O-Level Candidate Index No :

Name:

) Class: Thumb Drive :



# BUKIT VIEW SECONDARY SCHOOL **PRELIMINARY EXAMINATION 2018 Secondary 4 Express**

COMPUTING

Paper 2 Practical (Lab-based)

7155/2 29 Aug 2018 2 hour 30 minutes

Electronic version of CREDITLOAN.XLSX data file Additional Materials: Electronic version of ITEMCATEGORIES.PY file Electronic version of ISBNCHECKDIGIT.PY file Insert Quick Reference Glossary

## **READ THESE INSTRUCTIONS FIRST**

## Do not open this booklet until you are told to do so.

Write your name, index number, and class in the spaces provided at the top of this page.

Answer all 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.



### Task 1

Horizon Credit Pte Ltd uses spreadsheet software to record the amount of money loaned and how many months the money will be loaned. The borrowers will repay the loan in a lump sum at the end of the loan period. Interest is compounded monthly. You are required to finish setting up the spreadsheet to record the lump sum amount that will be paid by the borrowers.

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

Save the file as LOAN\_<your name>\_<centre number>\_<index number>

а	A	в	с	D	E	F	G
1			Horizon Cre	dit Pte Ltd			
2							
-	Borrower Number	Amount Loaned	Number of Months Loaned	Borrower Interest Rate	Lump Sum Amount Payable	Check Credit History	
4	HC001	\$4,500.00	17				
5	HC002	\$6,000.00	23				
6	HC003	\$500.00	4				
7	HC004	\$3,600.00	10				
8	HC005	\$1,000.00	6				
3	HC006	\$250.00	18				
10	HC007	\$3,800.00	24				
11	HC008	\$700.00	20				
12	HC009	\$2,500.00	30				
13	HC010	\$2,000.00	3				
14	HC011	\$5,200.00	8				
15	HC012	\$3,000.00	25				
16	HC013	\$4,600.00	12				
17	HC014	\$1,800.00	22				
18	HC015	\$5,500.00	25				
19							
20	Total						
21	Number of Loans						
22	Range of Months						
23		Rates		Rates			
24				Months Boundary	Description	laterest Rate per Year	
25				0	0 to 5 months	6.0%	
26				6	6 to 11 months	5.5%	
27				12	12 to 17 months	5.0%	
28				18	18 to 23 months	4.5%	
29				24	24 months and over	4.0%	
30							

1	In cell <b>B20</b> , enter a formula to calculate the total amount loaned.	[1]	

- 2 In cell **B21**, enter a formula to count how many loans have been made. [1]
- 3 In cell **B22**, enter a formula to calculate the range of number of months loaned. [1]
- 4 Use an appropriate function to search for the **Interest Rate per Year** in the **Rates** table and use it to complete the **Borrower Interest Rate** column. [2]
- 5 Enter a formula to calculate the lump sum payable for borrowers and use it to complete the **Lump Sum Amount Payable** column. Round these values to the nearest whole number. [3]
- 6 Use a conditional statement, to identify those borrowers who have borrowed \$2000 or more for over 12 months and put YES in the Check Credit History column. Otherwise, put NO in the Check Credit History column.
  [2]

Save and close your file.

### Task 2

A small store sells 15 items in a day. The categories of items include drinks, sandwiches and cakes. Each item sold is identified by a 3-digit code. All item codes which start with 0 are drinks category. All item codes which start with 1 are sandwiches category. All item codes which start with 2 are cakes category.

Item Code	Category Name	
000 to 099	drinks	
100 to 199	sandwiches	
200 to 299	cakes	

The following program accepts the 3-digit code for the items sold in a day. It calculates the number of items sold in each category.

```
drinks_category = 0
sandwiches_category = 0
cakes_category =0
items = 15
for i in range (items):
    item_string = input("Enter the item code ")
    item_code = int(item_string)
    if item_code < 100:
        drinks_category = drinks_category + 1
    elif item_code < 200:
        sandwiches_category = sandwiches_category + 1
    else :
        cakes_category = cakes_category + 1
print ("The number of drinks, sandwiches and cakes sold are ",
drinks_category, sandwiches_category, cakes_category)</pre>
```

### Open the file ITEMCATEGORIES.py

Save the file as MYCATEGORIES\_<your name>\_<centre number>\_<index number>

- 7 Edit your program so that it:
  - (a) Accepts the item codes for 20 items. [1]
  - (b) Prints out the category name with the highest number of items sold and also its number of items sold. You may assume that there is only 1 category with the highest number of items sold. [4]
  - (c) Includes validations to test whether the item code entered is valid, and if not, asks the user for input again as necessary. [3]

Save your program.

# 8 Save your program as VARCATEGORIES\_<your name>\_<centre number>\_<index number>

Edit your program so that it works for any number of items.

[2]

Save your program.

### Task 3

Check digit is used in the International Standard Book Number. An example is the ISBN 13, where the 13<sup>th</sup> digit of the ISBN code is calculated using the following algorithm.

- Step 1 : Add all the odd numbered digits together, excluding the check digit, and multiply the result by the weight, 1.
- Step 2: Add all the even numbered digits together and multiply the result by the weight, 3.
- Step 3 : Add the results from Step 1 and Step 2 together and divide by 10.
- Step 4 : Take the remainder, if it is zero use this value as the check digit. Otherwise subtract the remainder from 10 to find the check digit.

For example, a valid ISBN 13 is 9789814176712 and an invalid ISBN 13 is 9789814176703.

The following program should check whether an ISBN number is valid by deriving its resulting check digit from the first 12 digits.

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

```
multiplier weight = 1
total = 13
isbn = input ("Enter an ISBN-13 number with no spaces ")
check digit = int(isbn[13])
for i in range(11):
     total = total + isbn * multiplier_weight
     if multiplier weight == 1:
        multiplier weight = 0
     else:
        multiplier weight = 1
remainder = total % 11
result = remainder - 11
if result == remainder:
     print ("ISBN Number is valid")
else:
     print ("ISBN Number is invalid")
```

#### Open the file ISBNCHECKDIGIT.py

Save the file as MYISBNCHECKDIGIT\_<your name>\_<centre number>\_<index number>

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

Save your program.

### Task 4

A company employs salesmen to sell cars. The company operates on a 5-days week from Monday to Friday. You have been asked to write a program to calculate the average number of cars sold per week by four salesmen. Each salesman may sell any number of cars in a day.

The program should allow you to:

- Enter data in the format a, b, c, d, e where a, b, c, d, e are the number of cars sold by a salesman over the 5 days. An example is 2, 0, 3, 1, 1 where a Salesman sells 2 cars on Day 1 (Monday), 0 car on Day 2 (Tuesday), 3 cars on Day 3 (Wednesday), 1 car on Day 4 (Thursday) and 1 car on Day 5 (Friday).
- Only allow data entry of 0 or more cars.
- Calculate the total number of cars sold by each salesman.
- Repeat this for a total of 4 salesmen.
- Find the average number of cars sold by a salesman in the 5-days week rounded to the nearest whole number.
- Calculate the total number of cars sold for the 5-days week.
- Display this on the screen. Your output **must** look like this:

Salesman	1	7 car(s)		
Salesman	2 9	) car(s)		
Salesman	3 (	5 car(s)		
Salesman	4 5	ō car(s)		
Average number of cars 7				
Total number of cars for the week 27				

**10** Write your program and test that it works.

Save your program as SALESMAN1 \_<your name>\_<centre number>\_<index number>

11 When your program is working, use the following test data to show your test results.

~				
2,	Ο,	3,	1,	1
,	- /	- /		
2	2	$\cap$	1	2
J,	<i>∠</i> ,	υ,	<b>⊥</b> /	5
$\cap$	1	$\hat{}$	$\sim$	1
υ,	<b>⊥</b> ,	Ζ,	Ζ,	$\perp$
-	~	~	~	~
⊥,	υ,	Ζ,	υ,	2
1,	0,	2,	0	,

Take a screen shot of your results at the screen output and save it as a bitmap **SALESMANRESULTS**\_<your name>\_<centre number>\_<index number>

[5]

[10]

12 Save your program as **SALESMAN2** \_<your name>\_<centre number>\_<index number>

Extend your program to identify days that have less than 5 cars sold. Print out the day number and the number of cars sold on that day. Your output should look like this:

Day	2 3	car(s)
Day	4 4	car(s)

[3]

Save your program.

Save your program as SALESMAN3\_<your name>\_<centre number>\_<index number>
 Extend your program to work for any number of salesmen. [2]
 Save your program.

### - End of Paper -