

Assignment 4 Solution

Submitted by : Yameen Ali

[Go to Github repository](#)

Question 1

Lists

a) Create an empty list and populate it with integers.

```
In [25]: int_list = [12,24,34,67,30,5,56]
```

b) Perform the following operations on the list:

1. Add an element at the beginning and end of the list

```
In [26]: # Add an element at the beginning of the list
int_list.insert(0, 13)

# Add an element at the end of the list
int_list.append(14)

print(int_list)
```

```
[13, 12, 24, 34, 67, 30, 5, 56, 14]
```

2. Remove an element from the list by its value.

```
In [27]: # Remove an element from the list by its value
int_list.remove(12) # Removes the element with the value 12

print(int_list)
```

```
[13, 24, 34, 67, 30, 5, 56, 14]
```

3. Sort the list in ascending order.

```
In [39]: # Sort the list in ascending order
int_list.sort()
print(int_list)
```

```
[5, 13, 14, 24, 30, 34, 56, 67]
```

4. Check if a specific element exists in the list.

```
In [36]: check_int = 5
if check_int in int_list:
    print(f"Yes {check_int} exists")
else:
    print("not exists")
```

```
Yes 5 exists
```

5. Display the final list after each operation.

```
In [37]: print(int_list)
```

```
[5, 13, 14, 24, 30, 34, 56, 67]
```

Question 2

Tuples

a) Create a tuple of your favorite foods.

```
In [42]: favorite_foods = ('pizza','biryani','pasta','icecream')
```

b) Access and print the elements of the tuple.

```
In [43]: # Access and print each element of the tuple
for food in favourite_foods:
    print(food)
```

```
pizza
biryani
pasta
icecream
```

c) Convert the tuple to a list and modify one of the elements.

```
In [44]: # Convert the tuple to a list
favourite_foods_list = list(favourite_foods)

# Modify one of the elements in the list
favourite_foods_list[1] = 'burgers'

print(favourite_foods_list)
```

```
['pizza', 'burgers', 'pasta', 'icecream']
```

d) Convert the modified list back to a tuple.

```
In [49]: favourite_foods_tuple = tuple(favourite_foods_list)
```

e) Display the final tuple.

```
In [48]: print(favourite_foods_tuple)

('pizza', 'burgers', 'pasta', 'icecream')
```

Question 3

Sets

a) Create two sets containing random integers.

```
In [52]: # set one
set1 = {34, 67, 45, 43, 12}

# set two
set2 = {56, 38, 23, 12, 43}

print("Set 1:", set1)
print("Set 2:", set2)
```

```
Set 1: {34, 67, 43, 12, 45}
Set 2: {38, 23, 56, 43, 12}
```

b) Perform set operations on the sets, including union, intersection, and difference..

```
In [56]: # Perform set operations
union_set = set1.union(set2)           # Union of set1 and set2
intersection_set = set1.intersection(set2) # Intersection of set1 and set2
difference_set = set1.difference(set2)    # Set difference of set1 - set2

# Print the results
print("Set 1:", set1)
print("Set 2:", set2)
print(" ")
print("Union of set1 and set2:", union_set)
print("Intersection of set1 and set2:", intersection_set)
print("Set difference of set1 - set2:", difference_set)
```

```
Set 1: {34, 67, 43, 12, 45}
Set 2: {38, 23, 56, 43, 12}
```

```
Union of set1 and set2: {34, 67, 38, 43, 12, 45, 23, 56}
Intersection of set1 and set2: {43, 12}
Set difference of set1 - set2: {34, 67, 45}
```

c) Determine if one set is a subset or superset of the other.

```
In [58]: # Check if set1 is a subset of set2
is_subset = set1.issubset(set2)

# Check if set1 is a superset of set2
is_superset = set1.issuperset(set2)

# Print the results
if is_subset:
    print("Set 1 is a subset of Set 2")
else:
    print("Set 1 is not a subset of Set 2")

if is_superset:
    print("Set 1 is a superset of Set 2")
else:
    print("Set 1 is not a superset of Set 2")
```

Set 1 is not a subset of Set 2
Set 1 is not a superset of Set 2

Question 4

Dictionaries

a) Create a dictionary with keys representing fruits and values representing their prices.

```
In [64]: fruit_prices = {
    "apple": 1.99,
    "banana": 0.99,
    "orange": 0.79,
    "grape": 2.49,
    "kiwi": 1.29
}

print(fruit_prices)

{'apple': 1.99, 'banana': 0.99, 'orange': 0.79, 'grape': 2.49, 'kiwi': 1.29}
```

b) Add new key-value pairs to the dictionary

```
In [65]: # add new key value pair to the dictionary
fruit_prices["mango"] = 2.99
fruit_prices["pineapple"] = 3.22

# print updated dictionary
print(fruit_prices)

{'apple': 1.99, 'banana': 0.99, 'orange': 0.79, 'grape': 2.49, 'kiwi': 1.29, 'mango': 2.99, 'pineapple': 3.22}
```

c) Access and print the value of a specific key.

```
In [67]: print(fruit_prices['mango'])

2.99
```

d) Modify the value of an existing key.

```
In [68]: # Modify the value of an existing key
fruit_prices["banana"] = 1.49

# Print the updated dictionary
print(fruit_prices)

{'apple': 1.99, 'banana': 1.49, 'orange': 0.79, 'grape': 2.49, 'kiwi': 1.29, 'mango': 2.99, 'pineapple': 3.22}
```

e) Delete a key-value pair from the dictionary.

```
In [70]: del fruit_prices['mango']

# Print the updated dictionary
print(fruit_prices)
```

```
{'apple': 1.99, 'banana': 1.49, 'orange': 0.79, 'grape': 2.49, 'kiwi': 1.29, 'pineapple': 3.22}
```

f) Display the final dictionary after each operation.

```
In [71]: print(fruit_prices)
```

```
{'apple': 1.99, 'banana': 1.49, 'orange': 0.79, 'grape': 2.49, 'kiwi': 1.29, 'pineapple': 3.22}
```

Question 5

You are given a dataset containing information about employees in a company. The dataset includes the following fields for each employee: employee ID (integer), name (string), department (string), and salary (float)

Using Python data structures

a) Create a list called "employees" and populate it with three employee records, each represented as a dictionary

```
In [77]: employees = [
    {"employee_id": 1, "name": "Yameen Ali", "department": "Data Scientist", "salary": 90000.0},
    {"employee_id": 2, "name": "Abdul Rafay", "department": "Software Engineer", "salary": 60000.0},
    {"employee_id": 3, "name": "Saad Malik", "department": "HR", "salary": 50000.0}
]
```

b) Calculate the average salary of all employees in the "employees" list and print the result:

```
In [81]: # Calculate the average salary of all employees
total_salary = sum(employee["salary"] for employee in employees)
average_salary = total_salary / len(employees)
print("Average Salary of Employees:", average_salary)
```

Average Salary of Employees: 66666.66666666667

c) Convert the list of employees into a set called "unique_departments" to obtain a set of unique department names:

```
In [82]: # Convert the list of employees into a set of unique department names
unique_departments = {employee["department"] for employee in employees}
print("Unique Departments:", unique_departments)
```

Unique Departments: {'Data Scientist', 'Software Engineer', 'HR'}

d) Create a tuple called "highest_paid_employee" and assign the details (dictionary) of the employee with the highest salary to it:

```
In [83]: # Find the employee with the highest salary
highest_paid_employee = max(employees, key=lambda x: x["salary"])
print("Highest Paid Employee:", highest_paid_employee)
```

Highest Paid Employee: {'employee_id': 1, 'name': 'Yameen Ali', 'department': 'Data Scientist', 'salary': 90000.0}

e) Display the final "employees" list, "unique_departments" set, and "highest_paid_employee" tuple:

```
In [84]: # Display the final results
print("Employees List:")
for employee in employees:
    print(employee)
print("\nUnique Departments:", unique_departments)
print("\nHighest Paid Employee:", highest_paid_employee)
```

Employees List:

```
{'employee_id': 1, 'name': 'Yameen Ali', 'department': 'Data Scientist', 'salary': 90000.0}
{'employee_id': 2, 'name': 'Abdul Rafay', 'department': 'Software Engineer', 'salary': 60000.0}
{'employee_id': 3, 'name': 'Saad Malik', 'department': 'HR', 'salary': 50000.0}
```

Unique Departments: {'Data Scientist', 'Software Engineer', 'HR'}

Highest Paid Employee: {'employee_id': 1, 'name': 'Yameen Ali', 'department': 'Data Scientist', 'salary': 90000.0}

