Task 1

For this task, submit all code as `T1_<index number>_<name>.py`.

The Taliban's takeover of Afghanistan had seen the United States (US) evacuating people. The Republic of Singapore Air Force offers its A330 Multi-Role Tanker Transport plane (A330 MRTT) to help the US airlift groups of evacuees from Afghanistan.

Task 1.1

To prevent Taliban interference, messages sent between evacuees and the A330 MRTT are encrypted.

The encryption converts each character of the message into its ASCII number representation, adds the ASCII number representation of a character of a secret key which is in the same position, and then converts it back into ASCII text.

Secret keys are generated from passwords known only by the sender and receiver. Passwords that are shorter in length than the message are lengthened to match by repeating the password until the same length as the message is achieved.

For example, given that the inputs are

```
password            : "cat"
message             : "hello"
```

Step 1 - Extend password by repeating until matches length of message

```
"cat" -> "catca"
```

Step 2 - Convert each character of password and message into ASCII number representation

```
"hello" -> 104, 101, 108, 108, 111
"catca" -> 99, 97, 116, 99, 97
```

Step 3 - Add the ASCII number representation of same positions together

```
104+99, 101+97, 108+116, 108+99, 111+97 -> 203, 198, 224, 207, 208
```

Step 4 - Convert each number to its ASCII character

```
203, 198, 224, 207, 208 -> "ËÆàÏÐ"
```

Therefore, the results are

```
Secret key          : "catca"
Encrypted message   : "ËÆàÏÐ"
```

Write a function `generateKey(password, message)` that takes in two argument strings `message` and `password` and returns the secret key.

Test your function using `generateKey("cat", "hello")` and show your output. [4]

Task 1.2

Write functions `encrypt(password, message)` and `decrypt(password, message)` that takes in `password` and message strings and returns the encrypted or decrypted `message`. It should use the above function `generateKey(password, message)` to obtain the secret key before performing the encryption.

Test your function with `decrypt("cat", encrypt("cat","hello"))`. Show your output.

[6]

Task 1.3

Since the A330 MRTT plane can only take a maximum of 266 passengers at once, implement a Queue to manage incoming pickup requests and dropoffs. Write a class `Queue` with the following methods:

- `enqueue(string)` which takes in a string and adds it to the queue or returns `"Queue is full!"` if queue is full
- `dequeue` which removes the next item for the queue or returns `"Queue is empty!"` if the queue is empty
- `display` which returns a string of queue items in sequence from head to tail or returns `"Queue is empty!"` if the queue is empty
- `current_size` which returns an integer of the number of items in the queue.

[8]

Task 1.4

Using socket programming, write the client program for the evacuee teams (client) in Afghanistan and the A330 MRTT plane (server) to communicate. The program should

- use the above queue data structure to manage the requests
- request the user to set a password for encryption at the start of the program
- present the user with incoming pickup requests and pickup details (Team name and group size) as well as queue details (size of current queue)
- allow the user(server) to accept/reject pickup requests
- encrypt all information sent and received using `encrypt(password, message)` and `decrypt(password, message)` functions.

You are provided with the client program in `client.py`. Write and submit the server program as `T1_<index number>_<name>.py`.

Study the following sample program output to determine your code design, output format and socket protocol. User inputs are underlined.

Sample server program:

```
Please set a password:
Answer: WDIB4
```

```
Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:2

Queue is empty!


Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:1

Awaiting connection from pickup client...

Connection established!
Receiving data from client…
Checking client's secret key…
Client's secret key is correct!

Client's Team name: Team1
Group size: 143
No. of passengers in queue: 0
You have capacity for them.

Confirm pickup? Y/N?
Answer:Y

Added to queue. Items in queue now are:
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1,
Connection disconnected.

Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:1

Awaiting connection from pickup client...
```

```
Connection established!
Receiving data from client…
Checking client's secret key…
Client's secret key is correct!

Client's Team name: Team2
Group size: 151
No. of passengers in queue: 143
You do NOT have capacity for them.
Connection disconnected.


Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:1

Awaiting connection from pickup client...

Connection established!
Receiving data from client…
Checking client's secret key…
Client's secret key is correct!

Client's Team name: Team3
Group size: 19
No. of passengers in queue: 143
You have capacity for them.

Confirm pickup? Y/N?
Answer:N

Connection disconnected.

Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:1

Awaiting connection from pickup client...

Connection established!
Receiving data from client…
Checking client's secret key…
Client's secret key is correct!

Client's Team name: Team4
Group size: 15
No. of passengers in queue: 143
You have capacity for them.

Confirm pickup? Y/N?
```

```
Answer:Y

Added to queue. Items in queue now are:
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1, Team1,
Team1, Team1, Team1, Team4, Team4, Team4, Team4, Team4, Team4, Team4,
Team4, Team4, Team4, Team4, Team4, Team4, Team4, Team4,
Connection disconnected.

Next action?
Menu:
1) Wait for connection from the next pickup location client
2) Dequeue
Type an option:2

How many to dequeue? 143

Items in queue now are:
Team4, Team4, Team4, Team4, Team4, Team4, Team4, Team4, Team4, Team4,
Team4, Team4, Team4, Team4, Team4,
Connection disconnected.

Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:1

Awaiting connection from pickup client...

Connection established!
Receiving data from client…
Wrong secret key.
Connection disconnected.

Next action?
Menu:
1) Wait for connection from client for next pickup request
2) Dequeue
Type an option:
```

Sample client programs (in sequence):

CLIENT 1

```
Please set a password.
Answer: WDIB4

What is your Team Name?
Answer: Team1

Group size?
Answer: 143

Establishing connection...
Connection established!
Data sent!

Waiting for the server to confirm your request…

Pickup confirmed! Please wait for pickup.

Connection disconnected.
```

CLIENT 2

```
Please set a password.
Answer: WDIB4

What is your Team Name?
Answer: Team2

Group size?
Answer: 151

Establishing connection...
Connection established!
Data sent!

Waiting for the server to confirm your request…

Pickup rejected!
Wrong password or request rejected.
Please try again.

Connection disconnected.
```

CLIENT 3

```
Please set a password.
Answer: WDIB4

What is your Team Name?
Answer: Team3
```

```
Group size?
Answer: 19

Establishing connection...
Connection established!
Data sent!

Waiting for the server to confirm your request…

Pickup rejected!
Wrong password or request rejected.
Please try again.

Connection disconnected.
```

CLIENT 4

```
Please set a password.
Answer: WDIB4

What is your Team Name?
Answer: Team4

Group size?
Answer: 15

Establishing connection...
Connection established!
Data sent!

Waiting for the server to confirm your request…

Pickup confirmed! Please wait for pickup to arrive.

Connection disconnected.
```

CLIENT 4

```
Please set a password.
Answer: FEWF

What is your Team Name?
Answer: Team5

Group size?
Answer: 10

Establishing connection...
Connection established!
Data sent!

Waiting for the server to confirm your request…
```

```
Pickup cancelled.
Wrong password or request rejected.
Please try again.

Connection disconnected.
```

[15]

Task 2

Name your Jupyter Notebook and save all parts for this task as

`TASK2_<index_number>_<name>.ipynb`

You will be writing a Math game program for Form Teachers to play with students in school. The game auto-generates math expressions and tracks scoring.

Task 2.1

Using a stack data structure, write a function `solver(expr)` that takes in a string of a mathematical expression `expr` such as `"((1*7)+6)"` and returns `13`. You may assume that the entire expression would never have spaces, and would always be enclosed in an opening and closing parenthesis "( )". Do not use the built-in function `eval()`.

Test your function with `solver("((1*7)+6)")` and show your output. [6]

Task 2.2

Write a function `generate_expression` that takes in an integer `operator_count` and returns a string of a mathematical expression which has the specified number of operators (i.e. `+ - * /`) in `operator_count`. The function should use **recursion** to form up the operators and operands. The operands, operators and positions of operands and operators should be random.

For example, `generate_expression(5)` would output `"(4*(6-(2+((1*7)+6))))"` and `generate_expression(5)` would output `"((8+(6+((2*4)-3)))+6)"`.

Test your function with `generate_expression(5)` and show your output. [7]

Task 2.3

Implement the following using object-oriented programming:

- `Person`, a class, which
  - initialises with these attributes
    - `name: string`
    - `gender: string` where male is `"M"` and female is `"F"`
    - `score: integer`
  - has the following methods
    - `display_info()` which displays the `Person`'s `name`, `gender` and `score`
      1. eg `"Nelson(M)'s score is 3."`

        - `attempts()` which
            1. uses the function `generate_expression` from Task 2.2 to generate and display a random math expression of 2 operators
            2. queries the user to give an answer rounded up to the nearest integer
            3. displays `"Good job!"` if the input is correct or `"Wrong answer. (Correct answer: <answer>)"` where `<answer>` is the correct answer.
            4. increases the score of the student by `1` if the answer is correct
            5. displays the user's latest `score`
- `Student`, a subclass of `Person`, which
    - also has the following attribute
        - `role: string` which is `"no role"` by default unless the `Student` has a class committee role such as `"chairperson"`
    - also has the following methods
        - `student_role()` which returns a string describing the `role` of the `Student`
- `FormTeacher`, a subclass of `Person`, which
    - also has the following methods
        - `display_info()` which uses polymorphism to display the `FormTeacher`'s information with salutation to the `FormTeacher`'s name
            1. eg: `"Ms. Norah's score is 0."` where "`Norah`" is her name, and "`Ms`." corresponds to her gender.

[14]

Task 2.4

Write driver code to test the earlier class you created. Also, create `groups` which is a list that uses a 2-dimensional array to store and associate each instance created below with his/her civics group. Use this 2-dimensional array to display the scores of all persons in each civics group indicating the student chairperson's name (if any). Test your code with the following steps in order:

- Create an instance of `Student` with `name "Melvin"` in civics group `5C35`
- Create an instance of `Student` with `name "Susan"` in civics group `5C35` whose role is `"chairperson"`
- Create an instance information of `FormTeacher` with `name "Norah"` in civics group `5C35`
- Create an instance of `Student` with `name "Ben"` in civics group `6C35`
- Create an instance of `FormTeacher` with name `"Jimmy"` in civics group `6C35`
- Display the information of Melvin
- Display the information of Susan
- Display the information of Norah
- Melvin attempts a math question
- Susan attempts a math question
- Jimmy attempts a math question
- Display the scores of all persons in each civics group with a header for each class

Here is a sample of an expected output:

```
Melvin(M)'s score is 0.
Susan(F)'s score is 0.
Ms. Norah's score is 0.

To  Melvin : ((5/7)+8) ?
Answer:1
Wrong answer. (Correct answer: 9 )
Total score is still 0.

To  Susan : (3*(5*6)) ?
Answer:90
Good job!
New total score for Susan is 1

To  Jimmy : ((6-2)-6) ?
Answer:3
Wrong answer. (Correct answer: -2 )
Total score is still 0.

5C35’s scores:
Melvin(M)'s score is 0.
Susan(F)'s score is 1. (Chairperson)
Ms. Norah's score is 0.

6C35’s scores:
Ben(M)'s score is 0.
Mr. Jimmy's score is 0.
```

Run your program and save your output. [7]

Task 3

In 2021, Singapore's Health Science Authority (HSA) recalled 18 brands of hand sanitisers due to high levels of acetaldehyde and/or methanol. The HSA keeps information on current hand sanitisers and uses it to monitor the types of chemical ingredients used to make the sanitisers.

Task 3.1

Create an SQL file to show the SQL code to create database `sanitisers.db` with the single table, `sanitisers`.

The table will have the following fields:

- `product_name` which is the primary key
- `active_ingredient`
- `alcohol-based`

Save your SQL code as
`TASK3_<index_number>_<name>.sql` [3]

Task 3.2

The text file, `sanitisers.txt`, contains data items for a number of sanitisers. It contains a header line. Each data item is separated by a comma, with each item data on a new line, as follows:
- product name
- active ingredient used to make the sanitiser product
- "Yes" or "No" to indicate if the product is alcohol-based

Write program code to read in the information from the text file, `sanitisers.txt`, and insert all the information into the `sanitisers.db` database. [3]

Run the program.

Save your program as
`TASK3_<index_number>_<name>.py`

Task 3.3

The information is to be displayed in a web browser.

Write a python program and the necessary files to create a web application that enables the list of sanitisers to be displayed.

For each record the web page should include the:
- product name
- ingredients used to make the sanitiser product
- "Yes" or "No" to indicate if the product is alcohol-based

Save your program as
`TASK3_<index_number>_<name>.py`
with any additional files/subfolders as needed in a folder named
`TASK3_<index_number>_<name>`
Run the web application and save the output of the program as
`TASK3_OUTPUT_<index_number>_<name>.html` [6]

Task 3.4

HSA wants a form on the web page that allows users to enter in the name of an active ingredient and, upon submission, will display all the information of the products with the matching active ingredient.

Update your application to include this form feature so that users will be able to use the form after seeing the list of sanitisers displayed as required in Task 3.3.

Run the web application, test your program with the ingredient "`Triclosan`" and save the output. [4]

Save, zip up and submit your program code and all related files for Task 3 as

`TASK3_<index_number>_<name>.zip`

Task 4

Name your Jupyter Notebook and save all parts for this task as

`TASK4_<index_number>_<name>.ipynb`

Task 4.1

Write a program to help staff of an events company to insert data into a NoSQL database `products` under the collection `balloons`.

The data is provided for you in `balloons.json` as well as in the table below where the first row are headers for the fields.

`design`	`amount`	`helium`	`colours`
car	88	no	red, yellow
cloud	14		blue, green
flower	75	yes	red, blue
bag	38	no	red, blue, black

Each colour in `colours` field should be an item in an array. [6]

Task 4.2

Write code to print the `amount` of the product with the design "`car`". [2]

Task 4.3

Write code to update the field `helium` to have the value "`no`" for all documents which do not have a field or value for `helium`.
[3]

Task 4.4

Write code to display the `design`(s) which do not contain helium and have colours that either contain `green` or do not contain `black`. [3]

Run the program.