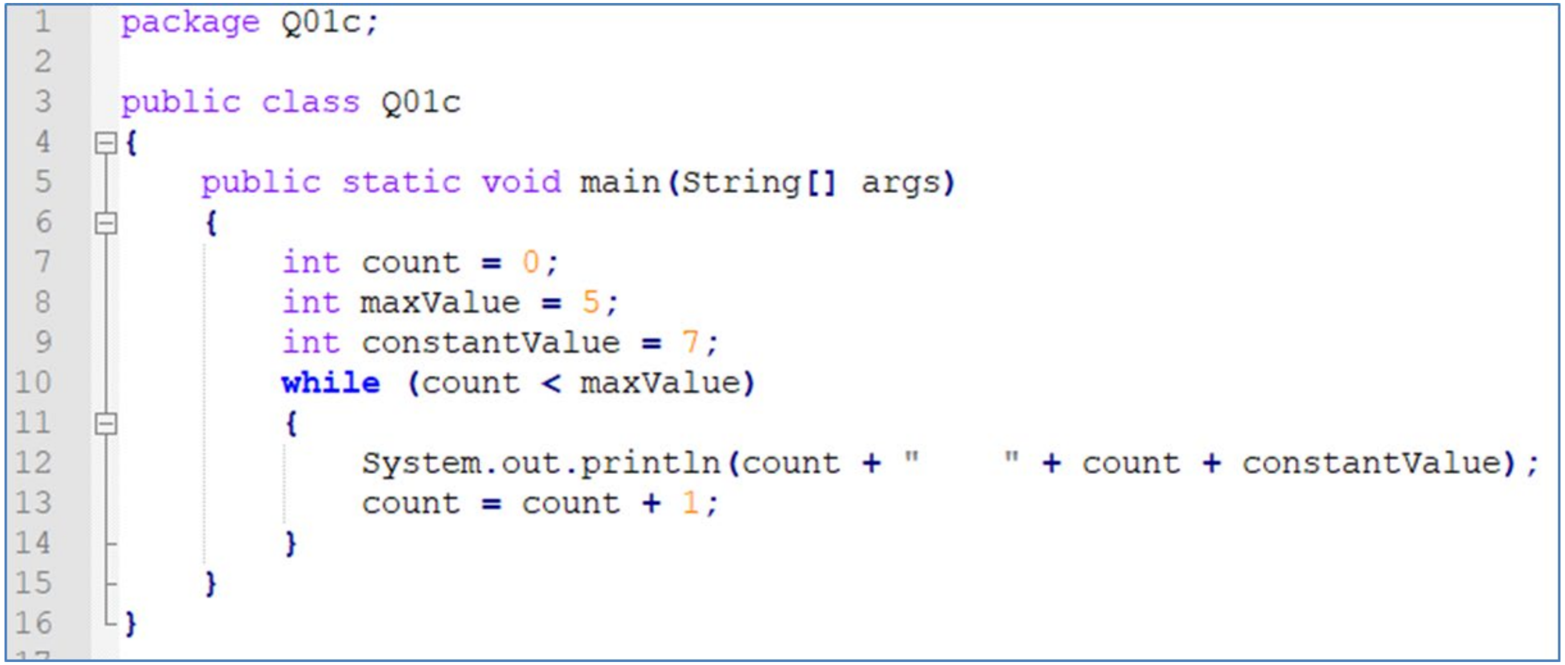
5



```
int total = 0;
double average = 0;
int[] books = new int[libraryRecord.GetLength(0)];
int gold = 0, silver = 0, bronze = 0;
int goldID = 0, silverID = 0, bronzeID = 0;

Console.WriteLine("Pupil ID of pupils who have read less than 10 books");

for (int i = 0; i < libraryRecord.GetLength(0); i++)
{
    books[i] = Convert.ToInt16(libraryRecord[i, 3]);
    if (books[i] > gold)
    {
        bronze = silver; silver = gold; gold = books[i]; goldID = i;
    }
    else if (books[i] > silver)
    {
        bronze = silver; silver = books[i]; silverID = i;
    }
    else if (books[i] > bronze)
    {
        bronze = books[i]; bronzeID = i;
    }
    total = total + books[i];

    if (books[i] < 10)
    {
        Console.WriteLine(libraryRecord[i, 0]);
    }
}
Console.WriteLine();
Console.WriteLine("Total books read by pupils = " + total);

Console.WriteLine();
average = total / libraryRecord.GetLength(0);

Console.WriteLine("Average books read by pupils = " + average);
Console.WriteLine();

Console.WriteLine("Gold winner is " + libraryRecord[goldID, 1] + " " + libraryRecord[goldID, 2]);
Console.WriteLine("Silver winner is " + libraryRecord[silverID, 1] + " " + libraryRecord[silverID, 2]);
Console.WriteLine("Bronze winner is " + libraryRecord[bronzeID, 1] + " " + libraryRecord[bronzeID, 2]);

Console.ReadLine();
```

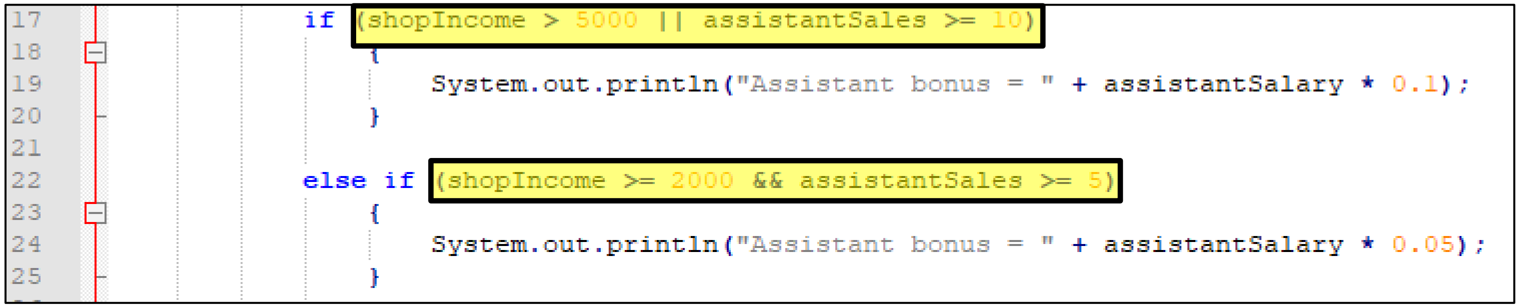
Java Example code

1(c)



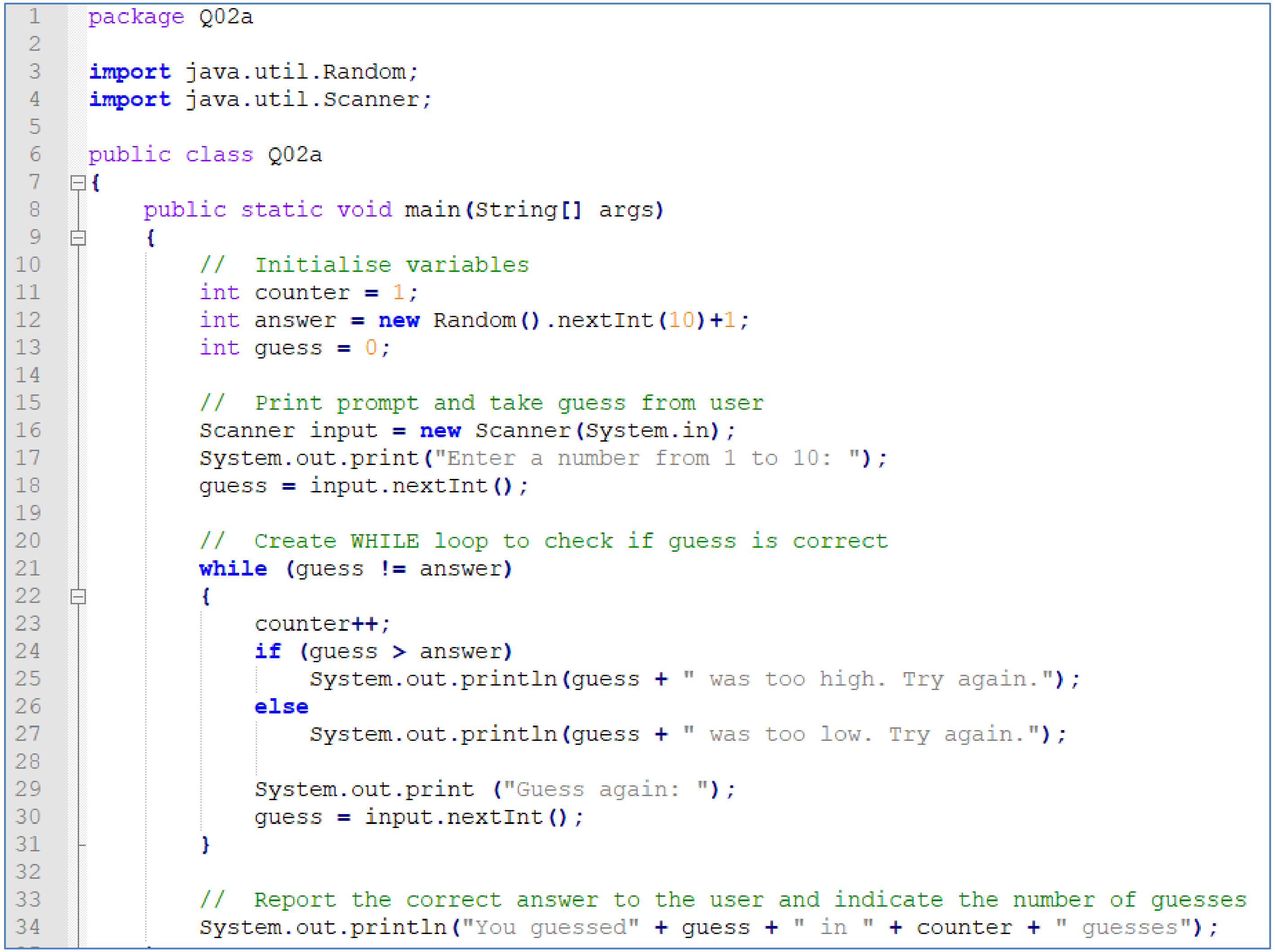
```
package Q01c;

public class Q01c
{
    public static void main(String[] args)
    {
        int count = 0;
        int maxValue = 5;
        int constantValue = 7;
        while (count < maxValue)
        {
            System.out.println(count + "     " + count + constantValue);
            count = count + 1;
        }
    }
}
```

1(f)



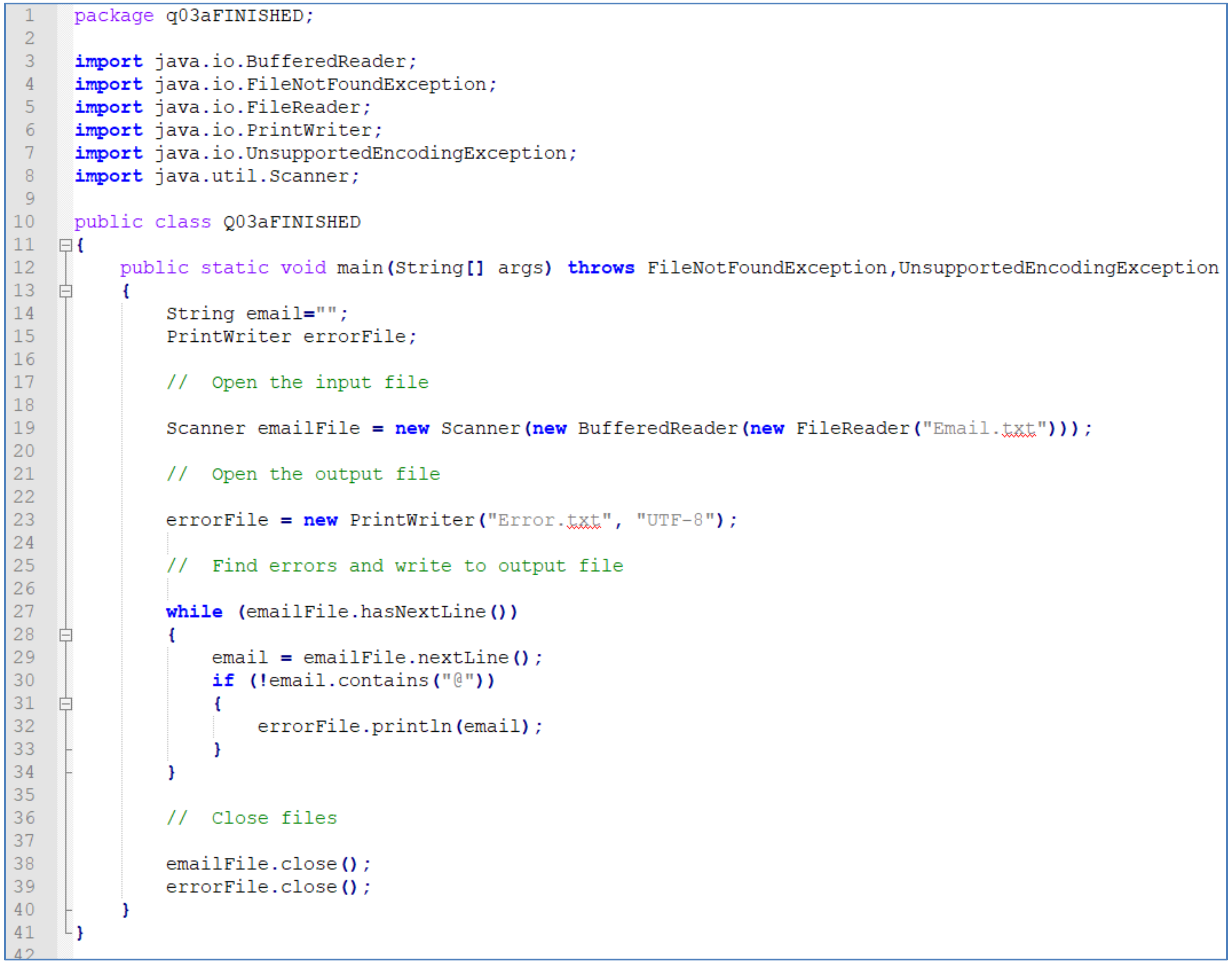
```
            if (shopIncome > 5000 || assistantSales >= 10)
            {
                System.out.println("Assistant bonus = " + assistantSalary * 0.1);
            }

            else if (shopIncome >= 2000 && assistantSales >= 5)
            {
                System.out.println("Assistant bonus = " + assistantSalary * 0.05);
            }
```

2(a)



```java
package Q02a

import java.util.Random;
import java.util.Scanner;

public class Q02a
{
    public static void main(String[] args)
    {
        //  Initialise variables
        int counter = 1;
        int answer = new Random().nextInt(10)+1;
        int guess = 0;

        //  Print prompt and take guess from user
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a number from 1 to 10: ");
        guess = input.nextInt();

        //  Create WHILE loop to check if guess is correct
        while (guess != answer)
        {
            counter++;
            if (guess > answer)
                System.out.println(guess + " was too high. Try again.");
            else
                System.out.println(guess + " was too low. Try again.");

            System.out.print ("Guess again: ");
            guess = input.nextInt();
        }

        //  Report the correct answer to the user and indicate the number of guesses
        System.out.println("You guessed" + guess + " in " + counter + " guesses");
```

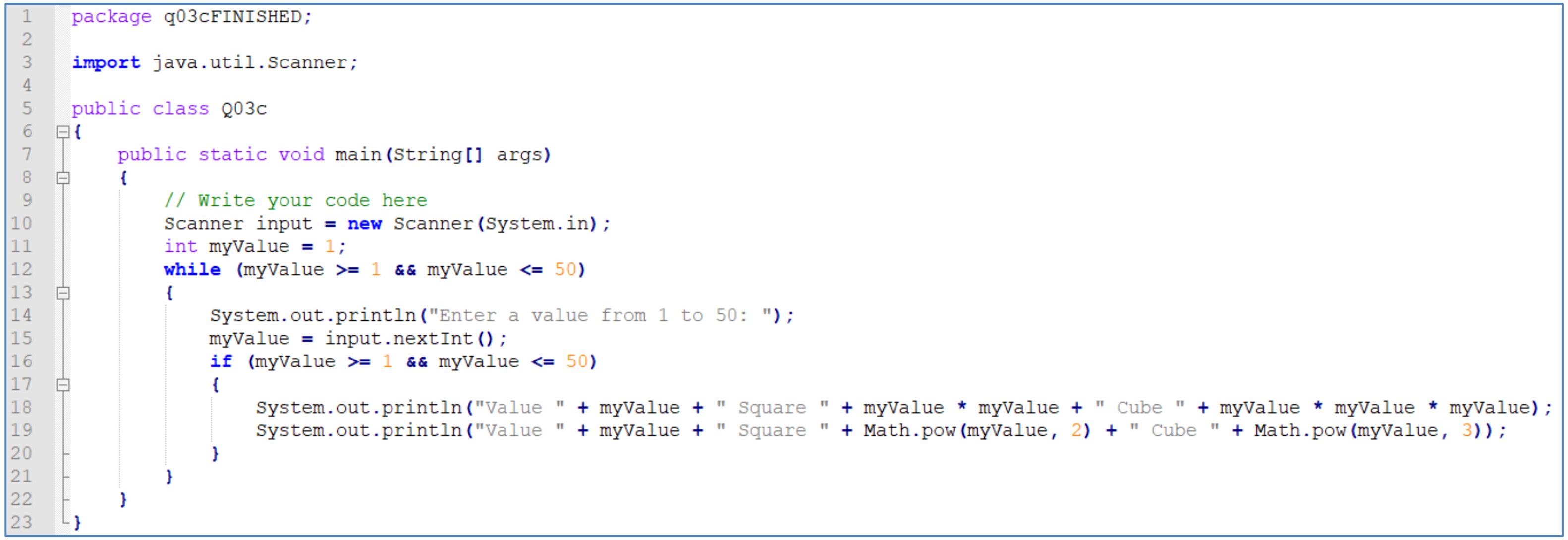
3(a)



```java
package q03aFINISHED;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.Scanner;

public class Q03aFINISHED
{
    public static void main(String[] args) throws FileNotFoundException,UnsupportedEncodingException
    {
        String email="";
        PrintWriter errorFile;

        //  Open the input file

        Scanner emailFile = new Scanner(new BufferedReader(new FileReader("Email.txt")));

        //  Open the output file

        errorFile = new PrintWriter("Error.txt", "UTF-8");

        //  Find errors and write to output file

        while (emailFile.hasNextLine())
        {
            email = emailFile.nextLine();
            if (!email.contains("@"))
            {
                errorFile.println(email);
            }
        }

        //  Close files

        emailFile.close();
        errorFile.close();
    }
}
```

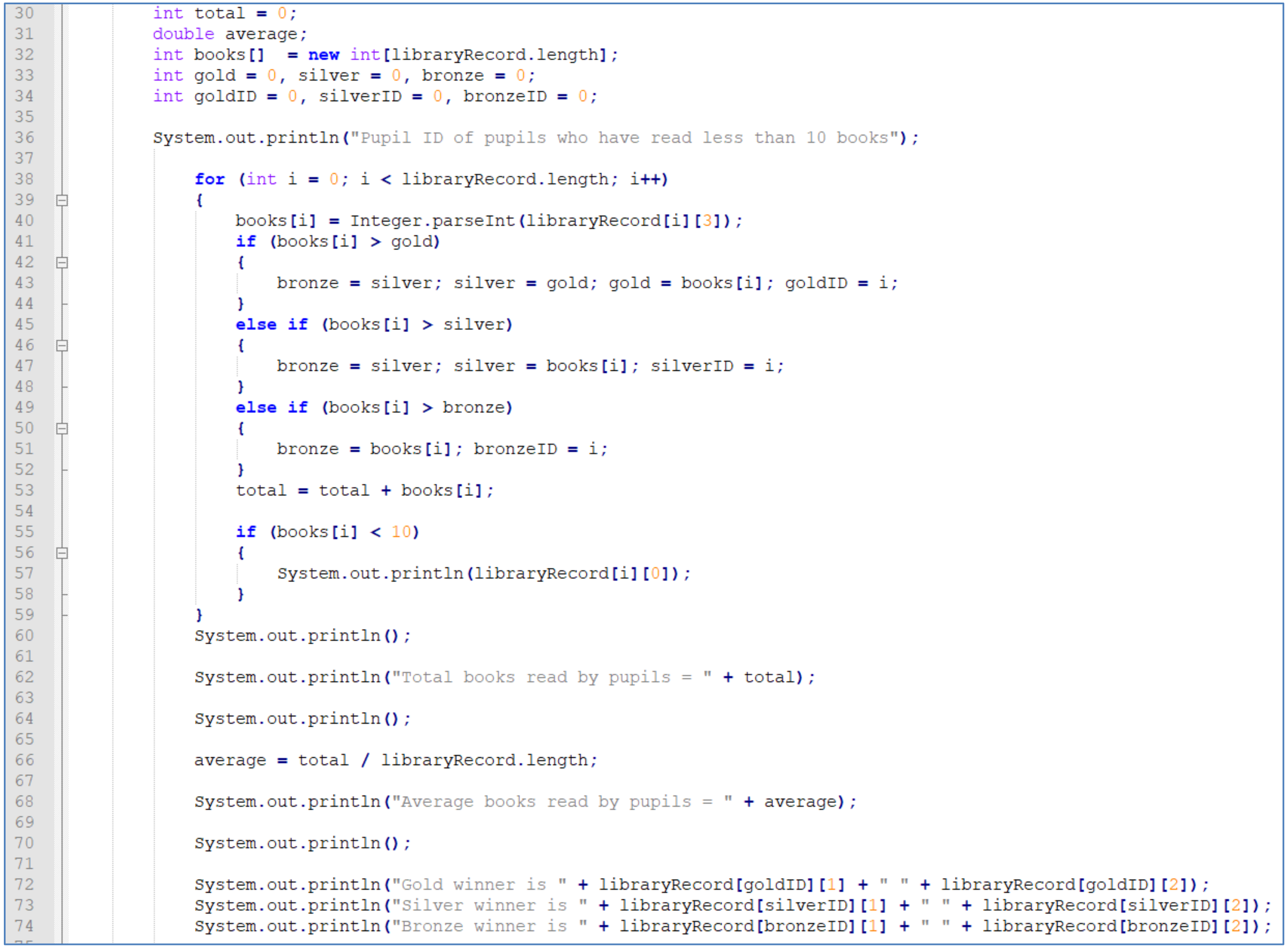
3(c)

Note: Line 19 is an alternative response to line 18



```
1   package q03cFINISHED;
2
3   import java.util.Scanner;
4
5   public class Q03c
6   {
7       public static void main(String[] args)
8       {
9           // Write your code here
10          Scanner input = new Scanner(System.in);
11          int myValue = 1;
12          while (myValue >= 1 && myValue <= 50)
13          {
14              System.out.println("Enter a value from 1 to 50: ");
15              myValue = input.nextInt();
16              if (myValue >= 1 && myValue <= 50)
17              {
18                  System.out.println("Value " + myValue + " Square " + myValue * myValue + " Cube " + myValue * myValue * myValue);
19                  System.out.println("Value " + myValue + " Square " + Math.pow(myValue, 2) + " Cube " + Math.pow(myValue, 3));
20              }
21          }
22      }
23  }
```

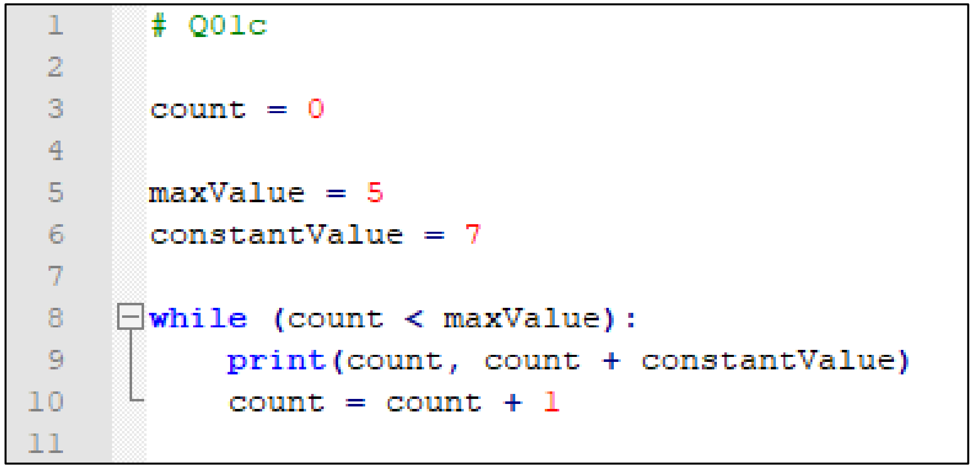
5



```
        int total = 0;
        double average;
        int books[]  = new int[libraryRecord.length];
        int gold = 0, silver = 0, bronze = 0;
        int goldID = 0, silverID = 0, bronzeID = 0;

        System.out.println("Pupil ID of pupils who have read less than 10 books");

            for (int i = 0; i < libraryRecord.length; i++)
            {
                books[i] = Integer.parseInt(libraryRecord[i][3]);
                if (books[i] > gold)
                {
                    bronze = silver; silver = gold; gold = books[i]; goldID = i;
                }
                else if (books[i] > silver)
                {
                    bronze = silver; silver = books[i]; silverID = i;
                }
                else if (books[i] > bronze)
                {
                    bronze = books[i]; bronzeID = i;
                }
                total = total + books[i];

                if (books[i] < 10)
                {
                    System.out.println(libraryRecord[i][0]);
                }
            }
            System.out.println();

            System.out.println("Total books read by pupils = " + total);

            System.out.println();

            average = total / libraryRecord.length;

            System.out.println("Average books read by pupils = " + average);

            System.out.println();

            System.out.println("Gold winner is " + libraryRecord[goldID][1] + " " + libraryRecord[goldID][2]);
            System.out.println("Silver winner is " + libraryRecord[silverID][1] + " " + libraryRecord[silverID][2]);
            System.out.println("Bronze winner is " + libraryRecord[bronzeID][1] + " " + libraryRecord[bronzeID][2]);
```

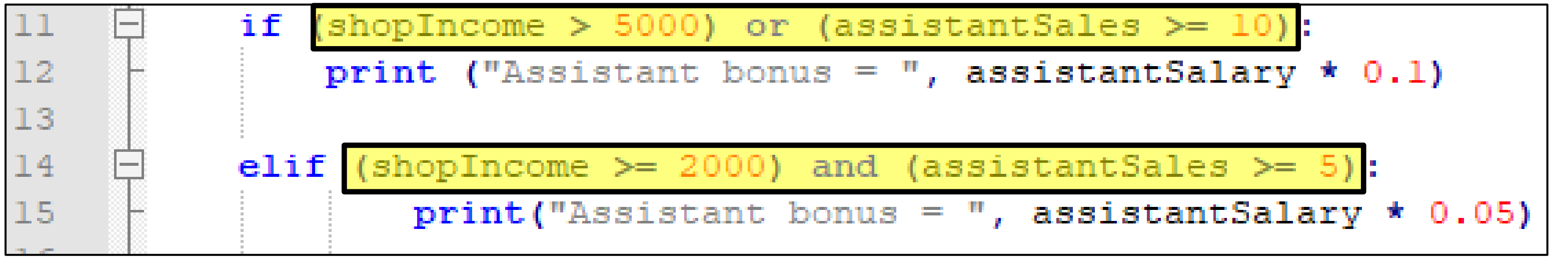
Python Example code

1(c)



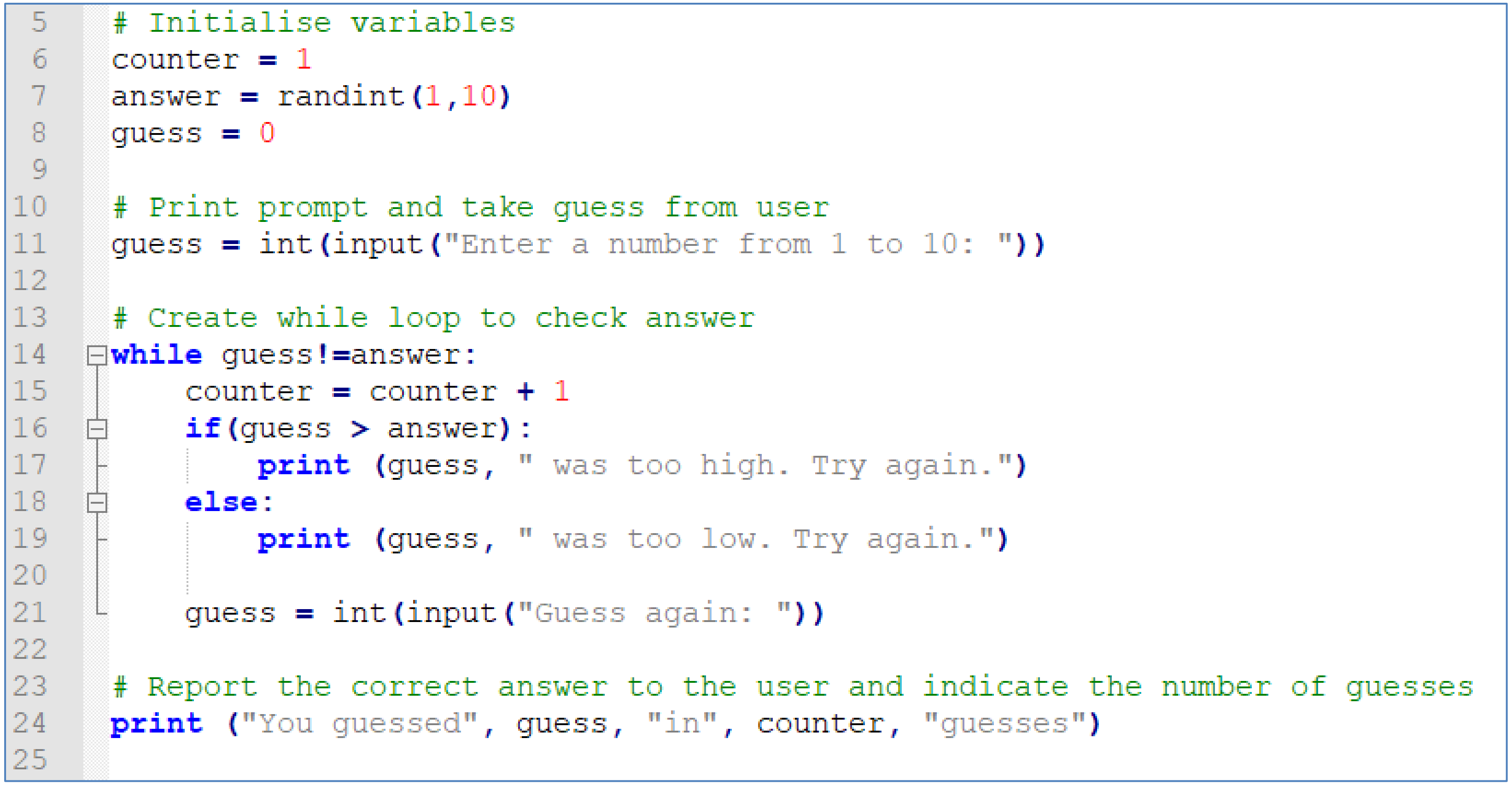
```
# Q01c

count = 0

maxValue = 5
constantValue = 7

while (count < maxValue):
    print(count, count + constantValue)
    count = count + 1

```

1(f)



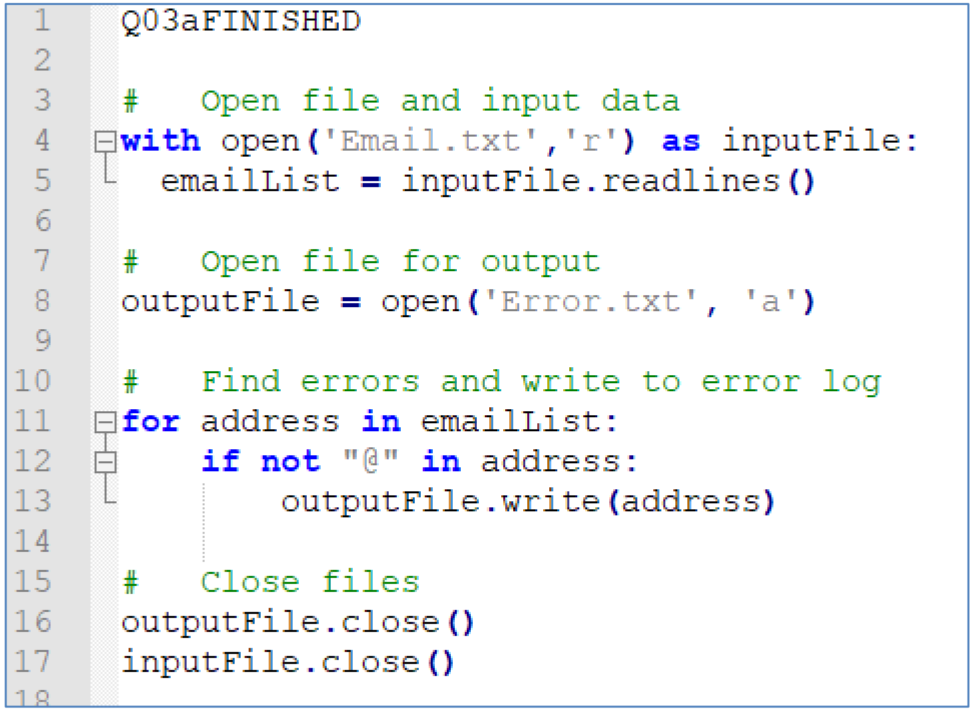
```
    if (shopIncome > 5000) or (assistantSales >= 10):
        print ("Assistant bonus = ", assistantSalary * 0.1)

    elif (shopIncome >= 2000) and (assistantSales >= 5):
        print("Assistant bonus = ", assistantSalary * 0.05)
```

2(a)



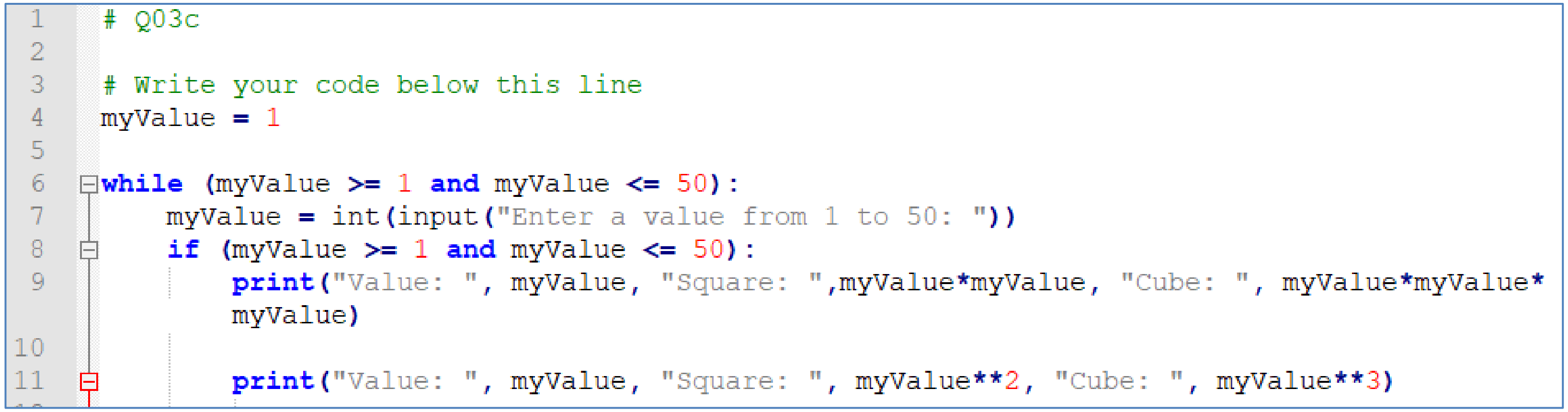
```python
# Initialise variables
counter = 1
answer = randint(1,10)
guess = 0

# Print prompt and take guess from user
guess = int(input("Enter a number from 1 to 10: "))

# Create while loop to check answer
while guess!=answer:
    counter = counter + 1
    if(guess > answer):
        print (guess, " was too high. Try again.")
    else:
        print (guess, " was too low. Try again.")

    guess = int(input("Guess again: "))

# Report the correct answer to the user and indicate the number of guesses
print ("You guessed", guess, "in", counter, "guesses")

```

3(a)



```
Q03aFINISHED

#   Open file and input data
with open('Email.txt','r') as inputFile:
  emailList = inputFile.readlines()

#   Open file for output
outputFile = open('Error.txt', 'a')

#   Find errors and write to error log
for address in emailList:
    if not "@" in address:
        outputFile.write(address)

#   Close files
outputFile.close()
inputFile.close()
```

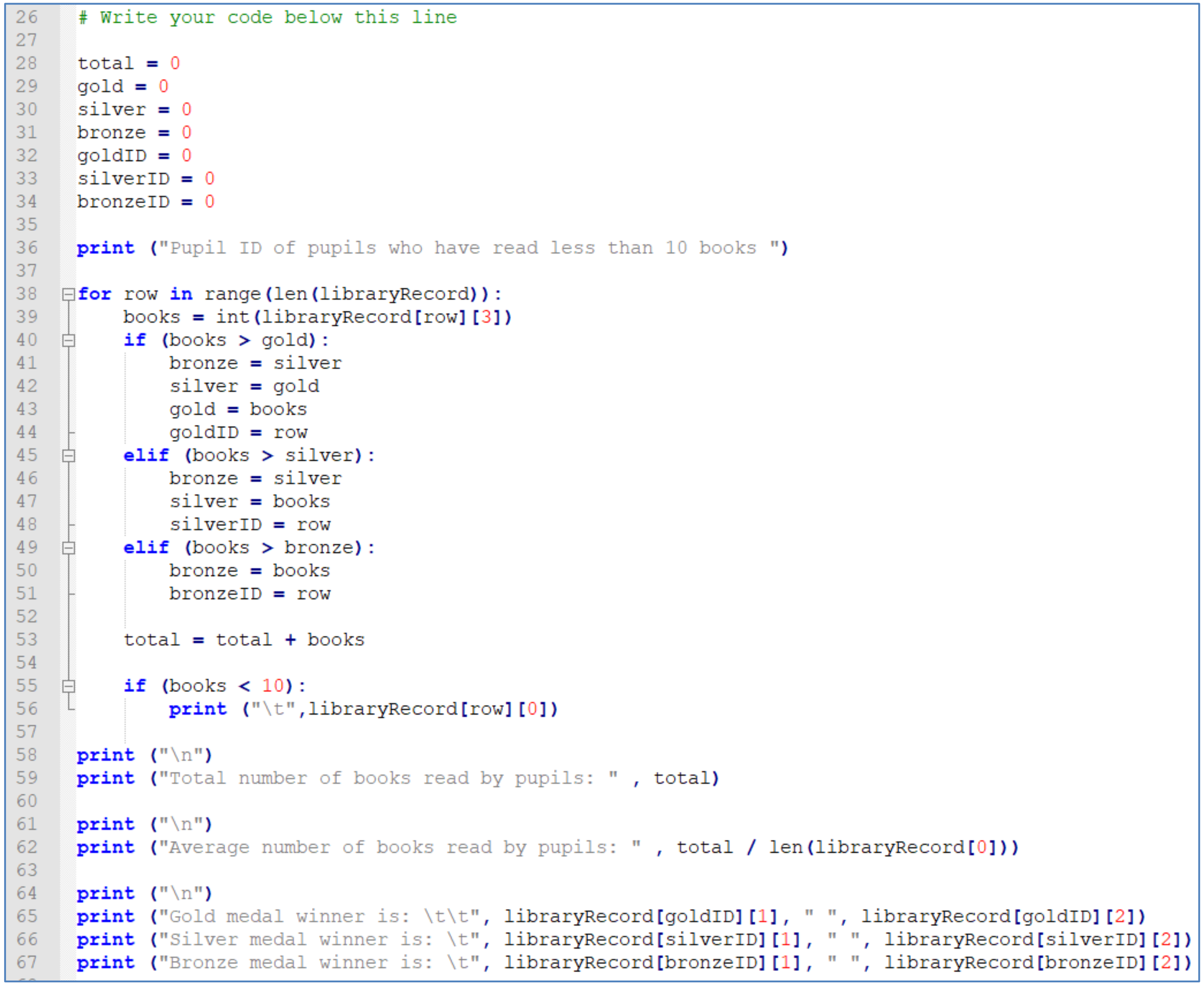
3(c)



```
1   # Q03c
2
3   # Write your code below this line
4   myValue = 1
5
6   while (myValue >= 1 and myValue <= 50):
7       myValue = int(input("Enter a value from 1 to 50: "))
8       if (myValue >= 1 and myValue <= 50):
9           print("Value: ", myValue, "Square: ",myValue*myValue, "Cube: ", myValue*myValue*
            myValue)
10
11          print("Value: ", myValue, "Square: ", myValue**2, "Cube: ", myValue**3)
```

Note: Line 11 is an alternative response to line 9.

5



```python
# Write your code below this line

total = 0
gold = 0
silver = 0
bronze = 0
goldID = 0
silverID = 0
bronzeID = 0

print ("Pupil ID of pupils who have read less than 10 books ")

for row in range(len(libraryRecord)):
    books = int(libraryRecord[row][3])
    if (books > gold):
        bronze = silver
        silver = gold
        gold = books
        goldID = row
    elif (books > silver):
        bronze = silver
        silver = books
        silverID = row
    elif (books > bronze):
        bronze = books
        bronzeID = row

    total = total + books

    if (books < 10):
        print ("\t",libraryRecord[row][0])

print ("\n")
print ("Total number of books read by pupils: " , total)

print ("\n")
print ("Average number of books read by pupils: " , total / len(libraryRecord[0]))

print ("\n")
print ("Gold medal winner is: \t\t", libraryRecord[goldID][1], " ", libraryRecord[goldID][2])
print ("Silver medal winner is: \t", libraryRecord[silverID][1], " ", libraryRecord[silverID][2])
print ("Bronze medal winner is: \t", libraryRecord[bronzeID][1], " ", libraryRecord[bronzeID][2])
```

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