Group project

Assuming your group of Python developers joining a competition organized by the **MetaSpeed mall** to develop a system to promote better security, integrity, responsibility and tracing ability for their online ordering and fulfillment systems.

In order to reach the final round of the selection procedure, your groups need to accompany two tasks, they are

1) Write a Python module which provide "Flexible Hashing and Checking Digit creation" to verify the validity of the Invoice produced by the online ordering systems.

2) Conduct a 20 minutes presentation on your work to the MetaSpeed mall Manager

Reference : The Problem to Solve & Sample Invoice

First or Frist 2 alphabet : **range from A to Z, then AA to AZ, BA to BZ, …., ZA to ZZ (A)**

Next 6 digits : **staff number - in charge of this online order (123456)**

Next alphabet : **indicate the Modulus used in calculating the check digit (B) A – Modulus 7; B – Modulus 8; C – Modulus 9**

Next 6 digits : **order number (567878)**

Next digit : **Number of items in one order (maximum 9), if more than 9 , then switch to a new order (4 items in this one)**

Next digit in ( ) : **Check digit of the order number (7)**

1. Calculation of Check Digit for the order number
   A) Using the staff number as the multiplier
   B) Using the alphabet after the staff number to determine the Modulus used

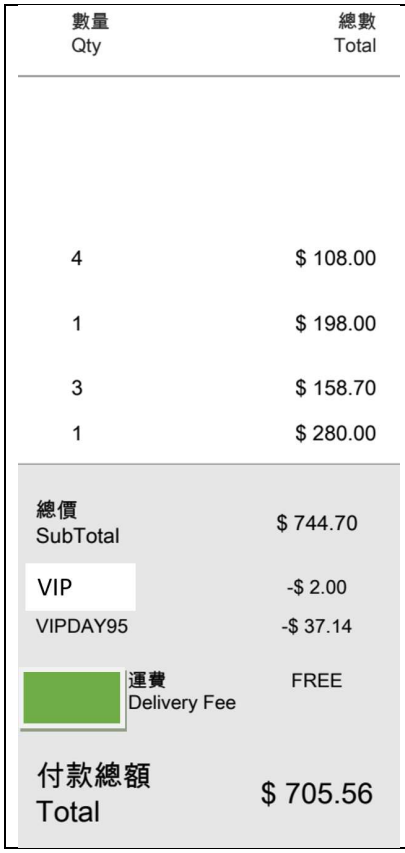
From the above, staff number is 123456 and using Modulus 8 to create check digit

1	2	3	4	5	6
*	*	*	*	*	*
5	6	7	8	7	8

= {5 + 12 + 21 + 32 + 35 + 48 + (Check Digit)} divisible by 8
= {153 + (Check Digit)} divisible by 8

So the check digit is 7

2. Cost Verification procedure



Sum of items total = Sub-total

There are 2 possible discount applied to the order

Delivery Fee depends on the Sub-total

Total = Sub-total – discount + delivery fee

Maximum number of items <= 9

Delivery Fee = 0 if Sub-total > = 500

= 5% of sub-total
(rounding to 2 decimal places; otherwise)

3. Hash Total for this order Calculation

1121. **2134.** **3019.** **4018.**	Hash Total for this order = sum of the 4 digits item number = 1121 + 2134 + 3019 + 4018 = 10292

4. New Hash Total for 1 - 10 orders under audit needed to develop
   You may use those non-money data for this purpose, your design

5. Mall dollars calculation
   Total > = 500, Mall dollars = Total * 0.002
   Total >= 1000, Mall dollars = Total * 0.0025

Input for the Program

Number of orders : range from 1 to 10

Within each order :

Determine the order number from the order file

For example, the last order number retrieved is A-567878; then the new order number is A – 567879; if the order number is z-999999; then the new order number is AA-000001

the 6 digit staff number in charge for the order

How many items within each order : range from 1 to 9, more than 10, it will go to another order (new order number)

For each items, enter 4 digit item number, quantity number and their cost

Discount 1 (VIP) for the order (actual amount less than 1% of the sub-total)

Discount 2 (VIPDay95) for the order (0 to 5%) apply to the sub-total

Customer number

Process

The group need to decide a method to assign which modulus method (9, 8, 7) are used

Determine the check digit for the order

Calculate the Hash Total for individual order

Calculate the New Hash Total for 1 - 10 orders **under audit needed**

Customer name and address are retrieved from a function call

(Name, Address) = name_address(customer_number)

You can arbitrary assigned the customer name and address with conditional statement within the function

Date_of_order : using week 10 material about datetime

Calculate the mall dollars

Calculate the sub-total, delivery fee and total

Output

Produce an output **similar** to the Sample Invoice especially for the format to print the Order No. 

as shown which contain not just order number information.

- (Clear output is the requirements) ; Formatting output **is not required!**

Also provide information about the Hash Total of individual order and 1-10 orders under audited. If more than 1 orders processed, provide clear division between orders

The output **file** that store the last order number

An audited file with extension (.txt) to store the information in the following format

Number_of_orders	XX	(1-10)
Hash_total_of_orders	XXXXX	(the value of the Hash total of all the orders)
Order 1 details		
Order_Number	A-XXXXXX	(start with Alphabet, then 6 digits number)
Agency_number	XXXXXX	(6 digit number)
Modulus_number	X	
Total	XXXXX	(the money amount)
Hash_total	XXXX	(Hash total of this order)

Empty line

Order 2 details

……

……

Marking Scheme for Programs (30 marks)

17 marks: Program achieved the requirements
3 marks: Comments within the Program
5 marks : Individual need to **write a reflection** on this exercise (max. 4 pages, font size 12)
5 marks : 20 minutes presentation for each group in the last week lecture

Bonus (3 marks maximum)

Innovative idea to improve the flexibility of the program

Due Date

Section 3 Due Date : 18/04/2023 Tuesday before 15:30 through moodle
Section 1 Due Date : 19/04/2023 Wednesday before 18:30 through moodle
Section 2 Due Date : 20/04/2022 Thursday before 18:30 through moodle

General requirements for Program

1) Provide the list of members in the beginning of the program
2) Write comments to explain your programming logic
3) When your capture input, provide routine to handle the wrong input problems
4) Good use of data structure available in Python
5) To solve the problem

Group submission

Program :	**GPX.py X is your group ****
Text file (3 files at least)	the input last order file, the output last order file and the audited file
Test data and output	in pdf format OR screen snapshots
The programs listing	in pdf format
The presentation ppt	in ppt
Workload distribution	in pdf

Peer assessment and Individual reflection **in pdf or WORD format**