

# NGEE ANN SECONDARY SCHOOL

# PRELIMINARY EXAMINATION

COMPUTING	7155/02
Paper 2 (Lab-based)	31 August 2022
	2 hr 30 min
Additional materials:	Electronic version of YUMCOM.xlsx file Electronic version of PASSWORD.py file Electronic version of NUMSYS.py file Insert Quick Reference Glossary

## Instructions to Candidates

Write your name, register number and class 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.

#### For Examiner's Use

Marks	/50

Checked by student: \_\_\_\_\_ Date:

This document consists of <u>8</u> printed pages and <u>2</u> blank pages.

The following spreadsheet is used to store the particulars of participants who had taken part in a competitive eating competition.

You are required to finish setting up the spreadsheet to complete the details of the participants.

You may make use of the following EXCEL function to join texts together: CONCATENATE(<*text1*>,<*text2*>)

Open the file **YUMCOM.xlsx**. You will see the following data.

	А	В	С	D	E	F	G	Н	I.		
1	1 Yummy Competitive Eating Contest 2022 Participant Details										
2		-									
3		Curent Year:	2022								
4											
5	S/N	Name	Birth Year	Tag ID	Age (years)	Area Represent	Insurance Coverage?	Time Taken (min)	Category		
6	1	Gregory Tan	1995			East	Yes	82			
7	2	Ng He Jin	1989			Central	Yes	40			
8	3	Raymond Bong	1999			West	Yes	72			
9	4	Veerapan S/O Manish	2001			Central	No	32			
10	5	Charlie Chong	1985			North	Yes	63			
11	6	Ang Man Wen	1993			West	Yes	95			
12	7	Ariff Bin Zaki	2002			East	No	38			
13	8	Hermes Quek Zi Long	1992			East	Yes	60			
14											
15											
16		Age (no. of years at least)	19	25	30						
17		Cateogory	Novice	Intermediate	Veteran						
18											
19											
20		Past Top 3 Names	Year		2022 Top 2	Timings (min)					
21		James Lai	2021		1st:						
22		Chen Ding Sheng	2021		2nd:						
23		Ng He Jin	2021								
24		Desmond Chua De Mun	2020		Average age (years) of 2022 participants to nearest whole number:						
25		Veerapan S/O Manish	2020								
26		Ramus Liu	2020								

Save the file as **MYYUMCOM\_2022**\_<your name>\_<centre number>\_<index number>.xlsx

1 The **Tag ID** of the participants consists of the last 3 letters of their name in column B, followed by the first 2 letters of the area they represent in column F.

In cells **D6** to **D13**, enter a formula that uses functions to create the **Tag ID** of all the participants. [2]

2 The age of the participants may be approximated to be the difference between the current year and their birth year.

In cells **E6** to **E13**, enter a formula that calculates the age (in years) of all the participants. [1]

**3** The table in cell range **B16:E17** shows the categories in the competition based on the age of the participants.

In cells **I6** to **I13**, enter a formula that displays the category of each participant based on their calculated ages. [2]

- In cells F21 and F22, enter the most appropriate formula for each cell to show the fastest 1<sup>st</sup> and 2<sup>nd</sup> timings among all the participants. [2]
- 5 In cell **124**, enter a formula that uses functions to calculate the average age (years) of all the participants, rounded off to the nearest whole number. [2]
- 6 In cells A6 to I13, use a formatting tool to change the colour of the row to yellow if the participant has **Insurance coverage** and has **time taken** that is less than 60 min. [1]

Save and close your file.

The following program is written to help set a new password whenever the existing password has expired. The password is to be made up of alphanumeric characters.

```
past words = ["PopeyetheSailorm00n",
              "123qweasd",
              "StrOngestpasswOrd"]
final = False
while final == False:
    alpha chk = False
    num chk = False
    p1 = input("Enter new password: ")
    for i in p1:
        if i.isalpha():
            alpha chk = True
        elif i.isdigit():
            num chk = True
        elif not i.isalnum():
            alpha chk = False
            num chk = False
            break
    if alpha chk == False or num chk == False:
        print("Password has to be alphanumeric.")
    if (alpha chk and num chk):
        print("Password set.")
        final = True
```

Open the file **PASSWORD**.py

Save the file as **MYPASSWORD\_2022**\_<your name>\_<centre number> <index number>.**py** 

7 The password has to be at least 8 characters long.

Edit the program so that it checks if the password is at least 8 characters in length. The program must display a suitable output message to inform users about the criteria if it is not met.

Save your program.

[3]

8 Passwords used previously cannot be reused.

The list variable, past words contains three previously used passwords.

```
past_words = ["PopeyetheSailorm00n",
                                "123qweasd",
                             "Str0ngestpassw0rd"]
```

Edit the program so that it checks if the new password has been reused. The program must display a suitable output message to inform users about the criteria if it is not met.

Save your program.

[3]

9 The new password has to be re-entered by the user to verify that it is correct.

Edit the program so that it prompts for the password to be entered again. The entire password setting process would have to restart if the re-entered password is not the same. The program must display suitable input and output messages to inform users.

Save your program.

5

The program below is written to convert either a positive integer input into its binary number format or a binary number input into its corresponding denary number.

```
choice = input ("Press B for conversion to Binary and D for conversion
to Denary: ")
choice = choice.upper()
while (choice != 'D' or 'B'):
    print("Wrong input.")
    choice = input("Press B for conversion to Binary and D for
conversion to Denary: ")
    choice = choice.upper()
if choice == "D":
    denary = input("Enter denary number: ")
    denary = (denary)
    binary = ""
    while True:
       r = denary%2
        binary += str(r)
        denary = denary%2
        if denary == 0:
            continue
    print("The binary number is:", binary[-1::])
elif choice == "D":
    denary = 0
    binary = input("Enter denary number: ")
    binary = binary[::-1]
    for n in range(binary):
        denary += int(binary[n])*2*n
    print("The denary number is:", denary)
```

## Open the file NUMSYS.py

Save your program as **MYNUMSYS\_2022**\_<Your name>\_<Centre number>\_<Index number>.**py** 

**10** Identify **and** correct the errors in the program so that it works according to the requirements given.

Save your program.

[10]

**BLANK PAGE** 

This page is intentionally left blank

A typical deck of poker cards has four different patterns (Diamond, Club, Heart, Spade), each numbered from 2 to 10 and additional cards labelled "Jack", "Queen", "King" and "Ace".

For the purpose of this task, "Jack", "Queen", "King", "Ace" and each pattern will be represented by their starting letter, namely, "J", "Q", "K", "A", "D", "C", "H", and "S". "Ace" also represents number 1.

A two-player game using such a deck requires each player to be dealt 26 cards each. The players will calculate their total points based on a number of rules:

- For every pair of numbered cards ("Ace" to 9) that total up to 10, the player gets 5 points.
- For every pair of Jacks or pair of Queens or pair of Kings, the player gets 10 points.
- For every one card that shows number 10 or card that cannot be paired, the player gets -1 point.
- 11 Write a function deal(), which generates and shuffles the deck of cards each time it is called. [4]

Save your program as **DEAL\_2022**\_<Your name>\_<Centre number>\_<Index number>.**py** 

#### Sample executions:

```
>>> deal()
['2D', '7S', '9H', '4S', '2C', 'KS', 'KC', '9C', 'QH', '4H',
'AC', '10S', 'AH', 'QC', '8D', 'KH', '4C', '6S', 'KD', '6H',
'JH', '3D', '7C', '8S', 'AS', '5C', '10C', '3S', '6D', '8C',
'2H', 'JC', 'JS', '9S', '7H', '10D', '7D', '3H', '5H', 'AD',
'JD', '3C', '2S', '4D', '9D', 'QS', 'QD', '10H', '5D', '5S',
'8H', '6C']
>>> deal()
['AS', '8S', 'QH', '4C', '6D', '8C', '9S', '6C', '5S', '2D',
'KS', '4S', '5H', '3S', '4H', '10H', '4D', '8H', '2S', 'QS',
'JD', '7C', 'KH', '7H', '7S', '5C', '6S', 'AC', '3C', '3D',
'JC', '3H', 'JH', '7D', '9D', '6H', 'JS', '8D', '2H', 'QC',
'10C', '10S', '5D', 'QD', '9H', 'KD', '10D', 'KC', 'AH', '9C',
'AD', '2C']
```

12 Call the function deal() twice and take a screenshot of it.

Save it as **DECK\_2022**</br>

Your name>\_
[1]

13 Save your program as **POINT\_2022**\_<Your name>\_<Centre number>\_<Index number>.py

In the same program, write a function points(), which takes in a list of cards to represent the cards on hand and calculates the number of points a player has according to the rules.

Save your program.

[12]

[3]

14 Save your program as **GAME\_2022**\_<Your name>\_<Centre number>\_<Index number>.py

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

- Divide the shuffled deck into two parts. First half for player 1 and second half for player 2.
- Output the number of points each player has.
- Output the results of the game, who the winner is or if there is a tie.

Save your program.

Sample execution 1:

Player 1 has 73 points. Player 2 has 73 points. It's a tie.

#### Sample execution 2:

Player 1 has 59 points. Player 2 has 68 points. Player 2 wins.

#### Sample execution 3:

Player 1 has 68 points. Player 2 has 59 points. Player 1 wins.

-- End of Paper --

# **BLANK PAGE**