Reference:
https://www.edureka.co/blog/data-structures-in-python/

Exercise 2: Remove and add item in a list

Write a program to remove the item present at index 4 and add it to the 2nd position and at the end of the list.

Given:

```
list1 = [54, 44, 27, 79, 91, 41]
```

Expected Output:

```
List After removing element at index 4  [34, 54, 67, 89, 43, 94]
List after Adding element at index 2  [34, 54, 11, 67, 89, 43, 94]
List after Adding element at last  [34, 54, 11, 67, 89, 43, 94, 11]
```

+ Show Hint

Use the list methods, `pop()`, `insert()` and `append()`

Exercise 3: Slice list into 3 equal chunks and reverse each chunk

Given:

```
sample_list = [11, 45, 8, 23, 14, 12, 78, 45, 89]
```

Expected Outcome:

```
Chunk  1 [11, 45, 8]
After reversing it  [8, 45, 11]
Chunk  2 [23, 14, 12]
After reversing it  [12, 14, 23]
Chunk  3 [78, 45, 89]
After reversing it  [89, 45, 78]
```

Exercise 4: Count the occurrence of each element from a list

Write a program to iterate a given list and count the occurrence of each element and create a dictionary to show the count of each element.

Given:

```
sample_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]
```

Expected Output:

```
Printing count of each item   {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
```

Exercise 5: Create a Python set such that it shows the element from both lists in a pair

Given:

```
first_list = [2, 3, 4, 5, 6, 7, 8]
second_list = [4, 9, 16, 25, 36, 49, 64]
```

Expected Output:

```
Result is  {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}
```

+ Show Hint

Use the `zip()` function. This function takes two or more iterables (like list, dict, string), aggregates them in a tuple, and returns it.

Exercise 6: Find the intersection (common) of two sets and remove those elements from the first set

See: Python Set

Given:

```
first_set = {23, 42, 65, 57, 78, 83, 29}
second_set = {57, 83, 29, 67, 73, 43, 48}
```

Expected Output:

```
Intersection is  {57, 83, 29}
First Set after removing common element  {65, 42, 78, 23}
```

+ Show Hint

- Use the `intersection()` and `remove()` method of a set

Exercise 7: Checks if one set is a subset or superset of another set. If found, delete all elements from that set

Given:

```
first_set = {27, 43, 34}
second_set = {34, 93, 22, 27, 43, 53, 48}
```

Expected Output:

```
First set is subset of second set - True
Second set is subset of First set -  False

First set is Super set of second set -  False
Second set is Super set of First set -  True

First Set  set()
Second Set  {67, 73, 43, 48, 83, 57, 29}
```

+ Show Hint

Use the below methods of a set class

- `issubset()`
- `issuperset()`
- `clear()`

Exercise 8: Iterate a given list and check if a given element exists as a key's value in a dictionary. If not, delete it from the list

Given:

```
roll_number = [47, 64, 69, 37, 76, 83, 95, 97]
sample_dict = {'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}
```

Expected Outcome:

```
After removing unwanted elements from list [47, 69, 76, 97]
```

Exercise 10: Remove duplicates from a list and create a tuple and find the minimum and maximum number

Given:

```
sample_list = [87, 45, 41, 65, 94, 41, 99, 94]
```

Expected Outcome:

```
unique items [87, 45, 41, 65, 99]
tuple (87, 45, 41, 65, 99)
min: 41
max: 99
```