Q-1

```
numbers = [1, 2, 3, 4, 2, 5, 2]
value_to_remove = 2

while value_to_remove in numbers:
    numbers.remove(value_to_remove)

print(numbers)
```

```
[1, 3, 4, 5]
```

Start coding or generate with AI.

Q-2

```
# List of strings
words = ["Data", "Science", "is", "awesome"]

# Use join() to concatenate with spaces
result = " ".join(words)

print(result)
```

```
Data Science is awesome
```

Q-3

```
# List of integers
numbers = [1, 2, 3, 4, 5]

# Reverse the list in place
numbers.reverse()

print(numbers)
```

```
[5, 4, 3, 2, 1]
```

Q-4

```
# List of numbers
numbers = [3, 1, 4, 1, 5, 9, 2]

# Sort the list in descending order
numbers.sort(reverse=True)

print(numbers)
```

```
[9, 5, 4, 3, 2, 1, 1]
```

Q-5

```
# Two lists of numbers
list1 = [1, 2, 3, 4]
list2 = [3, 4, 5, 6]

# Combine the lists and remove duplicates
combined_list = list(set(list1 + list2))

print(combined_list)
```

```
[1, 2, 3, 4, 5, 6]
```

Q-6

```
# Tuple of integers
numbers_tuple = (10, 20, 30, 40, 50)

# Convert the tuple into a list
numbers_list = list(numbers_tuple)

# Remove the first and last elements
modified_list = numbers_list[1:-1]

print(modified_list)
```

```
[20, 30, 40]
```

Q-7

```
# List of tuples
tuples_list = [(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd')]

# Extract the first elements using tuple unpacking
first_elements = [x for x, _ in tuples_list]

print(first_elements)
```

```
[1, 2, 3, 4]
```

Q-8

```
# Two tuples
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)

# Combine the tuples
combined_tuple = tuple1 + tuple2

print(combined_tuple)
```

```
(1, 2, 3, 4, 5, 6)
```

Q-9

```
# Tuple of numbers
numbers_tuple = (10, 20, 5, 40, 15)

# Find the maximum and minimum values
max_value = max(numbers_tuple)
min_value = min(numbers_tuple)

print(f"Maximum Value: {max_value}")
print(f"Minimum Value: {min_value}")
```

```
Maximum Value: 40
Minimum Value: 5
```

Q-10

```
# Tuple of strings
strings_tuple = ("apple", "banana", "cherry")

# Convert the tuple into a single string with comma separation
result = ", ".join(strings_tuple)

print(result)
```

```
apple, banana, cherry
```

Q-11

```
# Two sets
set1 = {1, 2, 3}
set2 = {3, 4, 5}

# Using the | operator to get the union
union_set = set1 | set2

# Alternatively, using the .union() method
# union_set = set1.union(set2)

print(union_set)
```

```
{1, 2, 3, 4, 5}
```

Q-12

```
# Two sets
set1 = {1, 2, 3, 4}
set2 = {3, 4, 5, 6}

# Using the .intersection() method
intersection_set = set1.intersection(set2)

# Alternatively, using the & operator
# intersection_set = set1 & set2

print(intersection_set)
```

```
{3, 4}
```

Q-13

```
# Two sets
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 6}

# Using the .difference_update() method
set1.difference_update(set2)

# Alternatively, using the -= operator
# set1 -= set2

print(set1)
```

```
{1, 2, 5}
```

Q-14

```
# Two sets
set1 = {1, 2, 3, 4, 5}
set2 = {3, 4, 6}

# Using the .difference() method
difference_set = set1.difference(set2)

# Alternatively, using the - operator
# difference_set = set1 - set2

print(difference_set)
```

```
{1, 2, 5}
```

Q-15

```
# List of integers with duplicates
numbers_list = [1, 2, 3, 2, 4, 5, 1]

# Convert the list to a set to remove duplicates
numbers_set = set(numbers_list)

# Convert the set back to a list
unique_numbers_list = list(numbers_set)

print(unique_numbers_list)
```

```
[1, 2, 3, 4, 5]
```

Q-16

```
# Two dictionaries
dict1 = {'a': 1, 'b': 2}
dict2 = {'b': 3, 'c': 4}

# Create a copy of the first dictionary
merged_dict = dict1.copy()

# Update the copy with the second dictionary
merged_dict.update(dict2)

print(merged_dict)
```

```
{'a': 1, 'b': 3, 'c': 4}
```

Q-17

```
# Sample dictionary
my_dict = {'a': 1, 'b': 2, 'c': 3}

# Extract all values and convert to a list
values_list = list(my_dict.values())

print(values_list)
```

```
[1, 2, 3]
```

Start coding or generate with AI.