Edexcel International GCSE O Level Computer Science

PAST PAPER 2 – PYTHON PRACTICAL

Dr. Saw Myat Sandar

Table of Contents

Find Errors

2017

(ii) Open the file **Q01c** in the code editor.
Amend the code to correct three program errors.
Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language. **(3)**

```
myNumbers = [10, 20, 30, 40 ,50, 60, 70, 80, 90, 100]
for theNumber in myNumbers:
   total = total + theNumber
   if(theNumber % 2 = 0):
      print("Even")
   else:
   print("Odd")
print(total)
```

```
1  myNumbers = [10, 20, 30, 40 ,50, 60, 70, 80, 90, 100]
2  total = 0
3  for theNumber in myNumbers:
4      total = total + theNumber
5      if(theNumber % 2 == 0):
6          print("Even")
7      else:
8          print("Odd")
9  print(total)
```

2019 May Jun

(c) Open **Q01c** in the code editor.
The program should print out a counter and the counter with 7 added to it.
There are **three** errors in the code.
Amend the code to correct the errors.
Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language. **(3)**

```
# Q01c
count = 0
maxValue = 5
while (count < maxValue)
   print(count, Count + constantValue)
   count = count + 1
```

```
1   # Q01c
2
3   count = 0
4
5   maxValue = 5
6   constantValue = 7
7
8   while (count < maxValue):
9       print(count, count + constantValue)
10      count = count + 1
```

2020 May Jun

(c) Open **Q01c** in the code editor.
The program should multiply two integers.
There are **three** errors in the code.
Amend the code to correct the errors.
Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language.
(3)

#Q01c
numOne = 25
num Two = 36
numThree = numone * numTwo
display numThree

```
numOne = 25
numTwo = 36
numThree = numOne * numTwo
print (numThree)
```

2021 May Jun

(c) Open **Q01c** in the code editor.
The program should calculate how many £20 are in £113 and how much is left over.
There are **three** errors in the code.
Amend the code to correct the errors.
Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language.
(3)

# Q01c
# Initialise variables
amount = 113
num_twenties == amount // 20
left_over = amount % 20

# Display results
print("There are "+ str(num_twenties) + " £20s in £113")
print("There is £" str(left over) + " left over")

Question	mp	Answer
1 (c)	C1	Change num_twenties == to num_twenties = (1)
	C2	The **left over** variable named the same in both places (1)
	C3	Add + before str(left_over) over in final print statement (1)

2022 May Jun

(f) Open **Q01f** in the code editor.
The program should allow the input of the length of the side of a square and output the area of the square.
There are **three** errors in the code.
Amend the code to correct the errors.
Save your amended code as **Q01fFINISHED** with the correct file extension for the programming language.
(3)

# Q01f
length = float(input("Enter the length of a side ")
area == length * length
print("The area of the square is,area")

```
3   length = float(input("Enter the first number "))
4
5   area = length * length
6
7   print("The area of the square is" , area)
```

2023 May Jun

(ii) The kitchen staff use a program to determine whether they have enough chips or how many more they need to order.

Open **Q02bii** in the code editor. There are **four** errors in the code.

Amend the code to correct the errors. Use this test data to help you find the errors.

Chips in stock (kilograms)	Number of adults	Number of children	Expected output
19	100	150	Stocks available
15	100	150	Order 4.0 kilograms

Save your amended code as **Q02biiFINISHED** with the correct file extension for the programming language.

(4)

```
#Q02bii
# Weight in grams for each portion of chips
PORTION_ADULT = 85
PORTION_CHILD = 70
# Get the total weight of chips in stock, in kilograms
inStock = int (input ("Enter the weight of chips available in kilograms: "))
inStock = 1000 * inStock
# Get the number of adults and the number of children
numAdults = int (input ("Enter the number of adult tickets sold: "))
numChildren = int (input ("Enter the number of child tickets sold: "))
# Calculate weight of adult and child portions
weightAdult = numAdults * PORTION_ADULT
weightChild = numChildren * PORTION_CHILD
# Calculate total weight required
requiredWeight = weightAdult =+= weightChild
# Tell staff the weight available and the weight required
print ("Available: ", inStock, "Required: ", requiredWeight)
# Check if available weight is enough for the dinner
if (requiredWeight > inStock):
   print ("Stocks available")
else:
   # Tell the staff how much more to order in kilograms
   weightNeed = (inStock - requiredWeight) / 0
   print ("Order", weightNeed, "kilograms.")
```

2(b)(ii)	Award **one** mark for each of:	
	B4	Change =+= to +
	B5	Change > to <=
	B6	Change order of subtraction to: requiredWeight - inStock
	B7	Change 0 to 1000

2024 May Jun

(b) Open **Q01b** in the code editor.
A user enters a name and an age when the program executes.
The program should display a welcome message when the user is less than 30 years of age.
Amend the code to complete the program.
Save your amended code as **Q01bFINISHED** with the correct file extension for the programming language.

```
# Q01b

# 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

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"
```

```
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
```

Use the code to answer these questions.

2019 May Jun

(b) Anna has written a program to help Graham with calculations for his science lessons.

Open **Q02b** in the code editor.

Use the code to answer these questions.

(i) State the value of **distance** when the program finishes.

(1)

.....

(ii) Give **one** reason why there are brackets around **newSpeed + initialSpeed**

(1)

.....

(iii) State the data type for the variable **distance**.

Justify your answer.

(2)

Data type

.....

Justification

.....

(iv) State the value of **distance** when **acceleration** is 2.75

(1)

.....

```
1  # Q02b
2
3  time = 0
4  initialSpeed = 5
5  acceleration = 2
6  newSpeed = 0
7  distance = 0
8
9  while (time < 4):
10     newSpeed = initialSpeed + time * acceleration
11     distance = 0.5 * time * (newSpeed + initialSpeed)
12     time = time + 1
13     print(distance)
```

2(b)(i)	24.0 (1)
2(b)(ii)	One of: By the rules of precedence multiplication is done before addition. (1)
2(b)(iii)	Real (1) Because real x integer = real (1)
2(b)(iv)	27.375 (1)

(c) Anna has written a program that manipulates text.
Open **Q02c** in the code editor.
Use the code to answer these questions.
(i) Identify a line number where the code includes a relational operator.
(1)

(ii) Identify a line number where a variable is set to a numeric value entered by a user.
(1)

(iii) Identify the name of a variable passed from the main program to the subprogram.
(1)

(iv) Identify the name of a variable that receives the value calculated by the subprogram.
(1)

(v) State the value returned by the subprogram when the input values are **educational**, **3**, **6**
(2)

(vi) State the purpose of the subprogram.
(2)

```
1  # Q02c
2
3  def amendMessage(inMessage,inValue1,inValue2):#educational,3,6
4      outMessage = inMessage[inValue1:inValue1+inValue2]#educational-[3:9]=cation
5      return outMessage
6
7  check = 0
8  while check == 0:
9      message = str(input("Enter a phrase 'end' to finish: "))
10     if (message == "end"):
11         check = 1
12         print("Goodbye")
13     else:
14         startPos = int(input("Start position: "))
15         characters = int(input("Number of characters: "))
16         newMessage = amendMessage(message,startPos,characters)
17         print(newMessage)
18         print("Thank you")
```

<table>
<tr><td>Q2(c)(i)</td><td><table><tr><td>C#:</td><td>16 or 20 (1)</td></tr><tr><td>Java:</td><td>16 or 21 (1)</td></tr><tr><td>Python:</td><td>8 or 10 (1)</td></tr></table></td></tr>
<tr><td>Q2(c)(ii)</td><td><table><tr><td>C#:</td><td>29 or 31 (1)</td></tr><tr><td>Java:</td><td>24 or 26 (1)</td></tr><tr><td>Python:</td><td>14 or 15 (1)</td></tr></table></td></tr>
<tr><td>Q2(c)(iii)</td><td>Any of:
1. Message (1)
2. startPos (Python) / pos (Java or C#) (1)
3. characters (1)</td></tr>
<tr><td>Q2(c)(iv)</td><td>newMessage (1)</td></tr>
<tr><td>Q2(c)(v)</td><td>Response starts with ‘c’ (1)
Response has six letters in sequence from ‘educational’ (1)</td></tr>
<tr><td>Q2(c)(vi)</td><td>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)</td></tr>
</table>

2020 May Jun

(e) Ayesha has written a program that calculates the sum of a set of numbers.
Open **Q01e** in the code editor.
Answer these questions about the code.
(i) Identify the line number where the code includes a logic operator.
(1)
(ii) Identify the line number where a subprogram definition starts.
(1)
(iii) Identify the name of a numeric variable.
(1)
(iv) Identify the name of a string variable.
(1)

```
1  #   Q01e
2
3  def sumOfNumbers(listNumbers):
4      amount = 0
5      for x in listNumbers:
6          if (x > 22 and x <= 40):
7              amount = amount + x
8      return (amount)
9
10
11 numArray = [23, 22, 41, 26, 55, 27, 34, 45, 24, 40, 25, 29, 51, 28, 55]
12
13 message = "Sum of selected numbers is "
14 total = sumOfNumbers(numArray)
15 print (message,total)
```

Question	mp	Answer
1 (e) (i)	E1	Python 6 C# 12 Java 8
1 (e) (ii)	E2	Python 3 C# 7 Java 3
1 (e) (iii)	E3	Award one mark for any of: • amount (1) • x (1) • total (1) • count (1)
1 (e) (iv)	E4	message

2021 May Jun

(d) Raoul has written a program that counts the number of vowels in a sentence.
Open **Q01d** in the code editor.
Amend the code by adding a suitable comment (you may need to add more than one comment to a line):
(i) at the end of the line where there is a **relational operator**
(1)
(ii) at the end of a line where **iteration** starts
(1)
(iii) at the end of the line where **selection** starts
(1)
(iv) at the end of a line where a **data structure** is initialised.
(1)
Save your amended code as **Q01dFINISHED** with the correct file extension for the programming language.

```
# Q01d

vowels = ["a","e","i","o","u"]
numVowels = [0,0,0,0,0]

sentence = input("Input the sentence ")

for letter in sentence:
    for vowel in vowels:
        if vowel == letter:
            numVowels[vowels.index(vowel)] +=1

print("Here are the number of vowels in the sentence "+ sentence)
for vowel in vowels:
    print("The number of",vowel,"is",numVowels[vowels.index(vowel)])

```

Question	mp	Answer
1 (d) (i)	D1	Award 1 mark for adding an appropriate comment at the end of the line where there is relational operator: `if vowel == letter:  # relational operator and selection`

1 (d) (ii)	D2	Award one mark for adding an appropriate comment at the end of a line where iteration starts:
		`for letter in sentence:  # iteration starts`
		`for vowel in vowels: # iteration starts`
		`for vowel in vowels:  # iteration starts`

1 (d) (iii)	D3	Award one mark for adding an appropriate comment at the end of the line where selection starts: `if vowel == letter:  # relational operator and selection`

1 (d) (iv)	D4	Award one mark for adding an appropriate comment at the end of a line where a data structure is initialised:
		`vowels = ["a","e","i","o","u"] # data structure initalised`
		`numVowels = [0,0,0,0,0] # data structure initalised`

2022 May Jun

2 Raza is writing a program to tell the user whether a number they input is a prime number.

A prime number is a whole number, larger than one, that can only be divided by one and itself with no remainder.

This pseudocode contains the logic required to complete the program.

```
FUNCTION checkPrime(pNumber)
BEGIN FUNCTION
    IF pNumber = 1 THEN
        check = False
    ELSE
        check = True
        FOR count FROM 2 TO pNumber DO
            IF pNumber MOD count = 0 THEN
                check = False
            END IF
        END FOR
    END IF
RETURN check
END FUNCTION

SEND "Enter a number: " TO DISPLAY
RECEIVE number FROM (INTEGER) KEYBOARD
SET result TO checkPrime(number)

IF result = True THEN
    SEND (number & " is a prime number") TO DISPLAY
ELSE
    SEND (number & " is not a prime number") TO DISPLAY
END IF
```

(a) Answer these questions about the pseudocode.

(i) Identify a line number where **selection** starts.

(1)

.....

(ii) Identify the line number where **iteration** starts.

(1)

.....

(iii) Identify a line number where a **string** and a **number** are printed on the same line.

(1)

.....

(iv) Identify the name of a **local variable**.

(1)

.....

(v) Identify the name of a **parameter**.

(1)

Question	mp	
2 (a)(i)	A1	3/8/20
2 (a)(ii)	A2	7
2 (a) (iii)	A3	21/23
2 (a)(iv)	A4	check/count/pNumber
2 (a)(v)	A5	pNumber

2023 May Jun

(b) A tax rate is applied to a gross value to give a net value.

Open **Q01b** in the code editor.

Use the code to answer these questions.

(i) Give the contents of a comment.

(1)

.....

(ii) Identify a logical operator used in this program.

(1)

.....

(iii) Give the name of a global variable.

(1)

.....

(iv) Give the name of a constant.

(1)

.....

(v) Give the name of a parameter.

(1)

.....

```
1
2  TAX = 0.05       # This is 5%
3
4  grossAmt = 0.0
5  netAmt = 0.0
6
7  def calcResult (inGross):
8      net = inGross * (1 - TAX)
9      return (net)
10
11 grossAmt = float (input ("Enter the gross amount: "))
12 if (grossAmt < 10.0) or (grossAmt > 200.0):
13     print ("Invalid input")
14 else:
15     netAmt = calcResult (grossAmt)
16     print ("Your net amount is", netAmt)
```

1(b)(i)	B1	This is 5%
1(b)(ii)	B2	or Accept || for C# and Java
1(b)(iii)	B3	grossAmt netAmt
1(b)(iv)	B4	TAX
1(b)(v)	B5	inGross

2024 May Jun

(c) Open **Q01c** in the code editor.

Use the code to answer these questions.

(i) Give the text of a comment used in the code.

(1)

(ii) Give the text of a line that creates and initialises a variable.

(1)

(iii) Give the keyword that starts the selection used in the code.

(1)

(iv) Give the logical operator used in the code.

(1)

Pseudocode to Python Code

2017

2 A football club uses computer applications.

(a) The club collects this data about visitors:

• country of origin

• number of adults and children in each party.

This pseudocode contains the logic required to do this.

```
# Print prompt and take country from user
SEND "Enter the country you're visiting from: " TO DISPLAY
RECEIVE country FROM (STRING) KEYBOARD

# Tell the user their country
SEND ("You are from: " & country) TO DISPLAY

# Take number of adults in party
SEND "Enter the number of adults in your party: " TO DISPLAY
RECEIVE adults FROM (INTEGER) KEYBOARD

# Take number of children in party
SEND "Enter the number of children in your party: " TO DISPLAY
RECEIVE children FROM (INTEGER) KEYBOARD

# Calculate total number in party
SET total TO adults + children

# Tell user the total
SEND ("The total in your party is: " & total)

```

```
# Print prompt and take country from user
country = input ("Enter the country you're visiting from: ")

# Tell the user their country
print ("You are from: ", country)

# Take number of adults in party from user
adult = int (input ("Enter the number of adults in your party: "))

# Take number of children in party from user
children = int (input ("Enter the number of children in your party: "))

# Calculate total number in party
total = adult + children

# Tell the user the total number of people in their party
print ("The total in your party is: ", total)
```

2019 May Jun

2 Anna has a 5-year-old daughter, Beatrice, and a 14-year-old son, Graham.
(a) Anna wants to program a guessing game for Beatrice to play on the computer.
The program will generate a random number between 1 and 10.
Beatrice has to guess the number.
This pseudocode contains the logic required to create the game.

```
1   #   Initialise variables
2       SET counter TO 1
3       SET answer TO RANDOM(9) + 1     # i.e. a random integer 1 to 10 inclusive
4       SET guess TO 0
5
6   #   Print prompt and take guess from user
7       SEND "Enter a number from 1 to 10: " TO DISPLAY
8       RECEIVE guess FROM (INTEGER) KEYBOARD
9
10  #   Create while loop to check guess
11      WHILE guess <> answer DO
12          SET counter TO counter + 1
13          IF guess > answer THEN
14              SEND (guess & " was too high. Try again.") TO DISPLAY
15          ELSE
16              SEND (guess & " was too low. Try again.") TO DISPLAY
17          END IF
18          SEND "Guess again: " TO DISPLAY
19          RECEIVE guess FROM (INTEGER) KEYBOARD
20      END WHILE
21
22  #   Report the correct answer to the user and display the number of guesses
23      SEND ("You guessed " & guess & " in " & counter & "guesses.") TO DISPLAY
```

Write a program to implement the logic in the pseudocode.
Open **Q02a** in the code editor**.**
You **must** use the structure give in **Q02a** to write the program.
Do not add any further functionality.
Save your code as **Q02aFINISHED** with the correct file extension for the programming language.
(10)

```
# Q02a
from random import *
#  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 if guess is corrwect
while guess!=answer:
   counter=counter+1
   if(guess>answer):
      print(guess,"was too high. Try again")
   else:
      print(guess,"was too low.Try agian")
   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.")
```

2020 May Jun

2 Rama wants some computer programs for his daughter, Ayomi.
(a) Rama wants a program to calculate the area of a triangle.
(area of a triangle = ½ base x height)
This pseudocode contains the logic required to create the program.

```
1  #  Initialise variables
2
3  SET base TO 50
4  SET heightChk TO TRUE
5
6  #  Print prompt and take input from user
7
8  WHILE heightChk DO
9      SEND "Enter the height (between 1 and 100): " TO DISPLAY
10     RECEIVE height FROM (INTEGER) KEYBOARD
11     IF (height >= 1 AND height <= 100)
12     THEN heightChk = FALSE
13     END IF
14 END WHILE
15
16 #  Calculate and print out values
17
18 SET area TO 0.5 * base * height
19
20 SEND ("Base of triangle is: " , base) TO DISPLAY
21 SEND ("Height of triangle is: " , height) TO DISPLAY
22 SEND ("Area of triangle is: " , area) TO DISPLAY
```

Write a program to implement the logic in the pseudocode.
Open Q02a in the code editor.
You must use the structure given in Q02a to write the program.
Do not add any further functionality.
Save your code as Q02aFINISHED with the correct file extension for the programming language.
(10)

```
1  #   Q2a
2
3  #   Initialise variables
4
5
6
7  #   Print prompt and take input from user
8
9
10
11 #   Calculate area and print out values
```

```
#       Q2a

#       Initialise variables

base=50
heightChk=True

#       Print prompt and take input from user
while heightChk:
   height=int(input("Enter the height between 1 and 100"))
   if height>=1 and height<=100:
      heightChk=False

#       Calculate area and print out values

area=0.5*base*height

print("Base of triangle is:", base)
print("Height of triangle is:", height)
print("Area of triangle is:", area)
```

2022 May Jun

2 Raza is writing a program to tell the user whether a number they input is a prime number.
A prime number is a whole number, larger than one, that can only be divided by one and itself with no remainder.
This pseudocode contains the logic required to complete the program.

```
FUNCTION checkPrime(pNumber)
BEGIN FUNCTION
    IF pNumber = 1 THEN
        check = False
    ELSE
        check = True
        FOR count FROM 2 TO pNumber DO
            IF pNumber MOD count = 0 THEN
                check = False
            END IF
        END FOR
    END IF
RETURN check
END FUNCTION

SEND "Enter a number: " TO DISPLAY
RECEIVE number FROM (INTEGER) KEYBOARD
SET result TO checkPrime(number)

IF result = True THEN
    SEND (number & " is a prime number") TO DISPLAY
ELSE
    SEND (number & " is not a prime number") TO DISPLAY
END IF
```

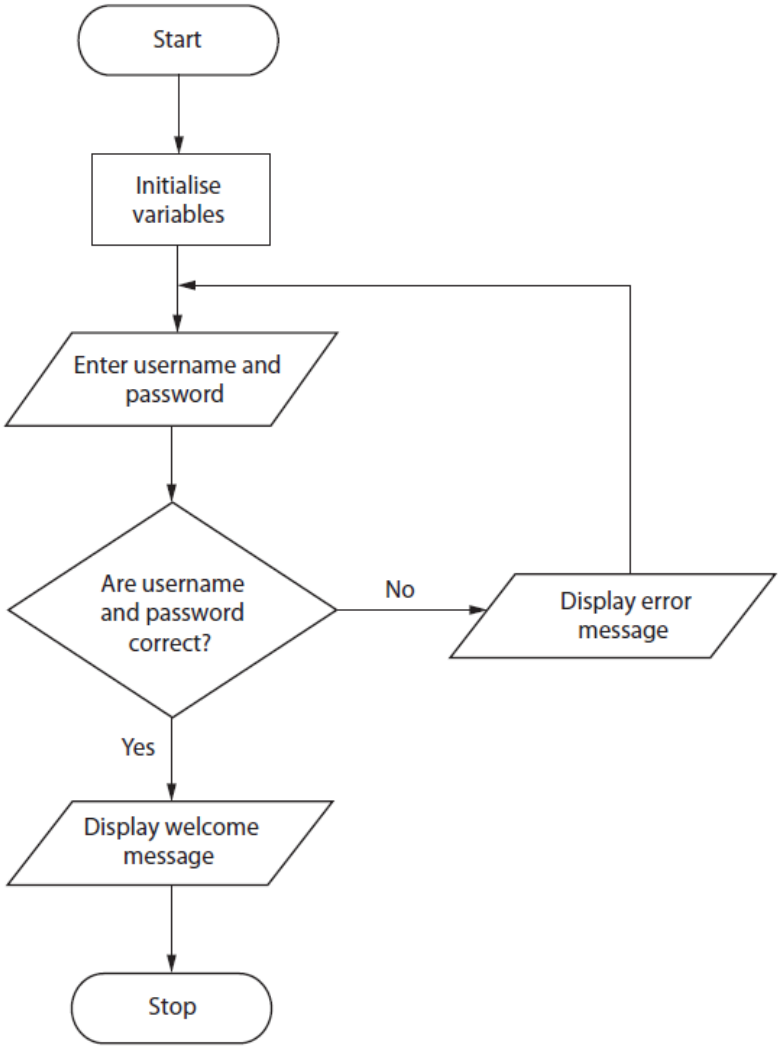
(b) Write a program to implement the logic in the pseudocode.
Open Q02b in the code editor.
Write the program.
You must use the structure given in Q02b to complete the program.
Do not add any further functionality.
Save your code as Q02bFINISHED with the correct file extension for the programming language.
(11)

```
# Q02b

# Write the function here
def checkPrime(pNumber):
    if pNumber==1:
        check=False
    else:
        check=True
        for count in range(2,pNumber):
            if pNumber%count==0:
                check=False
    return check

# Write the main program here
number=int(input("Enter a number: "))
result=checkPrime(number)
if result==True:
    print(number," is a prime number")
else:
    print(number, "is not a prime number")
```

Flowchart to Python Code

2021 May Jun

2 A book club uses computer applications.
(a) The club wants a program to ensure that logins are valid.
A flowchart for a test version of the program has been designed.
The test version uses:
• a username of bard423
• a password of nX2934?
Here is the flowchart.



Write a program to implement the logic in the flowchart.
Open Q02a in the code editor.
You must use the structure given in Q02a to write the program.
Do not add any further functionality.
Save your code as Q02aFINISHED with the correct file extension for the programming language. (11)

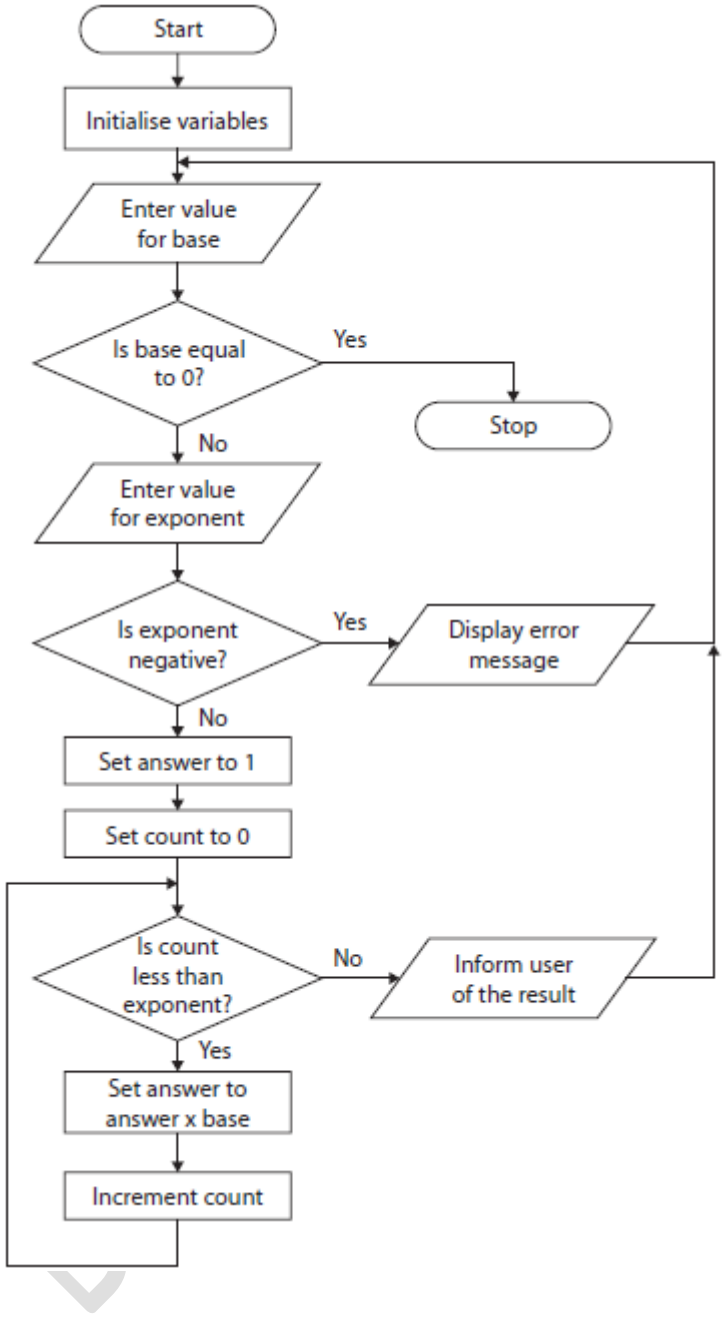
```
# Initialise variables
username = "bard423"
password = "nX2934?"
count = 0
inputUsername = ""
inputPassword = ""

# Print promts, take and check input from user
while inputUsername != username or inputPassword != password:
    if count > 0:
        print("There is a problem with the login details. Try again")
    count = count + 1
    inputUsername = input("Enter your username ")
    inputPassword = input("Enter your password ")
print("Welcome")
```

```
# Q02a

# Initialise variables

username="bard423"
password="nX2934?"
count=0
inputUsername=""
inputPassword=""

# Print prompts, take and check input from user

while inputUsername!=username or inputPassword!=password:
   if count>0:
      print("Try again")
   count=count+1

   inputUsername=input("Enter user name:")
   inputPassword=input("Enter password:")
print("Welcome")
```

2023 May Jun

(c) A program is required to calculate the result of raising one integer (the base) to the power of another (the exponent).

Figure 3 shows a flowchart for the algorithm.



The program has these requirements:

- outputs meaningful error messages
- outputs the final answer with the base and the exponent.

Open Q03c in the code editor.

Write a program to implement the logic in the flowchart.

Do not add any further functionality.

Save your code as Q03cFINISHED with the correct file extension for the programming language.

(10)

```
# Q03c

# ------------------------------------------------------
# Write your code below this line

base=int(input("Enter base number:"))
while (base!=0):
   exponent=int(input("Enter exponent:"))
   if (exponent<0):
      print("Invalid")
   else:
      answer=1
      count=0

      for count in range(0,exponent):
         answer=answer*base
      print(answer)

   base=int(input("Enter base number (0 exist):"))
```

Comparison IF Statement

2017

(c) The football club makes money selling tickets, food and other items. Attendance at matches is also monitored.
At the end of each week, the data collected is processed.
Attendance is considered along with income (money coming in).
Open the file **Q02c** in the code editor.
Amend the code to complete the 'If statement' used to produce the outputs described in the table. Do not add any further functionality.
Save your amended code as **Q02cFINISHED** with the correct file extension for the programming language. (4)

Condition	Output message
Attendance is at least 1500	Sufficient profit made this week
Income is at least 45000	Sufficient profit made this week
Attendance is at least 750; income is at least 22500	Income in line with attendance this week
Attendance is fewer than 500	Attendance is very low this week Contact fan club
All other inputs	Possible accounting error

```
if (attendance >= 1500) or (income >= 45000):
    print ("Sufficient profit made this week")
elif (attendance >= 750) and (income >= 22500):
    print ("income in line with attendance this week")
elif (attendance < 500):
    print ("Attendance is very low this week.  Contact the fan club.")
else:
    print ("Possible accounting error.")
```

2019 May Jun

(f) Trevor is the manager of a shoe shop.
Sales assistants can earn a bonus based on the numbers of pairs of shoes they sell and the total income for the shop each day.
Open **Q01f** in the code editor.
Amend the code to complete the 'if statement' used to produce the outputs described in the table.

Condition	Output
Shop income is more than £5000 or sales assistant has sold at least 10 pairs of shoes	Bonus is 10% of salary
Shop income is £2000 or more and sales assistant has sold at least 5 pairs of shoes	Bonus is 5% of salary

Do not add any further functionality.
Save your amended code as **Q01fFINISHED** with the correct file extension for the programming language.
(4)

```
# Q01f

list_1f =[(2010,8), (4800, 11), (2011,4), (5000,9)]
assistantSalary = 1000.00

for pair in list_1f:
   shopIncome = pair[0]
   assistantSales = pair[1]
   print ("Shop income :", shopIncome, "Assistant sales ", assistantSales)

   if (    ):
      print ("Assistant bonus = ", assistantSalary * 0.1)

   elif (   ):
      print("Assistant bonus = ", assistantSalary * 0.05)

   else:
      print("Assistant bonus = ", 0)
```

```
11      if (shopIncome > 5000) or (assistantSales >= 10):
12          print ("Assistant bonus = ", assistantSalary * 0.1)
13
14      elif (shopIncome >= 2000) and (assistantSales >= 5):
15              print("Assistant bonus = ", assistantSalary * 0.05)
```

2020 May Jun

4 Gianluca is a schoolteacher. He is in charge of a year group of pupils.
(a) He has started to create a program to analyse test results.
Open **Q04a** in the code editor.
Amend the code to complete the 'if statement' used to produce the outputs described in this table.

Condition	Output text
English test score was less than 40; Maths test score was less than 50	Student failed both tests
English test score was less than 40 or Maths test score was less than 50	Student failed one test
English test score was at least 80; Maths test score was at least 85	Student passed both tests with distinction
Other conditions	Student passed both tests

Do not add any further functionality.
Save your code as **Q04aFINISHED** with the correct file extension for the programming language.
(6)

```
#  Q4aFINISHED

#  Initialise variables
#  Data format: First name, Last name, English test mark, Maths test mark

studentTestScores = [
["Kevin", "Horney", 71, 55],
["Tony", "Tison", 36, 50],
["David", "Smith", 39, 48],
["Vicky", "Bertwistle", 58, 71],
["Jason", "Blashaw", 49, 80],
["Louise", "Smith", 81, 67],
["Sara", "Acton", 37, 66],
["Donna", "Alexander", 84, 86],
["Michael", "Mitchelle", 65, 55],
["Anthony", "Lewis", 48, 50],
["Sharon", "Grant", 53, 70],
["Peter", "Hughes", 82, 90],
["Deborah", "Biddle", 51, 47],
["Dawn", "Doens", 35, 44],
["William", "Dennis", 72, 73],
["Selim", "Biddle", 52, 67],
```

```
["Ian", "Nadeem", 40, 36],
["Jenny", "Thomson", 56, 43],
["Rowena", "Moore", 50, 77],
["Jane", "Murphy", 44, 48]]

message = [
"student failed both tests",
"student failed one test",
"student passed both with distinction",
"student passed both tests"]

for student in studentTestScores:
    first = student[0]
    last = student[1]
    english = student[2]
    maths = student[3]

    print(first, last, ": ")
    if (  ):     # student failed both
        messageIndex = 0
    elif (  ):     # student failed one
        messageIndex = 1
    elif (  ):   # student passed both with distinction
        messageIndex = 2
    else:

    print(message[messageIndex])
```

```
34 for student in studentTestScores:
35     first = student[0]
36     last = student[1]
37     english = student[2]
38     maths = student[3]
39 
40     print(first, last, ": ")
41     if (english<40 and maths<50):      # student failed both
42         messageIndex = 0
43     elif (english<40 or maths<50  ):     # student failed one
44         messageIndex = 1
45     elif (english>=80 and maths>=85  ):   # student passed both with distinction
46         messageIndex = 2
47     else:
48         messageIndex = 3
49     print(message[messageIndex])
```

2021 May Jun

(b) The club sells books.
It needs a computer program to monitor:
• the number of books sold
• the amount of profit made.
Q02b provides a structure for the program.
Open **Q02b** in the code editor.
Amend the code to complete the If statement used to produce the outputs described in the table.

Condition	Output message
Number of books sold is at least 5 and profit made is at least 10	Sales and profit are good this week
Number of books sold is over 20 and profit made is at least 20	Sales and profit are excellent this week
Number of books sold is under 5 or profit made is under 5	Poor performance this week
All other inputs	Alert manager

Do not add any further functionality.
Save your code as **Q02bFINISHED** with the correct file extension for the programming language.
(4)

```
# Q02(b)

# Initialise variables
booksSold = 0
profit = 0

booksSold = int(input("Enter the number of books sold: "))
profit = int(input("Enter the profit made: "))

# Add your code
if :

   print("Poor performance this week")
elif :
   print("Sales and profit are excellent this week")
elif :
   print("Sales and profit are good this week")
else:
```

```
if booksSold < 5 or profit < 5:
    print("Poor performance this week")
elif booksSold > 20 and profit >= 20:
    print("Sales and profit are excellent this week")
elif booksSold >=5 and profit >= 10:
    print("Sales and profit are good this week")
else:
    print("Alert manager")
```


2024 May Jun

3(c) A program validates a number input by the user.
Figure 3 shows the output messages based on the inputted number.

Input	Output
<empty>	You must provide a number
Any negative number	The number must be greater than zero
0	The number must be greater than zero
1 to 20	Acceptable
60 or more	Acceptable
31 to 39	Centre
30	Perfect
Any other number	No message

Figure 3

Open **Q03c** in the code editor.
Amend the code to ensure the messages are generated correctly.
Do not add any further functionality.
Save your code as **Q03cFINISHED** with the correct file extension for the programming language.
(6)

```
# 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")
```

File Handling

2017

(c) The holiday company needs to assign numbers to their most popular destinations.

Cities.txt	Numbered.txt
London	1 London
Hong Kong	2 Hong Kong
Delhi	3 Delhi
Istanbul	4 Istanbul
Tokyo	5 Tokyo
Mumbai	6 Mumbai
Mexico City	7 Mexico City
New York City	8 New York City
Rio de Janeiro	9 Rio de Janeiro
Singapore	10 Singapore

The files **Q03c** and **Cities.txt** are provided. Open the code named **Q03c** in the code editor.
Write a program to implement these requirements.
For all lines in the **Cities.txt** file, the code must:
• read the line
• append a line number and a space to the front
• write the new line to a **Numbered.txt** file
• print the line to the display
You must use the structure given in file **Q03c** to complete the program.
Do not add further functionality. Save your amended code as **Q03cFINISHED** with the correct file extension for the programming language. **(7)**

```
# Open "Cities.txt" as input
theFile = open ("Cities.txt", "r")

# Open "Numbered.txt" as output
outFile = open ("Numbered.txt", "w")

# Use a for/each loop to read each line of
# the input file into a variable named 'theLine'
for theLine in theFile:

    # Increment the line count
    count = count + 1

    # Add the line number to the front of the line followed by a space
    theLine = str(count) + " " + theLine

    # print the line to the display
    print (theLine)

    # Write the new line to the output file
    outFile.writelines (theLine)

# Close the input file
theFile.close()

# Close the output file
outFile.close()
```

```
# Do not use any other data structure such as an array or a list.
count = 0            # A counter for the line numbers
theLine = ""         # Holds each line of the file

# Open "Cities.txt" as input
theFile = open ("Cities.txt", "r")

# Open "Numbered.txt" as output
outFile = open ("Numbered.txt", "w")

# Use a for/each loop to read each line of
# the input file into a variable named 'theLine'
for theLine in theFile:

   # Increment the line count
   count = count + 1

   # Add the line number to the front of the line followed by a space
   theLine = str(count) + " " + theLine

   # print the line to the display
   print (theLine)

   # Write the new line to the output file
   outFile.writelines (theLine)

# Close the input file
theFile.close()

# Close the output file
outFile.close()
```

2019 May Jun

3 Different algorithms can be used to manipulate data.
(a) **Email.txt** contains a list of email addresses.
Open **Q03a** in the code editor.
Write a program to implement these requirements.
The code must:
• check each email address to ensure it contains the '@' symbol.
• write email addresses that do not contain the '@' symbol to an **Error.txt** file.
You must use the structure given in the file **Q03a** to complete the program.
Do not add further functionality.
Save your code as **Q03aFINISHED** with the correct file extension for the programming language.
(6)

```
# Q03a

#   Open the file and input data

#   Open output file


#   Find errors and write to output file


#   Close files
```

```
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()
```

```
# Q03a

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

inputFile=open('Email.txt','r')
emailList=inputFile.readlines()

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

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

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

2021

3 Several encryption algorithms have been developed.
(a) Ahmed has started to create a Caesar cipher encryption program.
Caesar cipher encryption works by giving a number value to a key.
Each letter in a plaintext message is replaced by a new ciphertext letter using the key.
For example, as shown in the table, a key value of +2 would change the plaintext message **encrypt** to the ciphertext message **gpetarv**.

Plaintext	e	n	c	r	y	p	t
Ciphertext	g	p	e	t	a	r	v

Ahmed's program already converts from plaintext to ciphertext.
He wants to extend the program to:
• accept the input of a plaintext message
• accept the input of the key, which must be a positive integer between 1 and 25
• write the generated ciphertext to a text file named **Cipher.txt**
• display the ciphertext.
Open **Q03a** in the code editor.
Complete the program code.
You must use the structure given in the file **Q03a** to complete the program.
Do not add any further functionality.
Save your code as **Q03aFINISHED** with the correct file extension for the programming language.
(8)

```
# Q03(a)

# Initialise variables
alphabet = ["a","b","c","d","e","f","g","h","i","j",
            "k","l","m","n","o","p","q","r","s","t",
            "u","v","w","x","y","z"]

plaintext = ""
ciphertext = ""
key = 0
plaintextLength = 0
count = 0

# Add your code to get the plaintext and convert it to lowercase
plaintext=input("Enter your plain text")
plaintext=plaintext.lower()

# Add your code to get the key and make sure the key is between 1 and 25

while key<1 or key>25:
   key=int(input("Enter key"))
```

```
# Ciphertext is generated
plaintextLength = len(plaintext)


# Each plaintext letter is convereted into a ciphertext letter
while count <  plaintextLength:
   found = False
   alphabetCount = 0
   while alphabetCount < 26 and found == False:
      if alphabet[alphabetCount] == plaintext[count]:
         found = True
         if alphabetCount + key - 26 < 0:
            ciphertext += alphabet[alphabetCount + key]
         else:
            ciphertext += alphabet[alphabetCount + key - 26]
      alphabetCount = alphabetCount + 1
   count = count + 1

# Add your code to write the ciphertext to a text file
   cipherFile=open("Cipher.txt","w")
   cipherFile.write(ciphertext)
   cipherFile.close()

# Add your code to display the ciphertext

print("The cipher message is:",ciphertext)
```

2022

5 Julia runs a computer gaming club.
(a) She wants a program to check passwords stored in a file.
The file **passwords.txt** contains the list of passwords.
The program must:
• check each password to ensure that:
• the first character is an uppercase letter
• if it is, that it also includes at least one digit (0–9)
• if a password does not meet these requirements:
• display the password
• increment the number of incorrect passwords
• display the total number of incorrect passwords after all the passwords have been checked.
Open **Q05a** in the code editor.
Write the program.
You must use the structure given in **Q05a** to write the program.
Do not add any further functionality.
Save your code as **Q05aFINISHED** with the correct file extension for the programming language. **(9)**

```
# Q05(a)
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; # Valid uppercase letters in the
alphabet
digit = "0123456789"; # Valid digits
passwordFile = open("passwords.txt", "r") # Open the file password.txt to read
# Add your code here
incorrectPassword=0

for password in passwordFile:
   valid=False
   if password[0] in alphabet:
      index=1
      while valid==False and index in range(1,len(password)):
         if password[index] in digit:
            valid=True
         else:
            index=index+1

   if valid==False:
      incorrectPassword=incorrectPassword+1
      print(password)
print(incorrectPassword,"passwords are incorrect")
passwordFile.close()
```

2023

(c) The scientist is collecting and storing data about tuna.
The data collected is:
• species
• length in centimetres
• weight in kilograms
• age in years.
The data is stored in an array.
The collected data needs to be written to a file named **TunaData.txt**
Each record stored in the file must have a code number in the first field.
Code numbers must start at 101
Each field in a record should be separated by a comma.
Figure 7 shows the contents of the file.

```
101,Yellowfin,105,15,3
102,Albacore,90,15,5
103,Skipjack,50,3,4
104,Bigeye,105,25,4
105,Atlantic Bonito,50,4,2
106,Northern Bluefin,190,120,11
107,Southern Bluefin,190,120,11
108,Tongol,90,20,4
```

Figure 7

Open **Q05c** in the code editor.
Write the program.
You must use the structure given in **Q05c** to write the program.
Do not add any further functionality.
Save your code as **Q05cFINISHED** with the correct file extension for the programming language. **(6)**

```
# Q05c

tbl_tuna = [["Yellowfin",105,15,3],
        ["Albacore",90,15,5],
        ["Skipjack",50,3,4],
        ["Bigeye",105,25,4],
        ["Atlantic Bonito",50,4,2],
        ["Northern Bluefin",190,120,11],
        ["Southern Bluefin",190,120,11],
        ["Tongol",90,20,4]]
```

```
# ---------------------------------------------
# Write your code below this line
number=101
file=open("TunaData.txt","w")
for tuna in tbl_tuna:
   line=str(number)+","+tuna[0]+","+str(tuna[1])+","+str(tuna[2])+","+str(tuna[3])+"\n"
   file.write(line)
   number=number+1
file.close()
```

2024 May Jun

(c) **Figure 7** shows the **Sales.txt** file. It stores sales information.

```
264,140,120,284,192
420,377,435,376,392
619,589,606,586,600
799,811,788,814,788
982,1007,1013,989,1009
```

Figure 7

A program is required to calculate and display:

- a subtotal for each line of sales
- a grand total for all the sales in the file.

Figure 8 shows the intended output from the program.

```
1000
2000
3000
4000
5000
Grand total: 15000
```

Figure 8

Open **Q05c** in the code editor.
Amend the code to produce the intended output.
You must use the structure and variables given in **Q05c** to complete the program.
Do not add any further functionality.
Save your code as **Q05cFINISHED** with the correct file extension for the programming language.
(8)

```
# Q05c

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 =

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

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

    subTotal = 0
    # Add code to sum up each value in the set of five strings


    # Add code to display the subtotal for the line


    # Add code to calculate the running total


# Add code to display the total of all lines in the file

```

```
# Q05c

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 =

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

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

   subTotal = 0
   # Add code to sum up each value in the set of five strings
```

```
    # Add code to display the subtotal for the line


    # Add code to calculate the running total


# Add code to display the total of all lines in the file


# Complete the code to close the opened file
theFile.
```

```
# 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 ()

```

Produce Code

2021 May Jun

4 Zak sells snacks at an after-school club.
(a) He wants a program that will hold details of the products he sells.
The program must:
• generate a five-character product code that:
– starts with the first three letters of the product name
– ends with a random number between 10 and 30
• display the product code followed by the product name.
Open **Q04a** in the code editor.
Write the program.
You must use the structure given in the file **Q04a** to complete the program.
Do not add any further functionality.
Save your code as **Q04aFINISHED** with the correct file extension for the programming language.
(7)

```
# Q04(a)

import random

# Get input

# Generate a random number between 10 and 30 inclusive

# Generate the product code - first three letters of product name and the random number

# Display the product code and the product name
```

```
# Q04(a)

import random

# Get input
productName=input("Enter product name:")

# Generate a random number between 10 and 30 inclusive
randomNumber=random.randint(10,30)

# Generate the product code - first three letters of product name and the random number
productCode=productName[0:3]+str(randomNumber)

# Display the product code and the product name
print(productCode," ",productName)
```

(b) A program is needed to create a key for a database.
A user enters a two-letter word, a whole number and a decimal number.
The program must ensure the word is only two characters long.
The program must display an error message when the word is not two characters long.
A key is generated from the whole number, the reversed word and the whole number part of the decimal number.
Figure 4 shows the input values and a valid key.

```
Enter a word: at
Enter a whole number: 12
Enter a decimal number: 7.89
12ta7
```

Figure 4
Open **Q04b** in the code editor.
Write the program.
Do not add any further functionality.
Save your code as **Q04bFINISHED** with the correct file extension for the programming language.
(8)

```
# Q04b

# ------------------------------------------------------------
# Write your code below this line
myword=input("Enter a word: ")
if len(myword)==2:
    mynum=int(input("Enter a whole number:" ))
    mydec=float(input("Enter a decimal number: "))
    key=str(mynum)+myword[1]+myword[0]+str(int(mydec))
    print(key)
else:
    print("Error")
```

2024

(c) A program is required to create a new key.
The program takes two inputs.
The first input is a four-character string.
The second input is a whole number.
The key is constructed by joining the first two characters from the string, the number and the final two characters from the string.
When the user enters the four-character string **abcd** and the integer **123**, the program must construct and display the new key **ab123cd**
Open **Q04c** in the code editor.
Amend the code to:
• complete the subprogram to construct the new key
• complete the call to the subprogram.
Do not add any further functionality.
Save your code as **Q04cFINISHED** with the correct file extension for the programming language.
(6)

```
# 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)

```