```
#gcse-edexcel-june23-paper2-WORKED
### QUESTION 1 ###

#1 ------------------------------------------------------------------
#2 Constants
#3 ------------------------------------------------------------------
DISCOUNT_5 = 0.05           # 5% discount
DISCOUNT_10 = 0.10          # 10% discount

#7 ------------------------------------------------------------------
#8 Global variables
#9 ------------------------------------------------------------------
theTemperatures = [27.7, 29.8, 33.0, 31.6, 28.4, 28.0, 27.9]
aCost = 0.0
total = 0.0

#14 -----------------------------------------------------------------
#15 Main program
#16 -----------------------------------------------------------------
for temp in theTemperatures:        # Display temperatures
    print (temp)

#20 Add up costs until the user enters zero
aCost = float (input ("Enter a cost: "))
while (aCost != 0.0):
    total = total + aCost
    aCost = float (input ("Enter a cost : "))

#25 Calculate discount based on the total of numbers entered by the user
if (total > 100.00):
    total = (1 - DISCOUNT_10) * total       # Get 10% discount
else:
    total = (1 - DISCOUNT_5) * total        # Get 5% discount

print (total)
# ------------------------------------------------------------------
# =====> Write your answers here in the right-hand column
# Left                                          # Right
# ------------------------------------------------------------------
# Name of a constant used in the program        DISCOUNT_5
# Name of an array used in the program          theTemperatures
# Line number of an initialisation of a
#      variable with a real number              11
# Line numbers for a selection construct        26
# Line number(s) for a repetition construct     17-18
# Line number(s) for an iteration construct     22-24
# Line number for an instruction that outputs
#      information to the screen                18
```

```
###
###
###



##### Question 2 #####
#1 ---------------------------------------------------------------------
#2 Import libraries
#3 ---------------------------------------------------------------------
import random
#5 ---------------------------------------------------------------------
#6 Global variables
#7 ---------------------------------------------------------------------
exerciseTable = ["squats", "planks", "pushups", "lunges", "burpees"]
index = 0
name = ""
numExercises = 0
#12 --------------------------------------------------------------------
#13 Main program
#14 --------------------------------------------------------------------
print ("Here is the exercise table")

for exercise in exerciseTable:
    print(exercise)
numExercises = int(input("How many exercises do you need (1-5)? "))

for count in range (numExercises):
    index = random.randint(0,4)
    name = exerciseTable[index]
    print (name)




####
####
####



##### Question 3 #####
# ---------------------------------------------------------------------
# Constants
# ---------------------------------------------------------------------
SPECIFIED_MATERIAL = "Copper"

# =====> Add the correct input file name
```

```
INPUT_FILE = "Screws.txt"

# =====> Add the correct extension to this file name
OUTPUT_FILE = "Bricks.txt"

# ---------------------------------------------------------------
# Global variables
# ---------------------------------------------------------------
# =====> Complete the line with the correct variable name for the array of bricks
brickTable = ["Rustic", "Heather",
              "Staffordshire", "Tudor", "Hampton",
              "Norman", "Northcote",
              "Tuscan", "Regency",
              "Concrete Common",
              "Old English",
              "Hadrian Gold"]

inFile = ""
outFile = ""
foundCount = 0

# =====> Choose the correct value to initialise the variable
total = 0


# =====> Choose the correct value to initialise the variable
outLine = ""


# ---------------------------------------------------------------
# Main program
# ---------------------------------------------------------------

# Process the screws

# =====> Choose the correct line to open the file
inFile = open (INPUT_FILE, "r")

for line in inFile:
    # =====> Choose the correct line to locate the
    #        substring in the line
    if (line.find (SPECIFIED_MATERIAL) != -1):


        foundCount = foundCount + 1
# =====> Complete the line to increment total
    total = total + 1

# =====> Choose the correct line to close the file
inFile.close ()
```

```
# =====> Choose the correct line to display the output
print ("Total screws: " + str (total) + " " + SPECIFIED_MATERIAL + " screws: " +
str (foundCount))

# Process the bricks

# =====> Choose the correct line to open the bricks file

outFile = open (OUTPUT_FILE, "w")


for brick in brickTable:
    # =====> Choose the correct line to convert the case
    brick = brick.upper ()


    # =====> Choose the correct line to complete the output line
    outLine = brick + "\n"


    # =====> Choose the correct line to write the line to the file
    outFile.write (outLine)


outFile.close ()

# =====> Choose the correct line to display the output
print ("Wrote", len (brickTable), "brick names to file")



###
###
###
##### Question 4 ######


# ---------------------------------------------------------------------
# Global variables
# ---------------------------------------------------------------------
height = 0.0
base = 0.0
length = 0.0
area = 0.0
volume = 0.0
layout = "Volume is (;<8.2f) cubic units"
```

```
# ------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------
# =====> Write your code here
# Take three decimal inputs from the user
base = float(input("Enter the width of the base of the triangle: "))
height = float(input("Enter the height of the triangle:"))
length = float(input("Enter the length of the prism: "))

# Check for invalid inputs, using relational and logical operators
if ((height <= 0.0)or base(<= 0.0) or (length<= 0.0)):

# Display an error message if any input is invalid
# Invalid input should not be processed
    print("Invalid input")
else:
# Process all valid inputs
# Calculate the area of the triangle
    area = (1/2) * base * height

# Display the area of the triangle, rounded to two decimal places
    print("Area of the triangle is",round(area,2))

# Calculate the volume of the prism
    volume = area * length

# Display the volume of the prism using the <string>.format() function
# in eight columns with two decimal places
    print(layout.format(volume))
# In all cases, display a goodbye message just before terminating
print("Goodbye")


###
###
###
##### Question 5 #####
# ------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------
lastName = ""
firstName = ""
dob = ""
myID = ""

# ------------------------------------------------------------------
# Subprograms
# ------------------------------------------------------------------
# =====> Change the names of the local variables to distinguish them
#        from the global variables with the same name
```

```
def makeID (pLast, pFirst, pDob):
    namePart = ""
    numberPart = 0

    namePart = pLast + pFirst[0]  # Letter part

    # =====> Correct the logic error caused by using the int() function
    #        in the number part calculation rather than using a function
    #        that returns the ASCII value of the character
    for character in pDob:
        numberPart = numberPart + ord(character)

    yourID = namePart + str (numberPart)

    return (yourID)

# =====> Add a procedure, with no parameters, to display a
#        welcome message for the user
def welcomeUser():
    print("Welcome to the program")
# ---------------------------------------------------------------------
# Main program
# ---------------------------------------------------------------------
# =====> Call the welcome procedure before taking input from the user
welcomeUser()

# Get last name and first name from the user
lastName = input ("Enter your last name: ")
firstName = input ("Enter your first name: ")

# =====> Convert last name and first name to lowercase after they
#        are inputted by the user
lastName = lastName.lower()
firstName = firstName.lower()

# Get date of birth from the user
dob = input ("Enter your date of birth (ddmmyyyy): ")

# =====> Check that only the digits 0 to 9 appear in the date of birth
if (dob.isdigit()):


# =====> Call the makeID() function, if the date of birth is valid
    myID=makeID(lastName,firstName,dob)
    print(myID)
else:
# =====> Tell the user, if the date of birth is invalid
    print("Invalid date of birth")
```

```
###
###
###
##### Question 6 ######

# ---------------------------------------------------------------------
# Global variables
# ---------------------------------------------------------------------
userTable = [["LArmstrong3", "RedChair"],
             ["SBarrett7", "PurpleDesk"],
             ["EChisholm4", "YellowStool"],
             ["VDunn1", "OrangeFuton"],
             ["DElms5", "GreenCouch"],
             ["EFirsova13", "PinkMattress"],
             ["JGolland6", "GreenTable"],
             ["FHartley13", "BrownMirror"],
             ["DJohnstone12", "GoldBed"],
             ["GKirkhope8", "WhiteNights"],
             ["LLemon8", "BeigeDresser"],
             ["HMacCunn6", "GreyOttoman"],
             ["PNewland10", "BlackWardrobe"],
             ["AOldham5", "OrangeFuton"],
             ["JPook8", "YellowStool"]]

# =====> Write your code here
name = ""
password = ""       #user types in
foundName = False  #foud user name
letIn = False       #found full match
index = 0

# ---------------------------------------------------------------------
# Main program
# ---------------------------------------------------------------------

# =====> Write your code here
name = input("Enter your name: ")
password = input("Enter your password: ")

##check if input is valid
if ((len(name)==0)or (len(password)==0)):
    print("Invalid input")
else:
    while ((not foundName)and (index<len(userTable))):
        if (userTable[index][0] == name):
            foundName = True
            if (userTable[index][1] == password):
                letIn = True
        else:
            index=index+1
```

```
if letIn:
    print("welcome")
elif foundName:
    print("Incorrect password")
else:
    print("user not found")
```