YISHUN INNOVA JUNIOR COLLEGE
JC1 MID YEAR EXAMINATIOIN

CANDIDATE NAME

CT GROUP | INDEX NUMBER

H2 COMPUTING **9569/01**

04 Jul 2019

2 Hr 30 Mins

80 Marks

Additional Materials: Removable storage device with the following files :

- `template.ipynb`
- `queue.py`

READ THESE INSTRUCTIONS FIRST

Type the candidate details in the Jupyter Notebook template provided.
Rename and save the file as **`19S1x`**`_<index number>_<your name>.ipynb` for submission.

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.

Approved calculators are allowed.

The number of marks is given in brackets [ ] at the end of each question or part question.

At the end of the examination, save all the program codes and submit your Jupyter Notebook file in the thumb drive provided.

This document consists of **9** printed pages.

Question 1

The exchange rate for one Singapore Dollar (SGD) to the US Dollar (USD) and Japanese Yen (JPY) are as follows:

```
USD = 0.72669, JPY = 79.97
```

Task 1

Write a program code `rate(currency_1,currency_2)` to return the conversion rate from `currency_1` to `currency_2`.

```
rate(SGD,USD) = 0.72669
rate(SGD,JPY) = 79.97
```

Test your program with `rate(USD, JPY)`. [2]

Question 2

The triangle number sequence is a series of whole numbers where the n^{th} term is calculated by adding the previous number to n. The first seven terms of the triangle number sequence are as follows:

1, 3, 6, 10, 15, 21, 28 and so on…

Task 2.1

Write a program code `triangle(n)` to generate the n^{th} term of the triangle number sequence using a recursive approach.

```
triangle(4) >> 6 + 4 = 10
triangle(7) >> 21 + 7 = 28
```

Test your program with the given test cases. [2]

A program, `sum_even_triangle(n)` takes in an integer `n` as input and return the sum of all the even-valued numbers in the first `n` terms of the triangle number sequence.

For example:

```
1. sum_even_triangle(1) >> 0
2. sum_even_triangle(3) >> 6
3. sum_even_triangle(7) >> 6 + 10 + 28 = 44
```

Task 2.2

Write a program code for `sum_even_triangle(n)` using a recursive approach.

Test your program with each of the above test cases. [6]

Question 3

A simple substitution cipher can be used to encrypt plain messages.

The cipher alphabet would consist of a key in front, e.g. `'zebra'` which consists of unique characters, and the remaining characters of the plain alphabet not found in the key, arranged in their usual alphabetical order.

plain alphabet: `'abcdefghijklmnopqrstuvwxyz'`

cipher alphabet: `'zebracdfghijklmnopqstuvwxy'`

In this example, `'a'` is replaced by `'z'`, `'b'` is replaced by `'e'` and so on.

The message `'hello world'` would become `'fajjm vmpjr'` after encryption with the cipher alphabet.

Assume that the plain messages and keys are in the lower case.

Task 3.1

Write a program code `cipher_alphabet(key)` that uses the `key` and returns the cipher alphabet string. [4]

Test your program by generating another cipher alphabet using a different key. [1]

Task 3.2

Write a program code `cipher_dictionary(key)` that uses the `key` and returns a dictionary which maps the plain alphabet to the cipher alphabet generated with the `key`. [5]

Task 3.3

Write a program code `encrypt(message, key)` to encrypt the plaintext message and return the ciphertext. You should use the program code `cipher_alphabet(key)` written in Task 3.1.

Test your program code for the following test cases:

```
encrypt('hello world', 'zebra') >> 'fajjm vmpjr'

encrypt('hello world', 'computing') >> 'nuddh whldp'
```

[8]

Question 4

Some scores, ranging between `0` and `100` inclusively, are compiled to be transmitted through the network to another station.

For example:

`to_string ([29,100,67,31,45,84])` should return `'029100067031045084'`.

Task 4.1

Write a program code `to_string(lst)` to compile the scores in the list `lst` and return the string.

[4]

When the information reaches the receiving station, it will execute a program to extract the data and find the average of the scores.

Task 4.2

Write a program code `average(string)` to extract the data and return the average of the scores in the string. Test your program with another set of scores. [8]

Question 5

Consider the following Pseudocode for a number guessing game:

1. The computer randomly generates a secret number.
2. The player repeatedly inputs a number to guess.
3. If the player makes the correct guess, the game ends.

```
01 REPEAT
02     SecretNumber <- Random whole number from 0 to 100
03     INPUT Guess
04     IF Guess > SecretNumber
05         THEN
06          Message to player "Input a smaller number"
07     ENDIF
08     IF Guess < SecretNumber
09         THEN
10             Message to player "Input a larger number"
11     ENDIF
12 UNTIL Guess = SecretNumber
```

There is an error in the algorithm causing the game to work incorrectly. Identify and correct the error.

Task 5

Write the correct program code in Python for this guessing game. When the player makes the correct guess, the program will output the number of guesses he made and stops. [7]

Question 6

An abstract data type (ADT) is used to collect the name, grade and score of each student.

The accessors with the following specifications are provided in the `module.py`:

```
get_name(tuple)-> string
get_grade(tuple)-> string
get_score(tuple)-> integer
size(list)-> integer
```

The results of the students are organised using the ADT and the user should use the correct accessors to retrieve the respective data.

```
students = [('tiffany', 'A', 15),
            ('jane', 'B', 10),
            ('ben', 'C', 8),
            ('simon', 'A', 21),
            ('eugene', 'A', 21),
            ('john', 'A', 15),
            ('jimmy', 'F', 1),
            ('charles', 'C', 9),
            ('freddy', 'D', 4),
            ('dave', 'B', 12)]
```

Task 6

Write a program code `deviation(lst)` to :

- extract the scores of the students and store them in a list `lst`
- compute and return the standard deviation of scores, correct to 2 decimal places.

The formula for computing the standard deviation is:

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n}}$$

The variable $\bar{x}$ is the average of n input values x_1 through x_n. [7]

Question 7

A food ordering program uses a queue data structure to manage food orders. A queue data structure can be implemented using a list.

The `queue.py` module provided contains the following functions:

- `make_queue()` constructs an empty queue
- `enqueue(q,item)` appends the `item` to the queue `q`
- `dequeue(q)` returns and removes the first item in the queue `q`, returns `None` if `q` is empty
- `front(q)` returns the item in the front of the queue `q`
- `size(q)` returns the size of the queue `q`

Task 7.1

Write the program codes for the following modifiers:

- `delete(q,item)` to remove the `item` from the queue `q`
- `shift(q,item)` to shift the `item` to the front of the queue `q`

Both the modifiers should return `None` if the `item` is not found in the queue `q`. [4]

Task 7.2

Write a program code to:

- construct a new empty queue named `orders`
- add the following items to the queue, `orders`
    - 'Fish Burger'
    - 'Chicken Burger'
    - 'Chicken Nuggets'
    - 'Fried Chicken'
    - 'Nasi Lemak Burger'
- remove one item from the queue
- return the first item of the queue
- delete the item 'Fried Chicken' from the queue
- shift the item 'Nasi Lemak Burger' to the front of the queue
- returns the number of items in the queue [4]

Question 8

Every letter in the alphabet has an Unicode which can be obtained using Python's built-in function, `ord()` as shown below:

```
ord('a')=97
```

The character `'a'` has an Unicode 97.

If the sum of the Unicodes of the letters in a person's name is even, then he has a *balanced name*.

For example, sum of the Unicodes of all the letters the name `'Hulk'` is computed as follows:

```
ord('H')+ ord('u')+ ord('l')+ ord('k')
= 72 + 117 + 108 + 107
= 404
```

Hence the name `'Hulk'` is a *balanced name* since 404 is an even number.

Task 8.1

Write a program code `balanced(name)` that takes in string `name` and returns `True` if the `name` is balanced, and `False` otherwise.

```
balanced('Hulk') returns True
```

Test your program with another two names, like `'IronMan'` and `'DoctorStrange'`. [4]

The villain Thanos was obsessed with the idea of bringing the universe to its balanced state. He fought through many wars to acquire all the infinity stones, so that with a snap of his fingers, he could remove all those without a *balanced name*. His action eliminated half of all the living creatures and he believed that in doing so, the universal will return to its balanced state.

Task 8.2

Write a program code `thanos(names)` to remove all those names in the list which are not a *balanced name* and return the list with only the survivors.

Test your program with the given list of singers. [3]

The Avengers searched through history to gather more information about the infinity stones and their power, with 1 being the most powerful:

```
infinity_stones = [('soul stone',13),
                   ('mind stone',6),
                   ('reality stone',9),
                   ('space stone',7),
                   ('time stone',1),
                   ('power stone',5)]
```

Task 8.3

Write a program code `top(lst)` to remove and return the most powerful stone from the list `lst`.

Test run your program twice consecutively to get the top two most powerful stones.

```
top(infinity_stones) >> ('time',1)
top(infinity_stones) >> ('power stone',5)
```

[6]

Task 8.4

Write a program code `sort_stones(lst)` that takes the list `lst` and returns a new list where the stones are sorted in ascending order of their power.

You may use the code `top(lst)` written in Task 8.3. [5]

The Avengers found a way to travel back in time to steal all the infinity stones before Thanos could collect them and eventually saved the universe.

- End of Paper -