



GCSE Computer Science: Exam Insights May/June 2024

Paper 2 Exemplars

Paper 2 – Q01

```
1 # -----
2 # Global variables
3 # -----
4 rainbow = ["Violet", "Indigo", "Blue", "Green", "Yellow", "Orange", "Red"]
5 waveTable = [380, 425, 450, 492, 577, 597, 622]
6 found = False
7 index = 0
8 wavelength = "0123"
9 colour = ""
10
11 # -----
12 # Main program
13 # -----
14 # User chooses a colour index
15 index = int(input("Enter an index: "))
16 if (index < 0):
17     print("Indexes cannot be zero")
18 elif (index > 6):
19     print("Indexes cannot be more than six")
20 else:
21     color = rainbow[index]
22     print(str(colour))
23
24 # User chooses a colour based on wavelength
25 wavelength = int(input("Enter a wavelength "))
26 if ((wavelength < 380) or (wavelength > 622)):
27     print("Invalid wavelength")
28 else:
29     index = 0
30     # Look for a wavelength less than or equal to user's choice
31     while (not found):
32         if (wavelength == waveTable[index]):
33             found = True
34             print(rainbow[index])
35         elif (waveTable[index] >= wavelength):
36             found = True
37             print(rainbow[index])
38         else:
39             index = index + 1
40
41
```




Paper 2 – Q02

```
1 # -----
2 # Global variables
3 # -----
4 plainText = ""
5 cipherText = ""
6 shift = 0
7
8 # -----
9 # Main program
10 # -----
11 plainText = input ("Enter a message: ")
12 shift = int (input ("Enter the shift: "))
13
14 for letter in plainText:
15
16     # =====> Choose the correct line to check for alphabetic letters
17     #if (letter.isalnum ()):
18     #if (letter.islower ()):
19     #if (letter.upper ()):
20     if (letter.isalpha ()):
21
22         value = ord (letter)
23         value = value + shift
24
25         # =====> Choose the correct line to check for upper case
26         #if (letter.upper ()):
27         #if (letter.isalpha ()):
28         #if (letter.islower ()):
29         if (letter.isupper ()):
30
31             # =====> Choose the correct line to check if the letter is outside the alphabet
32             #if (value > ord ('Z')):
33             if (value >= ord ('Z')):
34             #if (value < chr ('Z')):
35             #if (value < ord ('Z')):
36                 value = value - 26
```




```
37
38     # =====> Choose the correct line to check if the letter is outside the alphabet
39     #elif (value <= ord ('A')):
40     #elif (value > chr ('A')):
41     elif (value < ord ('A')):
42     #elif (value > ord ('A')):
43
44     value = value + 26
45
46     # =====> Choose the correct line to check for lower case
47     #elif (letter.lower ()):
48     elif (letter.islower ()):
49     #elif (letter.isupper ()):
50     #elif (letter.isalpha ()):
51
52     # =====> Choose the correct line to check if the letter is outside the alphabet
53     #if (value >= chr ('z')):
54     #if (value < ord ('z')):
55     if (value > ord ('z')):
56     #if (value <= chr ('z')):
57     value = value - 26
58
59     # =====> Choose the correct line to check if the letter is outside the alphabet
60     elif (value < ord ('a')):
61     #elif (value < chr ('z')):
62     #elif (value != ord ('a')):
63     #elif (value == chr ('z')):
64     value = value + 26
65
66     # =====> Choose the correct line to set the variable newLetter
67     #newLetter = ord (value)
68     #newLetter = chr (value)
69     newLetter = ord (letter)
70     #newLetter = chr (letter)
71
72     # =====> Choose the correct line to create the encrypted string
73     #cipherText = newLetter + cipherText
74     #newLetter = cipherText + newLetter
75     #newLetter = newLetter + cipherText
76     cipherText = cipherText + newLetter
77
78     else:
79     # =====> Choose the correct line to create the encrypted string
80     #cipherText = letter + cipherText
81     cipherText = cipherText + letter
82     #letter = cipherText + letter
83     #letter = letter + cipherText
84
85     print ("Plain text: ", plainText)
86     print ("Cipher text: ", cipherText)
```




Paper 2 – Q03

```
1 # -----
2 # Constants
3 # -----
4 PURCHASE_TYPE_ITEM = 1
5 PURCHASE_TYPE_WEIGHT = 5
6
7 PRICE_PER_KILOGRAM = 3.45
8 PRICE_PER_ITEM = 1.23
9
10 # -----
11 # Global variables
12 # -----
13 weight = 0.0
14 count = 0
15 totalCost = 0.0
16
17 # =====> Create an integer variable named purchaseType and set it to 0
18 purchaseType = 0
19
20 # -----
21 # Main program
22 # -----
23 purchaseType = int (input ("Enter a purchase type (1 or 5) "))
24
25 # =====> Complete the line with the correct logical operator and the correct constant
26 if ((purchaseType != PURCHASE_TYPE_ITEM) or (purchaseType != PURCHASE_TYPE_WEIGHT)):
27     print ("Invalid purchase type")
28
29 # =====> Complete the line with the correct constant
30 elif (purchaseType == PRICE_PER_ITEM):
31
32     # =====> Complete the line to accept a real value for the weight in kilograms
33     weight = float (input ("Enter weight in kilograms "))
34     if (weight <= 0):
35         print ("Invalid weight")
36     else:
37         # =====> Complete the line to calculate the total cost based on weight
38         totalCost = weight * PRICE_PER_KILOGRAM
39 else:
40     count = int (input ("Enter count of items "))
41     # =====> Complete the line to check for a 0 or negative count of items
42     if (count <=0):
43         print ("Invalid number of items")
44     else:
45         totalCost = count * PRICE_PER_ITEM
46
47 # =====> Complete the line with the correct relational operator
48 if (totalCost > 0.0):
49
50     # =====> Add a line to display an informative message and the total cost
51     print("Your total cost is £", totalCost)
```




Paper 2 – Q04

```
1 # -----
2 # Import libraries
3 # -----
4 import math
5
6 # -----
7 # Constants
8 # -----
9 CHEESE_PER_ADULT = 40      # Grams
10 CHEESE_PER_CHILD = 30     # Grams
11 MIN_CHEESE = 500         # 500 grams in a pack
12
13 ROLLS_PER_ADULT = 1.5     # Count
14 ROLLS_PER_CHILD = 0.5     # Count
15 MIN_ROLLS = 24           # Count of rolls in a pack
16
17 CRISPS_PER_ADULT = 0.75   # Of a bag
18 CRISPS_PER_CHILD = 0.33   # Of a bag
19
20 # -----
21 # Global variables
22 # -----
23
24 # =====> Write your code here
25 numAdults = 0
26 numChilds = 0
27
28 # -----
29 # Main program
30 # -----
31
32 # =====> Write your code here
33 numAdults = int(input("How many adults? "))
34 numChilds = int(input("How many children?"))
35
36
37 partialBags = (CRISPS_PER_CHILD * numChilds) + (CRISPS_PER_ADULT * numAdults)
38 print("Number of PARTIAL crisp bags needed:",partialBags)
39
40 wholeBags = partialBags // 1
41 print("Order",wholeBags,"bags of crisps")
42
43
44
45 cheeseGrams = (CHEESE_PER_CHILD * numChilds) + (CHEESE_PER_ADULT * numAdults)
46
47 if (cheeseGrams <= MIN_CHEESE):
48     print("Order one pack of cheese")
49 else:
50     wholeCheese = cheeseGrams // 1
51     print("Order",wholeCheese,"packs of cheese")
52
53
54
55 partialRolls = (ROLLS_PER_CHILD * numChilds) + (ROLLS_PER_ADULT * numAdults)
56 print("Number of PARTIAL rolls needed:",partialRolls)
57
58 wholeRolls = partialRolls // 1
59
60 if(wholeRolls <= MIN_ROLLS):
61     print("Order one pack of rolls")
62 else:
63     print("Order",wholeRolls,"rolls")
```




Paper 2 – Q05

```
1 # -----
2 # Import libraries
3 # -----
4 import random
5
6 # -----
7 # Constants
8 # -----
9 GET = 1
10 ADD = 2
11 SHOW = 3
12 EXIT = 4
13
14 # -----
15 # Global variables
16 # -----
17 pastaShapes = ["Bigoli", "Strozzapreti", "Trofie", "Gigli", "Chitarra",
18               "Penne", "Orecchiette", "Tagliatelle", "Chonchiglie",
19               "Fusilli"]
20
21 shape = ""
22 choice = 0
23
24 # -----
25 # Subprograms
26 # -----
27 # Get a menu item from the user
28 def getChoice ():
29     # =====> Write your code here
30     print ("1 - get a shape")
31     print ("2 - add a shape")
32     print ("3 - show the shapes")
33     print ("4 - exit program")
34     choice = int(input("What is your choice: "))
35
36     # =====> Write your code here
37     return(choice)
38
39 # Display all the shapes
40 def showShapes (pTable):
41     for pasta in pTable:
42         print (pasta)
43
44 # Get a random shape
45 def getShape (pTable):
46     # =====> Write your code here
47     print("Random shape generating: ")
48
49     for shape in pTable:
50         random = random.randint(0,9)
51
52 # Add a shape
53 def addShape (pTable):
54     # =====> Write your code here
55     newPasta=str(input("What is the name of your new pasta shape: "))
56     pastaShapes.append(newPasta)
57
58 # -----
59 # Main program
60 # -----
61
62 choice = getChoice ()
63
64 # =====> Write your code here
65
66 if choice == GET:
67     Rshape = getShape(pastaShapes)
68     print(Rshape)
69 elif choice == ADD:
70     Add= addShape(pastaShapes)
71 elif choice == SHOW:
72     show= showShapes(pastaShapes)
73 elif choice == EXIT:
74     print("Goodbye")
75 else:
76     print("invalid")
```




Paper 2 – Q06

```
1 # -----
2 # Global variables
3 # -----
4 cowTable = []
5
6 # =====> Write your code here
7 index = 0
8 # -----
9 # Subprograms
10 # -----
11 def showTable (pTable):
12     for cow in pTable:
13         print (cow)
14
15 # -----
16 # Main program
17 # -----
18 # =====> Write your code here
19 file = open("Cows.txt","r")
20 for line in file:
21     Linesplit = line.split(",")
22     tag = float(Linesplit[2])/100
23     #creates key
24     key =(Linesplit[0][0]+Linesplit[0][1]+str(tag)+Linesplit[1][0]+Linesplit[1][1])
25     #adds the key tag and breed to a record
26     record = key+str(tag)+Linesplit[1]
27     #adds record to list
28     cowTable.append(record)
29 showTable(cowTable)
30
```