

COMMONWEALTH SECONDARY SCHOOL PRELIMINARY EXAMINATION 2022

COMPUTING Paper 2 (Practical)

Name:) Class:	Class:
SECONDARY FOUR EXPRESS/N	C) Monday 29 August 2022	Monday 29 Augus
7155/2	1130 – 1400	1130
	2 hour 30 mins	2 hour 3

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.

Name of setter: Mrs Leong Sai Hong

Task 1

Commonwealth Hardware Company uses spreadsheet software to keep track of the amount of bonus given to its employees.

	А	в	С	D	E	F	G
1	Commonwealth Hardware Company						
2							
3	Current year	2022					
4							
5	Name of employee	Username	Year joined	Years of service	Long service award	Position	Bonus (\$)
6	Kelly Lee		1992			Manager	
- 7	Heidi Woo		2001			Operator	
8	Jerry Thong		2015			Sales Rep	
9	Jacintha Anna		1999			Sales Rep	
10	Toh Liang		2004			Manager	
11	Clement Choo		2017			Manager	
12	Seng Kin		2020			Operator	
13	Holly Gang		1987			Manager	
14	Ali Ahmad		1997			Sales Rep	
15	Krishna Jai		2005			Operator	
16	Jolene Ee		2021			Manager	
						Average	
						Bonus	
17						(\$)	
18	Long service award	by year					
		Long service					
19	Years of service	award					
20	0	Bronze					
21	10	Silver					
22	20	Gold					
23	30	Platinum					

Open the file **HARDWARE.xlsx** and you will see the following data.

Save the file as MYHARDWARE_<class>_<index_number>_<your name>.xlsx

- 1. In cells **B6 to B16** enter a formula that uses an appropriate function to generate the username for each employee. The username is created by the first n characters of the member's name where n is a random number between 1 and the length of the name. For example, for employee, Kelly Lee, her username will be Kel if n is 3. [2]
- In cells D6 to D16, enter a formula that calculates the number of years of service for each employee. [1]
- Use an appropriate function to search for the Long service award in the Long service award by year table and use it to complete the Long service award column.

4. In cells **G6** to **G16**, enter a formula that uses functions to calculate the bonus each employee is to receive based on the following criteria:

Long service award	Position	Bonus (\$)
Platinum or Gold	Manager	5000
Silver	Any	500
Any	Any	50

[3]

5. In cells **G17** enter a formula that uses functions to find the average amount of bonus given out. The average amount must be rounded to 2 decimal places. [2]

Task 2

In Singapore, electronic road pricing (ERP) is implemented on various expressways to regulate traffic. Lately there has been a change in the ERP rates at 5 gantries across some expressways.

The following program calculates the change in rates of these 5 gantries.

```
for i in range(5):
    expressway = input("Enter name of gantry:")
    old = float(input("Enter old rate:"))
    new = float(input("Enter new rate:"))
    change = new - old
    print("Change is", change)
```

Open the file EXPRESSWAY.py

Save the file as **MYEXPRESSWAY**_<class>_<index_number>_<your name>.py

- 6. Edit the program so that:
 - (a) It works for any number of gantries. The program must display a suitable input message. [1]
 - (b) The name of gantry is accepted if it is made up of only letters and is of a maximum length of 20. A suitable error message must be displayed and the program must loop until the name of the gantry is an input of a maximum of 20 letters.
 [4]
 - (c) The name of each gantry for which the ERP rate has been increased is stored in a list and then displayed. [4]
 - (d) The total number of gantries which saw an increase in the ERP rate is displayed. [1]

Task 3

A program asks the user to enter a letter and a sentence. It then converts each character of the sentence into lowercase before searching through the sentence for the letter entered. If the letter is not found, it will display a message to inform the user. Otherwise, it will display the position at which the letter is found in the sentence. It will also give a comment based on the frequency of the occurrence of the letter in the sentence as shown:

No. of times letter occurs in the sentence	Comment
1, 2	The letter rarely occurs.
3 - 5	The letter occurs some times.
6 or more	The letter occurs many times.

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

```
letter = input("Enter a lower case letter to find:")
sentence = input("Enter sentence:").islower()
frequency = 0
if sentence.find(letter) = 0:
    print("Letter is found at position:")
    while sentence.find(letter) != -1:
        frequency -= 1
        sentence = sentence.find(letter)
        print(idx)
        sentence = sentence[:idx] + "#" + sentence[idx:]
    if frequency > 5:
        print("Letter occurs many times")
    elif:
        print("Letter occurs some times")
    elif:
        print("Letter occurs rarely")
else
   print("Letter (letter) is not found").format()
```

Open the file OCCUR.py

Save the file MYOCCUR_<class>_<index number>_<your name>.py

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

Task 4

You have been asked to write a program to simulate the sending and receiving of binary data packet on a network.

8. Save your program as **ENTERDATA**_class>_<index number>_<your name>.py

In the same program, write a function enter_data() which asks the user to enter a binary data packet. Output a suitable error message if the binary data packet is not made up of only '0's or '1's. Repeatedly ask the user to re-enter until it does. [3]

Sample executions:

```
>>> enter_data()
Enter a binary data packet: 0020110
Data packet should consists of only '0's or '1's.
Enter a binary data packet: #bn0110
Data packet should consists of only '0's or '1's.
Enter a binary data packet: 0011011
0011011
```

Save your program.

9. Save your program as **COUNTONES**_<class>_<index number>_<your name>.py

In the same program, write a function count_ones(d), which counts the number of '1's in the d argument. [3]

Sample executions:

```
>>> count_ones('0011101')
4
>>> count_ones('00000000')
0
```

10. Save your program as **ADDBIT**_<class>_<index number>_<your name>.py

Use the count_ones(d) function to write a function add_bit(d,oddeven), which decides whether to append a '1' or a '0' to the d argument depending on whether the oddeven argument is 'odd' or 'even' using the following criteria:

Number of '1's in the d argument	oddeven argument	Character to append to the d argument
odd	'odd'	ʻ0'
even	'odd'	'1'
odd	'even'	'1'
even	'even'	ʻ0'

[4]

Sample executions:

```
>>> add_bit('0011101','odd')
00111011
>>> add_bit('0011101','even')
00111010
>>> add_bit('0000000','odd')
00000001
>>> add_bit('0000000','even')
00000000
```

Save your program.

11. Save your program as CHECKPARITY_<class>_<index number>_<your name>.py

Use the count_ones(d) function to write a function check_parity(d,oddeven), which returns True or False based on the oddeven argument and the number of '1's in the d argument:

Number of '1's in the d argument	oddeven argument	Boolean to return
odd	'odd'	True
even	'even'	True
odd	'even'	False
even	'odd'	False

[3]

Sample executions:

```
>>> check_parity(`00111010','odd')
False
>>> check_parity(`00111010','even')
True
```

12. Save your program as **PARITY**_<class>_<index number>_<your name>.py

Extend your program by creating a simple user interface. The program should

- Request user to enter a binary data packet.
- Perform a validation check to ensure that the data packet consists of only '1's or '0's and display error message as necessary.
- Request the user to choose the parity system to be used. The user is to enter 'even' if an even parity system is to be used and to enter 'odd' if an odd parity system is to be used.
- Ask the user if he is sending or receiving the data. The user is to enter 'S' if he is sending the data and to enter 'R' if he is receiving the data.
- If the user is sending the data, append the parity bit to the data and then output the new data.
- If the user is receiving the data, check if the data has been received correctly and output the result accordingly. [7]

Sample execution 1:

Enter a binary data packet: 0021101 Data packet should consist of only '0's or '1's. Enter a binary data packet: 0011101 Parity system (odd/even): odd Are you sending or receiving data? S Data packet with parity bit: 00111011

Sample execution 2:

Enter your data: 0011101 Parity system (odd/even): even Are you sending or receiving data? S Data packet with parity bit: 00111010

Sample execution 3:

Enter your data: 10100111 Parity system (odd/even): odd Are you sending or receiving data? R Data received correctly.

Sample execution 4:

Enter your data: 10100111 Parity system (odd/even): even Are you sending or receiving data? R Data received incorrectly.

Save your program.

END OF PAPER