Task 3

Hint: The following text file contains strings and text which you can simply copy-paste when attempting Task 3 and 4: `helper_for_Task3_and_4.txt`.

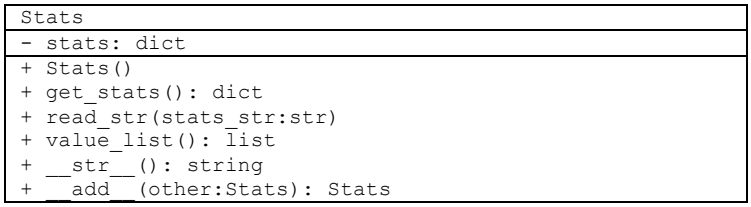
Star Road is a game created by the company Mohoyo. It is a turn-based game which allows you to control characters and battle with enemies, in Task 3 and 4, you are to aid the company to create an OOP class and a simple web application related to this game.

First, you need to create a class named Stats. The class supposed to store related stats for a character or a relic item, which would enhance the characters' statistics. The class contains 1 private attribute `stats` of type `dictionary`.

A typical input and output string would look like the following format (as one line):

```
'HP:1000;ATK:100;DEF:100;Speed:100;CRIT Rate:5;CRIT DMG:50;Break Effect:0;Effect
RES:0;Outgoing Healing:0;Energy Regen Rate:0;HP%:0;ATK%:0;DEF%:0'
```

Below is an UML class diagram for your reference.



Implement the classes based on the following descriptions:

Attributes/Methods	Description
`Character Class`	
`- stats: dict`	A dictionary which stores the value of stats, which has an initial value as `None`. Such as: `{'HP': None, 'ATK': None, 'DEF': None, ...}`
`+ Stats()`	Constructor of `Stats` class, which initialize the dictionary with all stats with `None` values.
`+ get_stats(): dict`	Getter method, which returns the current dict.
`+ read_str(stats_str:str)`	Method which read the stat from a `stats_str`, and store them into the dictionary based on the stat names and stat values. If the stat name is not currently found in the list of stats provided, do not add it (do not create any additional key-value pair in the dictionary). If the stat name is found, but the stat value is empty string `''` or `None`, treat it as a value of `0`. If the stat name is found, and stat value is an `integer` value, update the corresponding key-value pair in dictionary.

	For example, if the `stats_str` is `HP:1000;ATK:100`. Then it should update the key values pairs of `HP` and `ATK` to `1000` and `100` respectively.
`+ value_list(): list`	Return a list of stat values based on the stat names stored in the `stat_name_list`.
`+ __str__(): string`	String method for the class, to return a string with stat names and values in the following format '`name:value`', separate each stat by using an `';'`: `HP:1000;ATK:100;DEF:100;…`
`+ __add__(other:Stats): Stats`	Add all the stat values in the current `Stats` object (self) with another `Stats` object. Then return a `Stats` object as the result. If the stat value of one of the object is `None`, treat it as `0` when adding with the other object's stat value.

Save your program code as
`Task3.py` [12]

Task 4

In the Star Road game, there are relic sets, which are equipment items which characters can equip themselves with. Each relic sets have their unique set effects, which will boost the power of the characters wearing them.

For each relic sets, relics can be generated with random stats. Even if two relics are with the same name, they might have different stats associated with. For example, one of them may have stats that boost the attack value, the other relic may have stats that boost the energy regeneration rate.

Characters can choose to wear up to 6 relics, one at each position such as head, body, feet, etc.

You are tasked to design a relational database model and a web app to store and display the related information. You may assume that all inputs given are valid.

Task 4.1

The following information of each `Character` is stored:
`id` – auto increment integer value to keep track of ID of the Character.
`name` – name of the Character.
`element` – element of the Character's attacks, it should be one of the values:
`'Physical', 'Fire', 'Ice', 'Lightning', 'Wind', 'Quantum', 'Imaginary'`
`path` – path of the Character, it should be one of the values:
`'Destruction', 'The Hunt', 'Erudition', 'Harmony', 'Nihility', 'Preservation', 'Abundance'.`
`stats` – stats of a Character, which is a long string of text.

The following information of each `RelicSet` is stored:
`ID` – auto increment integer value to keep track of ID of the relic set.
`Name` – name of the relic set.
`SetEffect` – set effects when wearing more than 1 pieces of the same set.

The following information of each `Relic` is stored:
`ID` – auto increment integer value to keep track of the Relic.
`Name` – name of the relic.
`SetID` – Relic Set ID.
`Position` – position to be worn by the Character, it should be one of the values:
`'Head', 'Body', 'Hand', 'Feet', 'Planar Sphere', 'Link Rope'`
`Stats` – stats of a Relic, which is a long string of text.

The following information of each `Equipment` is stored:
`CharacterID` – Character ID.
`RelicID` – Relic ID.

The information is to be stored in above mentioned tables:
`Character`
`RelicSet`
`Relic`
`Equipment`

Task 4.1

Create an SQL file called `Task4_1.sql` to show the SQL code needed to create the database `star_road.db` with the above tables.

The tables `Character, RelicSet, Relic` must use `ID` as its primary key. The table `Equipment` must use `CharacterID` and `RelicID` as its primary key.

The `SetID` in table `Relic` must refer to `ID` in `RelicSet` table as foreign key. `CharacterID` and `RelicID` in table `Equipment` must refer to `ID` in `Character` table and `ID` in `Relic` table as foreign keys.

Save your SQL code as
`Task4_1.sql`

[6]

Task 4.2

The following files contains the past data records. The first row of each file contains the header of the respective columns. Each row in the files is a comma-separated list of information.

`characters.csv`
`relic_sets.csv`
`relics.csv`
`equipments.csv`

Write a Python program to insert all information from the files into the database `star_road.db`. Run the program.

Save your program code as
`Task4_2.py` [5]

Task 4.3

You are tasked to implement a function to display the number of relics of each relic set that are current equipped by the characters. Query and display a list of data with the following fields as shown in the table, sorted in the descending order by the Count value.

Note: *To make it simple, if 1 character wears 4 relic pieces from the same relic set, count 4 times.*

Set Name	Count
…	…

Write the SQL code required.

Save this code as
`Task4_3.sql` [5]

Task 4.4

You are required to implement a function to search and display a summary of the equipped relics of a character based on the character name.

You should display the character name, followed by a table containing the following columns:
`Relic Name, Position, Relic Set Name, Set Effect, Stats`

Here's a sample result page for your reference:

Result Page

The character is Seele.

Relic Name	Position	Relic Set Name	Set Effect	Stats
Genius's Ultraremote Sensing Visor	Head	HP:112;ATK:None;DEF:16;Speed:2;CRIT Rate:None;CRIT DMG:None;Break Effect:5;Effect RES:None;Outgoing Healing:None;Energy Regen Rate:None;HP%:None;ATK%:3;DEF%:None	Genius of Brilliant Stars	Two Set Effect: Increases Quantum DMG by 10%. | Four Set Effect: When the wearer deals DMG to the target enemy, ignores 10% DEF. If the target enemy has Quantum Weakness, the wearer additionally ignores 10% DEF.
Genius's Frequency Catcher	Body	HP:33;ATK:56;DEF:None;Speed:2;CRIT Rate:2;CRIT DMG:None;Break Effect:None;Effect RES:None;Outgoing Healing:None;Energy Regen Rate:None;HP%:None;ATK%:None;DEF%:4	Genius of Brilliant Stars	Two Set Effect: Increases Quantum DMG by 10%. | Four Set Effect: When the wearer deals DMG to the target enemy, ignores 10% DEF. If the target enemy has Quantum Weakness, the wearer additionally ignores 10% DEF.
Genius's Metafield Suit	Hand	HP:33;ATK:None;DEF:None;Speed:None;CRIT Rate:None;CRIT DMG:5;Break Effect:None;Effect RES:None;Outgoing Healing:None;Energy Regen Rate:None;HP%:3;ATK%:None;DEF%:4	Genius of Brilliant Stars	Two Set Effect: Increases Quantum DMG by 10%. | Four Set Effect: When the wearer deals DMG to the target enemy, ignores 10% DEF. If the target enemy has Quantum Weakness, the wearer additionally ignores 10% DEF.
Genius's Gravity Walker	Feet	HP:33;ATK:16;DEF:None;Speed:None;CRIT Rate:None;CRIT DMG:None;Break Effect:None;Effect RES:None;Outgoing Healing:None;Energy Regen Rate:None;HP%:3;ATK%:None;DEF%:4	Genius of Brilliant Stars	Two Set Effect: Increases Quantum DMG by 10%. | Four Set Effect: When the wearer deals DMG to the target enemy, ignores 10% DEF. If the target enemy has Quantum Weakness, the wearer additionally ignores 10% DEF.
Herta's Space Station	Planar Sphere	HP:None;ATK:None;DEF:0;Speed:2;CRIT Rate:None;CRIT DMG:5;Break Effect:5;Effect RES:None;Outgoing Healing:None;Energy Regen Rate:None;HP%:None;ATK%:3;DEF%:None	Space Sealing Station	Two Set Effect: Increases the wearer's ATK by 12%. When the wearer's SPD reaches 120 or higher, the wearer's ATK increases by an extra 12%.
Herta's Wandering Trek	Link Rope	HP:None;ATK:None;DEF:None;Speed:None;CRIT Rate:None;CRIT DMG:5;Break Effect:None;Effect RES:3;Outgoing Healing:None;Energy Regen Rate:None;HP%:3;ATK%:None;DEF%:4	Space Sealing Station	Two Set Effect: Increases the wearer's ATK by 12%. When the wearer's SPD reaches 120 or higher, the wearer's ATK increases by an extra 12%.

Search Character Again

Save all files and folders under the directory `Task4_4`.

Run the web application with a character name input of `Seele`.
Then save the output of the program as `Task4_4.html`. [12]

Task 4.5 [Bonus – Challenge by Choice]

You are encouraged to copy and paste your files from `Task4_4` to the new directory `Task4_5` before attempting this task.

You are tasked to display the information of this page with better clarity and format:

- Separate the repeated set name and their set effects.
- Display stats in individual columns.
- Calculate the overall stats by adding the base stats of the character with all the stats the character has gained from the equipped relics.

Here's a sample page for character `Seele`:

Result Page

The character is Seele.

Set Name: Genius of Brilliant Stars

Set Effect: Two Set Effect: Increases Quantum DMG by 10%. | Four Set Effect: When the wearer deals DMG to the target enemy, ignores 10% DEF. If the target enemy has Quantum Weakness, the wearer additionally ignores 10% DEF.

Set Name: Space Sealing Station

Set Effect: Two Set Effect: Increases the wearer's ATK by 12%. When the wearer's SPD reaches 120 or higher, the wearer's ATK increases by an extra 12%.

Relic Name	Position	HP	ATK	DEF	Speed	CRIT Rate	CRIT DMG	Break Effect	Effect RES	Outgoing Healing	Energy Regen Rate	HP%	ATK%	DEF%
Seele	Character	126	87	49	115	5	50							
Genius's Ultraremote Sensing Visor	Head	112		16	2			5					3	
Genius's Frequency Catcher	Body	33	56		2	2								4
Genius's Metafield Suit	Hand	33					5					3		4
Genius's Gravity Walker	Feet	33	16									3		4
Herta's Space Station	Planar Sphere				2		5	5					3	
Herta's Wandering Trek	Link Rope						5		3			3		4
Overall		211	72	16	6	2	15	10	3	0	0	9	6	16

Search Character Again

Save all files and folders under the directory `Task4_5`.

[2]

- End of Paper -