

### Q1.

Count the number of times each unique genre occurs.

1. Create an empty dictionary named `genre_counting`.
2. Loop through the `apps_data` list of lists (make sure you don't include the header row). For each iteration of the loop:
  - Assign the genre to a variable named `genre`. The genre comes as a string and has the index number `11`.
  - Check whether `genre` exists as a key in `genre_counting`.
    - If it exists, then increment the dictionary value at that key by `1` (the key is equivalent to the value stored in `genre`).
    - Else, create a new key-value pair in the dictionary, where the dictionary key is `genre` and the dictionary value is `1`.
3. Outside the loop, print `genre_counting` and try to determine what's the most common app genre in our data set.

#### Script.py

```
opened_file = open('AppleStore.csv', encoding="utf8")
from csv import reader
read_file = reader(opened_file)
apps_data = list(read_file)
```

### Q2.

1. Transform the frequencies inside `content_ratings` to proportions and percentages while creating separate dictionaries for each.
  - Assign the dictionary storing proportions to a variable named `c_ratings_proportions`.
  - Assign the dictionary storing percentages to a variable named `c_ratings_percentages`.
2. Optional challenge: try to solve this exercise using a single for loop (solution to this challenge provided).

#### Script.py

```
content_ratings = {'4+': 4433, '12+': 1155, '9+': 987, '17+': 622}
total_number_of_apps = 7197
```

### Q3.

1. Begin by finding the minimum and maximum value in the `rating_count_tot` column.

- Extract the values in the `rating_count_tot` column (index number `5`) in a separate list (don't forget to convert to integer or float).
- Find out the minimum and maximum value of that list using the `min()` and the `max()` commands.

2. Based on the minimum and maximum value you've found, choose a few intervals (try to choose five intervals or less).

3. Once you've chosen the intervals, compute the frequency of apps for each interval. Store the frequency table in a dictionary.

- Create a dictionary with intervals as dictionary keys and zeros as dictionary values.
- Loop through the `apps_data` data set. Count the frequency of each interval using an `if` statement followed by a series of `elif` clauses.

4. Inspect the frequency table and analyze the results.

#### Script.py

```
opened_file = open('AppleStore.csv', encoding="utf8")
from csv import reader
read_file = reader(opened_file)
apps_data = list(read_file)
```