XINMIN SS PRELIM 2018 Task 1

You are tasked to evaluate the loan packages from four banks in Singapore. All the loans are using compound interest. You are required to finish setting up the spreadsheet to show the total amount payable to each bank at the end of the loan period.

Open the file **LOAN**. You will see the following data.

зł.	А	В	с	D	E	F	G
1				Loan Evaluati	on		
2							
3	Loan Number	Amount Deposited	Loan Tenure (year)	Loan Tenure (month)	Bank	Loan Interest Rate	Total Amount Payable to Bank
4	L001	\$10,000.00	1		A		
5	L002	\$10,000.00	1		В		
6	L003	\$10,000.00	1		С		
7	L004	\$10,000.00	1		D		
8	L005	\$10,000.00	3		A		
9	L006	\$10,000.00	3		В		
10	L007	\$10,000.00	3		С		
11	L008	\$10,000.00	3		D		
12	L009	\$10,000.00	5		A		
13	L010	\$10,000.00	5		В		
14	L011	\$10,000.00	5		С		
15	L012	\$10,000.00	5		D		
16				· · · · · · · · · · · · · · · · · · ·	la di mari	-	
17							
18							
19							
20		Bank	Interest Rate per Month (1 year loan)	Interest Rate per Month (3 year Ioan)	Interest Rate per Month (5 year Ioan)		
21		A	0.32%	0.32%	0.32%		
22		В	0.42%	0.38%	0.48%		
23		С	0.59%	0.40%	0.40%		
24		D	0.47%	0.47%	0.47%		

Save the file as EVA_<Your name>_<Class>_<Index number>

- Use a formula to convert the number of years in the Loan Tenure (year) column to number of months and use it to complete the Loan Tenure (month) column. The values should be displayed as number with no decimal places. [3]
- 2 Use an appropriate function to search for the **Interest Rate per Month (x year Ioan)** in the **Rates** table and use it to complete the **Loan Interest Rate** column. The interest rate should be displayed as percentage of two decimal places. [4]
- 3 Use an appropriate function to calculate the total amount of payable to the bank at the end of the loan and use it to complete the **Total Amount Payable to Bank** column. These values should be displayed as currency of two decimal places. [3] Save and close your file.

Task 2

The following program accepts three words and joined them together to form a sentence by adding a space in between them, and eventually prints out the entire sentence.

```
num_of_words = 3
result = ""
for count in range(num_of_words):
    word = input("Enter a word: ")
    result += word
    if count < num_of_words-1:
        result += " "
print("Sentence =", result)</pre>
```

Open the file JOINWORDS.py

Save the file as MYWORDS_<Your name>_<Class>_<Index number>

- 4 Edit the program so that it:
 - (a) Accepts only 5 words [1]
 - (b) Prints out the total number of characters used to construct the entire sentence [1]
 - (c) Prints out the words entered on separate lines before the sentence is printed, e.g.

```
Word List:
I
love
Computing
Sentence = I love Computing
```

[5]

(d) Checks if the word input has a length of 1 to 15 characters, and if not, asks the user for the input again as necessary. [3]

Save your program.

Task 3

The following program should read a denary non-negative integer from the user. The program will then convert the denary integer to its binary value and print it to the screen. The "division by 2" method is employed to carry out the conversion.

There are several syntax errors and logical errors in the program.

```
NEW_BASE == 3
num = Input("Enter a non-negative integer: "
num = float(num)
result = ""
q = num
r = q % NEW_BASE
result = str(r) + result
q = q // NEW_BASE
while q > 0
r = q % NEW_BASE
result = result + str(r)
q = q / NEW_BASE
print(result, "in Decimal is", num, "in Binary.")
```

Open the file D2B.py

Save the file as MYD2B_<Your name>_<Class>_<Index number>

5 Identify and correct the errors in the program so that it works correctly according to the description above. [10]

Save your program.

Task 4

You have been asked to write a program to simulate the Caesar Cipher. The cipher was used by Julius Caesar (Roman dictator) to encrypt written instructions to his generals.

The Caesar Cipher is a type of substitution cipher where each letter in the original plaintext message is replaced by a letter some fixed positions down the alphabet. Consider the case where each letter in the plaintext is shifted by 3 places. As a result, A becomes D, B becomes E, C becomes F, D becomes G, etc. The last three letters in the alphabet are wrapped around to the beginning: X becomes A, Y becomes B and Z becomes C. Non-letter characters are not modified by the cipher.

Table 4 shows an example where the letters have been shifted to the right by 3. Note that the letters "wrap-around" at the end.

Α	В	С	D	Е	F	G	Н	Ι	J	Κ	L	Μ	Ν	0	Ρ	Q	R	S	Т	U	V	W	Х	Υ	Ζ
D	Е	F	G	Н	-	J	Κ	L	Μ	Ν	0	Ρ	Q	R	S	Т	U	V	W	Х	Υ	Ζ	Α	В	С

Table 4

The plaintext of "HELLO" will be encrypted as "KHOOR" based on the cipher that has the positions of the alphabet shifted to the right by 3.

The program should allow you:

- To enter a plaintext in uppercase.
- To enter the shift value for the cipher.
- To encrypt the plaintext.
- Not to encrypt non-letter characters.
- To display the encrypted message on the screen. Your output **must** look like this:

```
Please enter your plaintext: HELLO WORLD!
Please enter a shift value: 3
Encrypted message: KHOOR ZRUOG!
```

6 Write your program and test that it works.

Save your program as CIPHER1_<Your name>_<Class>_<Index number>

[12]

7 When your program is working, use the following plaintexts and shift values as test data to show your test results:

Plaintext	Shift value						
HELLO WORLD!	3						
XINMIN SEC	2						
PHYSICS RULES	1						

Take a screen shot of your results and save it as a bitmapCIPHERRESULTS_<Your name>_<Class>_<Index number>[5]

 8 Save your program as CIPHER2_<Your name>_<Class>_<Index number> Extend your program to only allow shift values of 1 to 25 to be entered, and if not, ask the user for the input again as necessary.
 [3]

Save your program.

End of paper