Name:

Index Number:



# Anglo-Chinese School (Barker Road)

# **PRELIMINARY EXAMINATION 2019**

# SECONDARY FOUR EXPRESS

# COMPUTING PAPER 2

# 7155/02

# 2 HOUR 30 MINUTES

## **INSTRUCTIONS TO CANDIDATES**

## Additional Materials: Electronic version of LOANS.XLSX data file Electronic version of FACTORS.PY Python file Electronic version of CHECKCODE.PY Python file Insert Quick Reference Glossary

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.

This document consists of **6** printed pages, inclusive of the cover page

A company uses a spreadsheet to record the loans from customers. You are required to finish setting up the spreadsheet to calculate the monthly instalment for each of the account number.

	А	В	С	D	E	F	G	
1		Loans						
2	Account number	Loan type	Loan Amount	Loan Tenure (Years)	Interest rate per annum	Processing Fee (\$)	Monthly Instalment	
3	1001Car		10000	4				
4	1002Cash		7000	2				
5	1003Study		5500	2				
6	1004Study		11500	4				
17	1015Cash		10000	4				
18	1016Cash		40000	2				
19	1017Study		10000	1				
20								
21			Loan Types					
22	Tenure (Years)	1	3	5		Car		
23	Interest Rate per annum	3.70%	3.88%	4.50%		Cash		
24	Processing Fee	0%	1%	2%		Study		

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

Save the file as **MYLOANS\_**<your name>\_<index number>

- 1 Use a function to extract the type of loans from account number and use it to complete the **Loan Type** column. The type of loans is the letters from the account number. [2]
- 2 Use an appropriate function to search for the **Interest rate per annum** in the **Rates** table and use it to complete the **Interest rate per annum** column. [2]
- 3 Use an appropriate function to search for the **Processing Fee** in the **Rates** table and use it to complete the **Processing Fee (\$)** column. The processing fee is a percentage from the loan amount. [2]
- 4 Use an appropriate function to calculate the monthly instalment amount and use it to complete the **Monthly Instalment** column. [2]
- 5 Use an appropriate function to find out the number of Car, Cash and Study in the Loan Types table. [2]

Save and close your file.

The following program accepts positive integer numbers and prints out the factors for the numbers.

```
for i in range(2):
    factor_count = 0
    number = int(input("Enter a number:"))
    for i in range(1,number+1):
        if number%i==0:
            factor_count += 1
```

Open the file FACTORS.py

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

6 Edit the program so that it:

(a)	Accept input for 5 numbers.	[1]
(b)	Prints out the factors for each of the number in a list.	[4]
(c)	Tests if the input is a positive number, and if not, asks the user for input again.	[3]
Save	your program.	
Save	your program as <b>MYFACTORS2</b> _ <your name="">_<index_number></index_number></your>	

Save your program as MYFACTORS2\_<your name>\_<index\_number>
 Edit your program so that the whole program will repeat as many times until zero is entered.

Save your program.

The alphabet at the end of a NRIC number is called its check code. As its name suggests, the check code allows us to check if a NRIC number has been entered correctly. The algorithm for generating the check code is as follows:

1. Obtain the weighted sum of the NRIC digits using the weights [2, 7, 6, 5, 4, 3, 2]. For NRIC number 9300007, the sum is

 $(9 \times 2) + (3 \times 7) + (0 \times 5) + (0 \times 4) + (0 \times 3) + (7 \times 2) = 53$ 

- 2. Find the remainder of the sum when divided by 11: 53 % 11 = 9
- 3. Subtract the remainder from 11: 11 9 = 2
- 4. Look the check code up:

1	2	3	4	5	6	7	8	9	10	11
Α	В	С	D	Е	F	G	Н	Ι	Ζ	J
The check code for 9300007 is B.										

The program accepts NRIC and checks if the NRIC is valid.

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

```
check = (2, 7, 6, 5, 4, 3, 2)
code = ["A", "B", "C", "D", "E", "F", "G", H, "I", "Z", "J"]
total = 0
counter = 0
NRIC = input("Enter the NRIC:")
while NRIC.isalnum() == False and len(NRIC) != 9:
    if NRIC.isalnum() == False:
        print("Special character is not allowed!")
    else:
        output("NRIC must be 9 digits")
    NRIC = input("Enter the NRIC again:")
while counter<len(NRIC)-1:
    for digit in check:
        total += digit * NRIC[counter]
        counter == 1
remainder = total//11
subtract = remainder - 11
if NRIC[1] == code[subtract-1]:
    print("The NRIC is valid.")
else:
    print("Invalid NRIC.")
```

### Open the file CHECKCODE.py

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

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

Save your program.

You have been asked to create an attendance program.

The program should:

- Enter the data in the format a b c d e where they are the attendance from Monday to Friday. 1 represents present and 0 represents absent. An example is 1 0 1 1 0
- Make sure that the input is in the correct format
- Only allow data entry of 0 and 1
- Calculate the total number of days of present for each student
- Repeat for seven students
- Calculate the present rate for the whole week rounded to 1 decimal place. [Present rate = total attendance / total possible attendance \* 100%]
- Output the student with full attendance of the week
- Display this on the screen. Your output **must** look like this:

```
Student 1
              3 day(s)
Student 2
              4 day(s)
Student 3
             3 day(s)
Student 4
            5 day(s)
Student 5
             4 day(s)
Student 6
              3 day(s)
Student 7
             5 day(s)
Present rate = 77.1 %
Student 4 has full attendance
Student 7 has full attendance
```

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

Save your program as **MYATT**\_<your name>\_<index number>.py

[10]

- **10** When your program is complete, use the following test data to show your test result:

Take a screen shot of your results and save it as bitmap **MYRESULT**\_<your name>\_<index number>

[5]

11 Save your program as MYATT2\_<your name>\_<index number>.py

Extend your program to calculate the number of students present for each day.

Your output should look like this:

Mon - 6 student(s)
Tue - 5 student(s)
Wed - 6 student(s)
Thu - 5 student(s)
Fri - 5 student(s)

Save your program.

12 Save your program as **MYATT3**\_<your name>\_<index number>.py

Extend your program to work for any number of students.

[2]

[3]

Save your program.

#### **End of Paper**