NAME	CLASS	INDEX NO.
MARKING SCHEME		



ST. PATRICK'S SCHOOL PRELIMINARY EXAMINATION 2021

SUBJECT	:	Computing Paper 2 (7155/02)	DATE :	24 August 2021		
LEVEL	:	Secondary 4 Express	DURATION:	2 hours 30 minutes		
ADDITIONAL	ı	Electronic version of ORDERS.				
MATERIALS:		Electronic version of DRINK.py file				
		Electronic version of REVERSE . py file				
		Quick Reference Glossary				

Candidates answer on the Question Paper.

INSTRUCTIONS TO CANDIDATES:

Write your Name, Class and Index No. in the spaces at the top of this page. Write in dark blue or back pen.

Answer **all** the questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Programs are to be written in Python. Save your work using the file name given in the question as and when necessary.

The number of marks is given in brackets [] at the end of each question or part question. The total number of marks for this paper is 50.

INFORMATION FOR CANDIDATES:

The allocated marks for each question are indicated in the brackets [].

For Examiner's Use		
Score	/50	

This question paper consists of 8 printed pages, including the cover page.

	А	В	С	D	E	F
1	Tuesday, A	ugust 24, 2021				
2			(Orders		
3	Order ID	Zone	Bento Set	Quantity	Total Price	Price After Discount
4	E5969	E	С	3	\$96.90	\$77.52
5	W2903	W	В	2	\$45.20	\$36.16
6	N4090	N	А	1	\$20.40	\$20.40
7	E5000	E	В	2	\$45.20	\$36.16
8	S5904	S	D	3	\$53.70	\$42.96
9	C4231	С	E	4	\$94.00	\$75.20
10	W2144	W	А	2	\$40.80	\$32.64
11	N5225	N	А	3	\$61.20	\$48.96
12	C4119	С	E	2	\$47.00	\$37.60
13	C1825	С	С	4	\$129.20	\$103.36
14	\$4616	S	В	2	\$45.20	\$36.16
15	W2660	W	В	3	\$67.80	\$54.24
16					Total Revenue	\$601.36
17						
18	Bento Set	Α	В	С	D	E
19	Price	\$20.40	\$22.60	\$32.30	\$17.90	\$23.50
20	Total Quantity	6	9	7	3	6

Urders					
Order ID	Zone	Bento Set	Quantity	Total Price	Price After Discount
E5969	=LEFT(A4,1)	С	3	=HLOOKUP(C4,\$B\$18:\$F\$19,2,FALSE)*D4	=IF(E4>40,0.8*E4,E4)
W2903	=LEFT(A5,1)	В	2	=HLOOKUP(C5,\$B\$18:\$F\$19,2,FALSE)*D5	=IF(E5>40,0.8*E5,E5)
N4090	=LEFT(A6,1)	A	1	=HLOOKUP(C6,\$B\$18:\$F\$19,2,FALSE)*D6	=IF(E6>40,0.8*E6,E6)
E5000	=LEFT(A7,1)	В	2	=HLOOKUP(C7,\$B\$18:\$F\$19,2,FALSE)*D7	=IF(E7>40,0.8*E7,E7)
\$5904	=LEFT(A8,1)	D	3	=HLOOKUP(C8,\$B\$18:\$F\$19,2,FALSE)*D8	=IF(E8>40,0.8*E8,E8)
C4231	=LEFT(A9,1)	E	4	=HLOOKUP(C9,\$B\$18:\$F\$19,2,FALSE)*D9	=IF(E9>40,0.8*E9,E9)
W2144	=LEFT(A10,1)	A	2	=HLOOKUP(C10,\$B\$18:\$F\$19,2,FALSE)*D10	=IF(E10>40,0.8*E10,E10)
N5225	=LEFT(A11,1)	A	3	=HLOOKUP(C11,\$B\$18:\$F\$19,2,FALSE)*D11	=IF(E11>40,0.8*E11,E11)
C4119	=LEFT(A12,1)	E	2	=HLOOKUP(C12,\$B\$18:\$F\$19,2,FALSE)*D12	=IF(E12>40,0.8*E12,E12)
C1825	=LEFT(A13,1)	С	4	=HLOOKUP(C13,\$B\$18:\$F\$19,2,FALSE)*D13	=IF(E13>40,0.8*E13,E13)
S4616	=LEFT(A14,1)	В	2	=HLOOKUP(C14,\$B\$18:\$F\$19,2,FALSE)*D14	=IF(E14>40,0.8*E14,E14)
W2660	=LEFT(A15,1)	В	3	=HLOOKUP(C15,\$B\$18:\$F\$19,2,FALSE)*D15	=IF(E15>40,0.8*E15,E15)
				Total Revenue	=SUM(F4:F15)
Bento Set	А	В	С	D	E
Price	20.4	22.6	32.3	17.9	23.5
Total Quantity	=SUMIF(\$C\$4:\$C\$15,B18,\$D\$4:\$D\$15)	=SUMIF(\$C\$4:\$C\$15,C18,\$D\$4:\$D\$15)	=SUMIF(\$C\$4:\$C\$15,D18,\$D\$4:\$D\$15)	=SUMIF(\$C\$4:\$C\$15,E18,\$D\$4:\$D\$15)	=SUMIF(\$C\$4:\$C\$15,F18,\$D\$4:\$D\$15)

Save the file as **MYORDERS**_<Class>_<Class_Index_Number>_<Your_Name>.xlsx

- The first character of Order ID denotes the zone of the order.
 In cells B4 to B15, enter a formula that uses an appropriate function to display the zone of the order.
- In cells E4 to E15, enter a formula that uses an appropriate function to search for the Price of the respective Bento Set and calculate the total price of each order. [2]
- A discount of 20% is given for orders whose Total Price is above \$40.
 In cells F4 to F15, enter a formula that uses an appropriate function to calculate the price after discount for each order. [3]
- 4 In cell **F16**, enter a formula that uses an appropriate function that calculates the total revenue for the day. [1]
- 5 In cells **B20** to **F20**, enter a formula that uses an appropriate function that calculates the total quantity of each bento set ordered. [2]
- In cells F4 to F15, use a formatting tool to change the colour of the cells whose value is at least \$50, to green fill.

The following program displays a list of drinks available and asks user to input a drink to search within the list.

Open the file **DRINK.py**

Save the file as MYDRINK_<Class>_<Class_Index_Number>_<Your_Name>.py

7 Edit the program so that it searches the list based on the user input and displays a message "Drink found. Enjoy!" when the drink is found within the list and "Drink not found." when the drink is not found.

8 Save your program as VARDRINK_<Class>_<Class_Index_Number>_<Your_Name>.py

Edit the program so that it:

- (a) Asks the user for input whether or not to remove the drink from the list after the drink is found within the list. If user inputs "Yes", then, the drink is removed from the list and the same updated list is displayed to the user.
- (b) Ask the user repeatedly to input a drink to search within the list (and also input whether or not to remove the drink from the list) until the letter "q" is entered or the list is empty. The program will then exit.

Save your program.

```
a_list = ["PowerBoost", "EnergyOne", "RedOx", "VitaJuice", "HydroPlus"]
print (a_list)
found = False #Q1: correct initialisation of variables [lm]
search = input("Enter a drink: ")
while search.upper() != "Q": #Q2b: correct while loop [lm]
    if a list == []:
        print ("Empty list, exiting now")
        preak #Q2b: correct exit condition when list is empty [lm]
    for e in a_list:
        if e == search: #Q1: correct loop and compare search word [lm]
           found = True
    if found == True:
       print ("Drink found. Enjoy!") #Q1: correct output statement for search found [lm]
        option = input ("Would you like to remove this drink from the list? (Y/N):") #Q2a: ask for user input to remove drink [lm]
        if option.upper() == 'Y': #Q2a: correct if condition [lm]
            a_list.remove(search)
                                             #Q2a: removal of item in list and output updated list [1m]
           print (a_list)
   else:
       print ("Drink not found.") #Ql correct output statement for search not found [lm]
```

search = input("Enter a drink: ") #Q2b: ask for user input again [lm]

[3]

[4]

```
input text = input("Enter input text: ") #
input_text = input("Enter input text: "
                                                             word_list = []#
 word_list
                                                             current_word = ""
result = ""#
current_word = ""
result ==
                                                             index = 0
index = 0
                                                             while index < len(input_text):#
while index > len(input text)
                                                                 if not input_text[index].isspace():
    if not input_text[index].isspace():
                                                                     current_word += input_text[index]
        current_word += input_text[index]
                                                                 else:
    else:
                                                                     if current word != "":#
    if current_word != "":
                                                                         word_list += [current_word]
            word_list += [current_word]
                                                                         current_word = ""
            current_word = '
                                                                 index += 1
    index += 1
                                                             if current word != "":
if current_word != "":
                                                                 word_list += [current_word] #
    word_list =+ [current_word]
                                                            index = 0
 index = 0
                                                             while index < len(word_list):
while index < len(word_list):</pre>
                                                                current_word = ""
    current word =
    letter_index = 0
                                                                 letter_index = 0
                                                                 while letter_index < len(word_list[index]):</pre>
    while letter_index < len(word_list[index]):</pre>
                                                                   current_word += word_list[index][-letter_index - 1]
letter_index += 1
       current_word += word_list[index][-letter_index - 1]
        letter_index += 1
    word_list[index] = current_word
                                                                 word_list[index] = current_word
    index -= 1
                                                                 index += 1#
 index = 0
                                                             index = 0
 while index < (word list):
                                                             while index < len(word_list):#</pre>
    result += word_list[index] + " "
                                                                 result += word_list[index] + " "
    index += 1
                                                                 index += 1
 result = result[:1]
                                                             result = result[:-1]#
output (result)
                                                            print(result) #
Open the file REVERSE.py.
```

Save the file as MYREVERSE _<Class>_<Class_Index_Number>_<Your_Name>.py

9 Identify **and** correct the errors in the program so that it works according to the requirements given.

Save your program.

[10]

Moving averages are one of the most commonly used technical indicators by traders to identify price trends. They are normally used to filter out market noise and smooth out fluctuations in price to help traders distinguish between typical market fluctuations and actual price reversals.

The most basic type of moving average is Simple Moving Average (SMA). It is simply the average price over the specified period. The chart below shows an example of 200 period SMA, i.e. average over 200 data points.



Source: https://www.fidelity.com/learning-center/trading-investing/technicalanalysis/technical-indicator-guide/sma

Example:

If the most recent 5 prices are \$1, \$2, \$3, \$4 and \$5, the average would be the sum of the rates i.e. (1+2+3+4+5) divided by the number of reporting periods. Therefore SMA = 15/5 =\$3

Each time a new price becomes available, the average "moves" so that the average is always based only on the last same number of variables. In this case, if the next number in the sequence is 6, the oldest rate, i.e. 1 would be dropped and the new average would (2+3+4+5+6)/5 which equals 4.

You are tasked to write a program that generates the SMA given user inputs of 5 data points for each of the 3 different commodities (Commodity A, Commodity B and Commodity C). The program should:

- For each commodity, take as input 5 data points in the format a, b, c, d, e where a, b, c, d and are the prices across 5 reporting periods. You may assume that the prices are positive integers. [2]
- Only allow data entry of positive integers. [2]
- If user enters invalid data points, output an appropriate error message and ask the user to enter the data points again. [2]
- Calculate the SMA using the formula given. [1]
- Display the 5 data points and the SMA. [1]
- Repeat the above steps for the 3 different commodities, i.e. Commodity A, Commodity B and Commodity C. [2]
- 10 Write your program and test that it works. Save your program as MYSMA_<Class>_<Class_Index_Number>_<Your_Name>.py
- 11 When your program is complete, use the following test data to show your test result.
 - Commodity A with 1, 2, 3, 4, 5 as its data points across 5 reporting periods. [1]
 - Commodity B with 3, 6, 4, -2, 2 as its data points across 5 reporting periods. [1]
 - Commodity B with 3, 6, 4, 2, 2 as its data points across 5 reporting periods. [1]
 - Commodity C with 4, 5, 9, 12, 20 as its data points across 5 reporting periods. [1]

Take a screenshot of your result and save it as: **SMATEST**_<Class>_<Class_Index_Number>_<Your_Name> [1]

Save your file in either .png or .jpg format.

12 Save your program as **MYSMA2_**<Class>_<Class_Index_Number>_<Your_Name>.py

Extend your program so that the average "moves" as described on the previous page. It should accept user input of the next data point in the format a, b, c where a is the next data point for Commodity A, b as the next data point for Commodity B and c as the next data point for Commodity C. [3]

The program should then calculate the new SMA for each of the 3 commodities. [1]

The program should also display the initial data points and its initial SMA, as well as the new data points and its new SMA, for each of the 3 commodities. [1]

Save your program.

[5]

[5]

[10]