NAME = PRANAV SINHA SAPID = 590029558 BATCH = 80

Q1. Scan n values in range 0-3 and print the number of times each value has occurred.

```
n = int(input("Enter number of values: "))


counts = [0, 0, 0, 0]


print(f"Enter {n} values (each between 0 and 3):")
for _ in range(n):
    val = int(input())
    if 0 <= val <= 3:
        counts[val] += 1
    else:
        print("Invalid input! Value must be between 0 and 3.")


for i in range(4):
    print(f"{i} occurred {counts[i]} times")
```

```
Enter number of values: 2
Enter 2 values (each between 0 and 3):
12
Invalid input! Value must be between 0 and 3.
1
0 occurred 0 times
1 occurred 1 times
2 occurred 0 times
3 occurred 0 times
```

Q2. Create a tuple to store n numeric values and find average of all values.

```
n = int(input("Enter number of values: "))


values = tuple(float(input(f"Enter value {i+1}: ")) for i in range(n))


average = sum(values) / n if n != 0 else 0

print("Values entered:", values)
print("Average of values:", average)
```

```
Enter number of values: 4
Enter value 1: 10
Enter value 2: 20
Enter value 3: 30
Enter value 4: 40
Values entered: (10.0, 20.0, 30.0, 40.0)
Average of values: 25.0
```

Q3.WAP to input a list of scores for N students in a list data type. Find the score of the runner up and print the output. Sample Input N = 5 Scores= 2 3 6 6 5 Sample output 5 Note: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.

```
N = int(input("Enter number of students: "))

scores = list(map(int, input(f"Enter {N} scores separated by spaces: ").split()))


unique_scores = list(set(scores))

unique_scores.sort(reverse=True)


if len(unique_scores) >= 2:
    runner_up = unique_scores[1]
    print("Runner-up score:", runner_up)
else:
    print("No runner-up score exists")
```

```
Enter number of students: 5
Enter 5 scores separated by spaces: 2 3 6 6 5
Runner-up score: 5
```

Q4.Create a dictionary of n persons where key is name and value is city. a) Display all names b) Display all city names c) Display student name and city of all students. d) count number of students in each of all students

```
n = int(input("Enter number of students: "))

students = {}
for _ in range(n):
    students[input("Name: ")] = input("City: ")

print("Names:", list(students.keys()))
print("Cities:", list(students.values()))

print("Name and City:")
for k, v in students.items():
    print(k, "-", v)

city_count = {}
for city in students.values():
    city_count[city] = city_count.get(city, 0) + 1

print("City count:", city_count)
```

```
Enter number of students: 2
City: Delhi
Name: Pranav
City: Mumbai
Name: Vivan
Names: ['Pranav', 'Vivan']
Cities: ['Delhi', 'Mumbai']
Name and City:
Pranav - Delhi
Vivan - Mumbai
City count: {'Delhi': 1, 'Mumbai': 1}
```

Q5.Store details of n movies in a dictionary by taking input from the user. Each movie must store details like name, year, director name, production cost, collection made (earning) & perform the following :- a) print all movie details b) display name of movies released before 2015 c) print movies that made a profit. d) print movies directed by a particular director.

```
n = int(input("Enter number of persons: "))


persons = {}
for _ in range(n):
    name = input("Enter name: ").strip()
    city = input("Enter city: ").strip()
    persons[name] = city

print("\nAll Names:")
for name in persons.keys():
    print(name)

print("\nAll Cities:")
for city in persons.values():
    print(city)


print("\nName and City of all persons:")
for name, city in persons.items():
    print(f"{name}: {city}")


city_count = {}
for city in persons.values():
    if city in city_count:
        city_count[city] += 1
    else:
        city_count[city] = 1

print("\nNumber of persons in each city:")
for city, count in city_count.items():
    print(f"{city}: {count}")
```

```
Enter number of persons: 2
Enter name: Jawan
Enter city: Mumbai
Enter name: Pathaan
Enter city: Kolkata
```

```
All Names:
Jawan
Pathaan

All Cities:
Mumbai
Kolkata

Name and City of all persons:
Jawan: Mumbai
Pathaan: Kolkata

Number of persons in each city:
Mumbai: 1
Kolkata: 1
```

Q6. Create a contact book where users can store, search, update, and delete contacts. Use dictionary for storing contacts.

```
contacts = {}

def add_contact():
    name = input("Enter name: ").strip()
    if name in contacts:
        print("Contact already exists!")
        return
    phone = input("Enter phone number: ").strip()
    email = input("Enter email: ").strip()
    contacts[name] = {"phone": phone, "email": email}
    print("Contact added successfully!")

def view_contacts():
    if not contacts:
        print("No contacts found.")
        return
    for name, details in contacts.items():
        print(f"\nName: {name}")
        print(f"Phone: {details['phone']}")
        print(f"Email: {details['email']}")

def search_contact():
    name = input("Enter name to search: ").strip()
    if name in contacts:
        print(f"\nName: {name}")
        print(f"Phone: {contacts[name]['phone']}")
        print(f"Email: {contacts[name]['email']}")
    else:
        print("Contact not found.")

def update_contact():
    name = input("Enter name to update: ").strip()
    if name in contacts:
        phone = input("Enter new phone number: ").strip()
        email = input("Enter new email: ").strip()
        contacts[name] = {"phone": phone, "email": email}
        print("Contact updated successfully!")
    else:
        print("Contact not found.")

def delete_contact():
    name = input("Enter name to delete: ").strip()
    if name in contacts:
        del contacts[name]
        print("Contact deleted successfully!")
    else:
        print("Contact not found.")

def menu():
    while True:
        print("\n===== Contact Book Menu =====")
        print("1. Add Contact")
        print("2. View All Contacts")
        print("3. Search Contact")
        print("4. Update Contact")
        print("5. Delete Contact")
        print("6. Exit")

        choice = input("Enter your choice (1-6): ")

        if choice == '1':
            add_contact()
```

```
        elif choice == '2':
            view_contacts()
        elif choice == '3':
            search_contact()
        elif choice == '4':
            update_contact()
        elif choice == '5':
            delete_contact()
        elif choice == '6':
            print("Exiting Contact Book. Goodbye!")
            break
        else:
            print("Invalid choice! Please try again.")


menu()
```

```
===== Contact Book Menu =====
1. Add Contact
2. View All Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
6. Exit
Enter your choice (1-6): 6
Exiting Contact Book. Goodbye!
```

Q7. Create a Todo list Manager where users can add, view, and remove tasks. Use List for storing tasks.

```
def show_menu():
    print("\n==== TODO LIST MANAGER ====")
    print("1. Add Task")
    print("2. View Tasks")
    print("3. Remove Task")
    print("4. Exit")


def add_task(tasks):
    task = input("Enter the task: ")
    tasks.append(task)
    print(f"Task '{task}' added successfully!")


def view_tasks(tasks):
    if not tasks:
        print("No tasks available.")
    else:
        print("\nYour Tasks:")
        for index, task in enumerate(tasks, start=1):
            print(f"{index}. {task}")


def remove_task(tasks):
    view_tasks(tasks)
    if tasks:
        try:
            task_number = int(input("Enter task number to remove: "))
            if 1 <= task_number <= len(tasks):
                removed = tasks.pop(task_number - 1)
                print(f"Task '{removed}' removed successfully!")
            else:
                print("Invalid task number.")
        except ValueError:
            print("Please enter a valid number.")


def main():
    tasks = []

    while True:
        show_menu()
        choice = input("Choose an option (1-4): ")

        if choice == '1':
            add_task(tasks)
        elif choice == '2':
            view_tasks(tasks)
        elif choice == '3':
            remove_task(tasks)
        elif choice == '4':
            print("Exiting Todo List Manager. Goodbye!")
```

```
            break
        else:
            print("Invalid choice. Please try again.")


if __name__ == "__main__":
    main()
```

```
==== TODO LIST MANAGER ====
1. Add Task
2. View Tasks
3. Remove Task
4. Exit
```

Git hub link = https://github.com/pranavssinha11-glitch/Python.git