https://github.com/Harishwar-reddi/ICP-2

Q1

```
[2]  ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

     #min and max age
     sorted_ages = sorted(ages)
     min_age = sorted_ages[0]
     max_age = sorted_ages[-1]
     print(f"Min age: {min_age}")
     print(f"Max age: {max_age}")

     # min age and the max age again to the list
     ages.extend([min_age, max_age])

     # median age
     n = len(sorted_ages)
     if n % 2 == 0:
         median_age = (sorted_ages[n//2 - 1] + sorted_ages[n//2]) / 2
     else:
         median_age = sorted_ages[n//2]
     print(f"Median age: {median_age}")

     #average age
     average_age = sum(ages) / len(ages)
     print(f"Average age: {average_age}")

     #range of the ages
     age_range = max_age - min_age
     print(f"Range of ages: {age_range}")
```

```
Min age: 19
Max age: 26
Median age: 24.0
Average age: 22.75
Range of ages: 7
```

Q2

```
dog = {}

# Add name, color, breed, legs, age to the dog dictionary
dog['name'] = 'nick'
dog['color'] = 'Brown'
dog['breed'] = 'Golden Retriever'
dog['legs'] = 4
dog['age'] = 7

# Create a student dictionary and add required keys
student = {
    'first_name': 'Bunny',
    'last_name': 'Re',
    'gender': 'Male',
    'age': 21,
    'marital_status': 'Single',
    'skills': ['Front-End', 'Full-Stack'],
    'country': 'USA',
    'city': 'Kansas city',
    'address': '115 Broad St'
}

#length of the student
print(f"Length of student dictionary: {len(student)}")

#value of skills and data type
skills_value = student['skills']
print(f"Skills: {skills_value}, Data type: {type(skills_value)}")
student['skills'].extend(['java', 'Java Developer'])
#dictionary keys as a list
keys_list = list(student.keys())
print(f"Keys: {keys_list}")
#dictionary values as a list
values_list = list(student.values())
print(f"Values: {values_list}")
```

```
Length of student dictionary: 9
Skills: ['Front-End', 'Full-Stack'], Data type: <class 'list'>
Keys: ['first_name', 'last_name', 'gender', 'age', 'marital_status', 'skills', 'country', 'city', 'address']
Values: ['Bunny', 'Re', 'Male', 21, 'Single', ['Front-End', 'Full-Stack', 'java', 'Java Developer'], 'USA', 'Kansas city', '115 Broad St']
```

Q3

```python
it_companies = {'facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

# Length
print(len(it_companies))

# Add Twitter
it_companies.add('Twitter')

# Add multiple companies
it_companies.update(['Calix', 'Meta'])

# Remove 'Calix'
it_companies.discard('Calix')

# Union and Intersection of A and B
print(A.union(B))
print(A.intersection(B))

print(A.issubset(B))
print(A.isdisjoint(B))

# Symmetric difference between A and B
print(A.symmetric_difference(B))

# Convert age to set and compare lengths
age_set = set(age)
print(len(age), len(age_set))

# Display size before deletion attempt
print(f"Size of it_companies before deletion: {len(it_companies)} items.")

try:
    del it_companies
except NameError:
    print("The set it_companies does not exist.")

# Attempt to display size after deletion
try:
    print(f"Size of it_companies after deletion: {len(it_companies)} items.")
except NameError:
    print("The set it_companies has been deleted and does not exist anymore.")
```

```
7
{19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26}
True
False
{27, 28}
8 5
Size of it_companies before deletion: 9 items.
The set it_companies has been deleted and does not exist anymore.
```

Q4

```python
class Employee:
    employee_count = 0
    total_salary = 0

    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
        Employee.employee_count += 1
        Employee.total_salary += salary

    @classmethod
    def average_salary(cls):
        return cls.total_salary / cls.employee_count


class FulltimeEmployee(Employee):
    pass

#instances
e1 = Employee("Bunny", 50000)
f1 = FulltimeEmployee("Chikki", 60000)

#data
print(f"Total Employees: {Employee.employee_count}")
print(f"Average Salary: {Employee.average_salary()}")
```

```
Total Employees: 2
Average Salary: 55000.0
```