

Python Data Structure Exercise

1: Given two lists create a third list by picking an odd-index element from the first list and even index elements from the second.

Given:

```
listOne = [3, 6, 9, 12, 15, 18, 21]
listTwo = [4, 8, 12, 16, 20, 24, 28]
```

Expected Output:

```
Element at odd-index positions from list one
[6, 12, 18]
Element at even-index positions from list two
[4, 12, 20, 28]
Printing Final third list
[6, 12, 18, 4, 12, 20, 28]
```

2: Given a list, remove the element 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:

```
Original list [34, 54, 67, 89, 11, 43, 94]
```

```
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]
```

3: Given a list slice it into 3 equal chunks and reverse each chunk

Given:

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

Expected Outcome:

```
Original list [11, 45, 8, 23, 14, 12, 78, 45, 89]
```

```
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]
```

4: Iterate a given list and count the occurrence of each element and create a dictionary to show the count of each element

Expected Output:

```
Original list [11, 45, 8, 11, 23, 45, 23, 45, 89]
```

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

5: Given a two list of equal size create a Python set such that it shows the element from both lists in the pair

Expected Output:

```
First List [2, 3, 4, 5, 6, 7, 8]
```

```
Second List [4, 9, 16, 25, 36, 49, 64]
```

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

6: Given the following two sets find the intersection and remove those elements from the first set

Expected Output:

```
First Set {65, 42, 78, 83, 23, 57, 29}
```

```
Second Set {67, 73, 43, 48, 83, 57, 29}
```

```
Intersection is {57, 83, 29}
```

```
First Set after removing common element {65, 42, 78, 23}
```

7: Given two sets, Checks if One Set is a subset or superset of another Set. if the subset is found delete all elements from that set

Given:

```
firstSet = {27, 43, 34}
```

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

Expected Output:

```
First Set {57, 83, 29}
```

```
Second Set {67, 73, 43, 48, 83, 57, 29}
```

```
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}
```

8: Iterate a given list and Check if a given element already exists in a dictionary as a key's value if not delete it from the list

Given:

```
rollNumber = [47, 64, 69, 37, 76, 83, 95, 97]
```

```
sampleDict ={'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}
```

Expected Outcome:

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

9: Given a dictionary get all values from the dictionary and add them to a list but don't add duplicates

Given:

```
speed ={'jan':47, 'feb':52, 'march':47, 'April':44, 'May':52, 'June':53, 'july':54, 'Aug':44, 'Sept':54}
```

Expected Outcome:

```
[47, 52, 44, 53, 54]
```

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

Given:

```
sampleList = [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
```