Python Exercises I

Write one script for each of the following exercises.

Exercise 1: Celestial event counter

The following list records all of the celestial events that were observed over a month:

```python
events = [
    "meteor shower", "lunar eclipse", "solar flare", "meteor shower",
    "comet sighting", "meteor shower", "lunar eclipse", "comet sighting"
]
```

Write a program that iterates through the list and builds a dictionary to count the frequency of each event. (The keys should be the names of the events, and the values should be the numbers of occurrences). After constructing your dictionary, print each event along with how many times it occurred.

Exercise 2: Observational methods

The following dictionary maps observation methods (the keys) to astronomical objects that have been observed with those methods (the values).

```python
observation_methods = {
    "optical": ["M31", "M42", "M45", "NGC224"],
    "radio": ["M1", "M87", "NGC5128"],
    "infrared": ["M31", "NGC1976", "M45"],
    "x-ray": ["M1", "NGC1275"]
}
```

Write a program that:

1. Prompts the user to enter the name of an astronomical object (e.g., `"M31"`).
2. Searches through the `observation_methods` dictionary to find all observation methods that list that object.
3. Prints out the methods where the object appears.
4. Displays an appropriate message if the object is not found under any observation method.

Exercise 3: Password strength checker

Write a program that prompts the user to enter a password, then evaluates the strength of the password that was entered. The password must meet the following criteria to be “strong”:

- At least 8 characters long.
- Contains both uppercase and lowercase letters.
- Includes at least one digit.
- Includes at least one of these special characters: !, @, _, $, %

If the password meets all these criteria, print `"Strong password!"`. Otherwise, print messages indicating which criteria were not met.

Exercise 4: Distance converter

Astronomers often convert distances between light years (ly) and parsecs (pc), where 1 parsec is approximately 3.26 light years. Using the following list of distances in light years:

```python
distances_ly = [10.5, 32.7, 4.3, 65.2, 21.1]
```

Write a script that:

1. Prompts the user to enter a minimum distance threshold (in light years).
2. Creates a new list containing only distances greater than or equal to the threshold.
3. Converts these filtered distances to parsecs.
4. Prints the (filtered) distances in both light-years and parsecs.

Exercise 5: Debugging

The code below is supposed to convert the distances to three galaxies from megaparsecs to light years. Run the code and analyze the resulting error traces. Identify and fix three errors that are preventing this code from running properly.

```python
galaxies = {"Andromeda": 0.78, "Triangulum"0.85, "Whirlpool": 7.0}
conversion_factor = 3.26e6

for galaxy in galaxies:
    ly_distance = galaxy * conversion_factor
    print("The galaxy " + galaxy + " is " + ly_distance + " light years away.")
```