```
#Q1. Scan n values in range 0-3 and print the number of times each value has occured.
num = int(input("Enter a number of values :"))
values = []
for i in range(num):
  val = int(input("Enter the value (0 to 3)"))
  values.append(val)

for i in range(4):
  print(f"{i} occured {values.count(i)} times")
```

```
Enter a number of values :6
Enter the value (0 to 3)1
Enter the value (0 to 3)0
Enter the value (0 to 3)0
Enter the value (0 to 3)2
Enter the value (0 to 3)0
Enter the value (0 to 3)1
0 occured 3 times
1 occured 2 times
2 occured 1 times
3 occured 0 times
```

```
#Q2. Create a tuple to store n numeric values and find average of all values.
num = int(input("Enter number of values:"))
n = []
for i in range(num):
  a = int(input("Enter a value:"))
  n.append(a)

avg = sum(n)/len(n)
print(avg)
```

```
Enter number of values:4
Enter a value:1
Enter a value:2
Enter a value:3
Enter a value:4
2.5
```

```
#Q3.WAP to input a list of scores for N students in a list data type. Find the score of the runnerup 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())

scores = list(map(int, input().split()))

unique_scores = sorted(list(set(scores)))

# Print the runner-up score directly
print(unique_scores[-2])
```

```
5
2 3 6 6 5
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 city

n = int(input("Enter number of persons: "))
person_data = {}

# Populate the dictionary
for i in range(n):
```

```
    name = input("Enter name: ")
    city = input("Enter city: ")
    person_data[name] = city

print("\na) All names:")
print(list(person_data.keys()))

print("\nb) All city names:")
print(list(person_data.values()))

print("\nc) Name and City:")
for name in person_data:

    print(name, "-", person_data[name])

print("\nd) Count per city:")
city_counts = {}
for city in person_data.values():
    if city in city_counts:
        city_counts[city] = city_counts[city] + 1
    else:
        city_counts[city] = 1

for city in city_counts:
    print(city, ":", city_counts[city])
```

```
Enter number of persons: 2
Enter name: Roushan
Enter city: Dehradun
Enter name: Raushan Kumar
Enter city: Rorkee

a) All names:
['Roushan', 'Raushan Kumar']

b) All city names:
['Dehradun', 'Rorkee']

c) Name and City:
Roushan - Dehradun
Raushan Kumar - Rorkee

d) Count per city:
Dehradun : 1
Rorkee : 1
```

```
#Q5.Store details of n movies in a dictionary by taking input from the user. Each movie must store details like name, year,
#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 the number of movies: "))
movie_db = {}

# Populate the dictionary
for i in range(n):
    print("\nEnter details for Movie", i + 1)
    name = input("Name: ")
    year = int(input("Year: "))
    director = input("Director: ")
    cost = float(input("Production Cost: "))
    earnings = float(input("Collection Made: "))

    # Store details in a nested dictionary
    movie_db[name] = {
        "year": year,
        "director": director,
        "cost": cost,
        "earnings": earnings
    }

# a) Print all movie details
print("\n--- a) All Movie Details ---")
for name, info in movie_db.items():
    print(name, ":", info)

# b) Display names of movies released before 2015
print("\n--- b) Movies released before 2015 ---")
for name, info in movie_db.items():
    if info["year"] < 2015:
```

```
# c) Print movies that made a profit
print("\n--- c) Movies that made a profit ---")
for name, info in movie_db.items():
    if info["earnings"] > info["cost"]:
        profit = info["earnings"] - info["cost"]
        print(name, "(Profit:", profit, ")")

# d) Print movies directed by a particular director
search_director = input("\nEnter director name to search: ")
print("Movies directed by", search_director, ":")
found = False
for name, info in movie_db.items():
    if info["director"].lower() == search_director.lower():
        print("-", name)
        found = True
if not found:
    print("No movies found for this director.")
```

```
Enter the number of movies: 1

Enter details for Movie 1
Name: DDLJ
Year: 1995
Director: Aditya Chopra
Production Cost: 1000000000
Collection Made: 2000000000

--- a) All Movie Details ---
DDLJ : {'year': 1995, 'director': 'Aditya Chopra', 'cost': 1000000000.0, 'earnings': 2000000000.0}

--- b) Movies released before 2015 ---
DDLJ

--- c) Movies that made a profit ---
DDLJ (Profit: 1000000000.0 )

Enter director name to search: Aditya Chopra
Movies directed by Aditya Chopra :
- DDLJ
```

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

while True:
    print("\n1.Add 2.Search 3.Update 4.Delete 5.Exit")
    ch = input("Choice: ")

    if ch == '1' or ch == '3': # Add and Update use the same logic
        name = input("Name: ")
        contacts[name] = input("Phone: ")

    elif ch == '2':
        name = input("Name: ")
        print(contacts.get(name, "Not found"))

    elif ch == '4':
        name = input("Name: ")
        if name in contacts:
            del contacts[name]
            print("Deleted")
        else:
            print("Not found")

    elif ch == '5':
        break
```

```
1.Add 2.Search 3.Update 4.Delete 5.Exit
Choice: 1
Name: Roushan
Phone: 123456

1.Add 2.Search 3.Update 4.Delete 5.Exit
Choice: 2
Name: Roushan
123456

1.Add 2.Search 3.Update 4.Delete 5.Exit
Choice: 5
```

```python
#Q.7 Create a Todo list Manager where users can add, view, and remove tasks. Use List for storing tasks.

tasks = []

while True:
    print("\n1.Add 2.View 3.Remove 4.Exit")
    choice = input("Choice: ")

    if choice == '1':
        task = input("Enter task: ")
        # Using append() to add task to the end of the list
        tasks.append(task)
        print("Task added.")

    elif choice == '2':
        if not tasks:
            print("No tasks found.")
        else:
            print("\nYour Tasks:")
            # Using enumerate() to show task numbers
            for i, t in enumerate(tasks, 1):
                print(i, ".", t)

    elif choice == '3':
        if not tasks:
            print("Nothing to remove.")
        else:
            try:
                num = int(input("Task number to remove: "))
                # Using pop() to remove task by its index
                removed = tasks.pop(num - 1)
                print("Removed:", removed)
            except:
                print("Invalid number.")

    elif choice == '4':
        break
```

```
1.Add 2.View 3.Remove 4.Exit
Choice: 1
Enter task: Assignment
Task added.

1.Add 2.View 3.Remove 4.Exit
Choice: 2

Your Tasks:
1 . Assignment

1.Add 2.View 3.Remove 4.Exit
Choice: 4
```

NAME = VINIT RANJAN

SAP ID = 590026420

GITHUB LINK = https://github.com/vinxtluvvv/python-