

SECONDARY 4 PRELIM EXAMINATION

COMPUTING Paper 2 Practical (Lab-based)

7155/02

27 August 2019 (Tuesday)

2 hour 30 minutes

CANDIDATE NAME			
CLASS		INDEX NUMBER	
Additional Materia	als: Electronic version of B Electronic version of S Electronic version of N Insert Quick Reference	ANK.XLSX data file HOPPING.PY file UMBERS.PY file Ə Glossary	

READ THESE INSTRUCTIONS FIRST

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.

The Bank of Singapore (BOS) uses a spreadsheet software to record the money deposited and how many months the money will be in the bank. You are required to finish setting up the spreadsheet to record the interest that will be paid. For consistency, you are also required to display all results as **positive numbers**.

	A	В	С	D	E	F	G
1	Account Number	Amount Deposited	Number of Months deposited	Interest Rate	Interest payable	Gift Voucher	What-If
2	ID001	\$4,050.00	13				
3	ID002	\$2,030.00	15				
4	ID003	\$780.00	22				
5	ID004	\$3,120.00	24				
6	ID005	\$2,440.00	38				
7	ID006	\$9,250.00	42				
8	ID007	\$8,850.00	7				
9	ID008	\$8,210.00	20				
10	ID009	\$6,490.00	40				
11	ID010	\$5,630.00	16				
12	ID011	\$480.00	28				
13	ID012	\$2,100.00	36				
14	ID013	\$1,460.00	40				
15	ID014	\$3,520.00	18				
16	ID015	\$2,200.00	6				
17	ID016	\$4,680.00	44				
18	ID017	\$6,270.00	32				
19	ID018	\$4,200.00	18				
20	ID019	\$1,530.00	28				
21	ID020	\$5,710.00	12				
22							
23	Standard deviation	of amount depo	osited				
24	Number of deposits	s more than					
25					RATES		
26					Minimum Number of Months Deposited	Description	Interest Rate per year
27					0	one year or less	0.00%
28					12	one to two years	0.50%
29					24	two to three years	1.00%
30					36	three to four years	1.50%
31					48	four years or more	2.00%
22						,	

Open the file BANK.xlsx. You will see the following data.

Save your file as **BANK_**<your name>_<index number>.

- (a) In cell C23, enter a formula to calculate the standard deviation of the amount deposited. [1]
- (b) In cell C24, enter a formula to count how many deposits are more than \$3000. [1]
- (c) Use an appropriate function to search for the Interest Rate per year in the RATES table and use it to complete the Interest Rate column. [2]
- (d) Enter a formula to calculate the simple interest payable for account holders and use it to complete the Interest Payable. [2]

- (e) Use a conditional statement to identify those savers who have deposited more than \$3000 over 24 months and put YES in the Gift Voucher column. Otherwise put NO in the Gift Voucher Column.
- (f) The bank manager wants to do some scenario planning, where each account holder deposits his/her money for 48 months with a compounded interest rate of 2.00% per annum. The amount of money in the account after 48 months is then recorded in the What-If column. Complete the column for all the bank accounts. [2]

Save and close your file.

The following program simulates a shopping cart. It accepts 3 different items and the quantity of each item, and places them in a list. After processing, it prints out the item with the largest quantity, as well as its respective quantity.

```
#input items in a shopping cart and the quantity
1
2
   different items = 3
3
   shopping cart = []
 4
5
   shopping item = []
   for i in range(different items):
6
7
       new item = input("Please enter an item: ")
       new quantity = input("Please enter the quantity of item: ")
8
9
       shopping_item = [new_item, int(new_quantity)]
       shopping cart.append(shopping item)
10
   print(shopping cart)
11
12
13
   largest quantity = 0
14
   sequence = 0
   for j in range(different items):
15
       if shopping_cart[j][1] > largest_quantity:
16
17
            sequence = j
18
            largest_quantity = shopping_cart[j][1]
19
   print("The item with the greatest quantity is
20
   ", shopping cart[sequence][0])
   print("There are ",largest quantity, shopping cart[sequence][0],
21
   "in total.")
```

Open the file **SHOPPING.py**

Save the file as SHOPPING _<your name>_<index number>

Edit the program so that it:

(a)	Allows the input of five different items and their respective quantities.	[1]
(b)	Ensures that the quantities input by the user are positive integers. Otherwise, the program should inform the user about the input requirements and ask for re-entry of the input.	[3]
(c)	Also displays the item name with the smallest quantity, followed by its quantity separated by a space.	[3]
(d)	Allows user to input the unit cost of each item. All items are priced in dollars.	[1]
(e)	Calculates and displays the total cost of all the items in the shopping cart.	[2]

Donald is fascinated with the number 1089 and he has recently learnt about a formula that will end up with the number 1089 as long as the input requirements are met.

The requirements are as follows:

- 1. A three-digit number is input.
- 2. The first and third digits of the number cannot be the same, e.g. 121 is not allowed.

The formula to produce 1089 is as follows:

- 1. Let A be the input number.
- 2. Let B be the number where first and third digits of A are swapped.
- 3. Let X = |A B| (absolute value of A B).
- 4. Let Y = the number where first and third digits of X are swapped.
- 5. Let Result = X + Y.

Open the file **NUMBERS**.py

Save the file as **NUMBERS**_<your name>_<index number>.py Write a program to help Donald test out this formula. Your program must be able to do the following:

- Get the required number from user with appropriate prompt.
- Verify the input. If the verification fails, display an appropriate error message and ask the user to re-enter the number.
- Display the subtraction step in a single line. E.g. "721 127 = 594".
- Display the addition step in a single line. E.g. "594 + 495 = 1089".
- Ask the user if he wants to try again. Repeat step 1 if user agrees; otherwise end the program.

Save your program.

[10]

Zelda is studying in Hyrule Secondary School (HSS). In HSS, the overall percentage for each subject is calculated using this formula:

40% of Sem1 + 60% of Sem2 = Overall (100%)

Where

Sem1 = 30% of CA1 + 70% of SA1Sem2 = 30% of CA2 + 70% of SA2 $0 \le marks \le 100$

This is Zelda's streaming year. In order to get into the class with the subject combination of her choice, Zelda needs to ensure she obtains a certain overall percentage for each of her subjects. Zelda knows her marks for *CA1*, *SA1* and *CA2* of Subject A.

- (a) Write a program to help Zelda find out how many marks she needs to score in *SA2* to obtain a desired overall percentage for Subject A. Your program needs to do the following:
 - Ask Zelda to input the marks *CA1*, *SA1* and *CA2* respectively in a single line. You may modify and use the following code:

cal, sal, ca2 = input().split()

- Verify the input. When verification fails, an appropriate message must be displayed. Your program should then ask Zelda to enter the marks again.
- Calculate the required marks Zelda needs to obtain for her *SA2* in order to achieve the following overall grades.

Overall Grade	Overall Percentage
A1	75 to 100
A2	70 to 74
В3	65 to 69
B4	60 to 64

 Display all four desired overall grades and the required SA2 in the following format "Grade : Required Mark(rounded to one decimal place)" If required mark is not obtainable, display "Not obtainable" instead. For example:

A1:77.4 ... B4:61.9 Save your file as **TESTQ4a** <your name> <index number>.py

- (b) Run your program with the following inputs
 - 1. 65 101 80 2. 65 81 a 3. 75 40 70

Take a screenshot of the program display and save it as TESTQ4b_<your name>_<index number>.jpg

(c) Zelda wants to know if she needs more consultation time with her subject teacher. She comes up with the following algorithm.

Value of Required Mark	Remarks
Required Mark \geq 120% of <i>SA1</i>	More time
120% of $SA1 > \text{Required Mark} \ge 80\%$ of $SA1$	Maintain
80% of <i>SA1</i> > Required Marks	Can divert

Modify the program to display the appropriate remarks if the required mark is obtainable. For example:

```
A1 : 77.4 (More time)
```

Save your program as **TESTQ4c**_<your name>_<index number>.py [6]

(d) Run your program with the following input

 53 55 60
 Take the screenshot of the program display and save it as

 TESTQ4d_<your name>_<index number>.jpg

--- END OF PAPER ----

[10]

[3]