Ex.No.2	LIST, TUPLE AND SET IN PYTHON	Reg.No: URK23CS1261
06.1.24		

2 A). Write a Python program to remove duplicate elements in a list.

Aim: The objective of this program is to remove duplicate elements in a list.

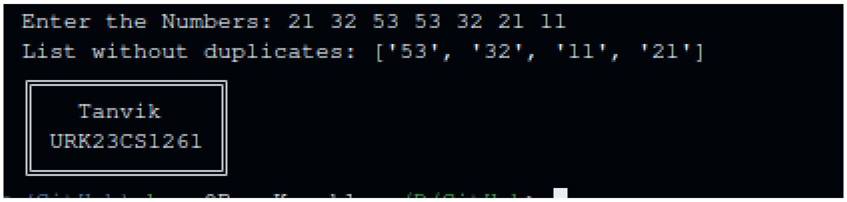
Algorithm:

Step 1: Start the program.
Step 2: Declare a variable name 'list1' to store numbers input by the user.
Step 3: The input is split using the split method to create a list of numbers.
Step 4: The 'list1' is converted into a set to remove duplicates.
Step 5: The unique elements are converted back into a list and displayed without duplicates.
Step 6: Stop the program.

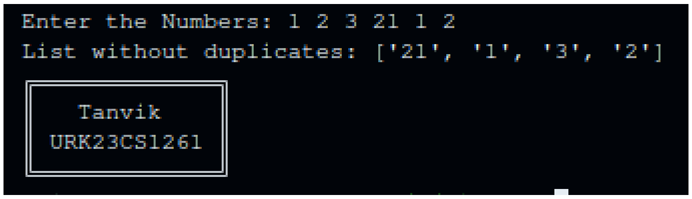
Program:

```
list1 = input("Enter the Numbers: ").split()

list1 = list(set(list1)- {''})

print(f"List without duplicates: {list1}")

print("╔═══════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═══════════════╝")
```

Output:



```
Enter the Numbers: 21 32 53 53 32 21 11
List without duplicates: ['53', '32', '11', '21']

╔═══════════╗
║  Tanvik   ║
║URK23CS1261║
╚═══════════╝
```



```
Enter the Numbers: 1 2 3 21 1 2
List without duplicates: ['21', '1', '3', '2']

╔═══════════╗
║  Tanvik   ║
║URK23CS1261║
╚═══════════╝
```

Result: Thus, The program has successfully produced the desired output.

2 B) Write a python program to print each and every element in reverse order.

Aim: The objective of this program is to print each and every element in reverse order.

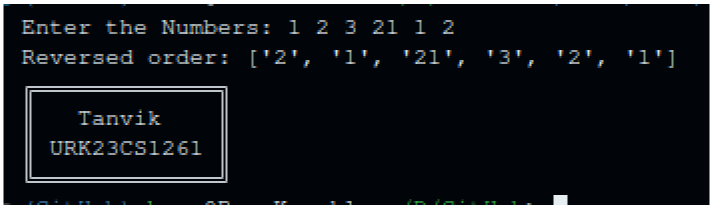
Algorithm:

```
Step 1: Start the program.
Step 2: Prompts the user to input numbers separated by spaces.
Step 3: Takes the input and splits it into a list named 'list1', using spaces as separators.
Step 4: Creates a new list 'reverse' using list slicing to reverse the order of elements in 'list1'.
Step 5: Prints the reversed list 'reverse' in a formatted string.
Step 6: Stop.
```

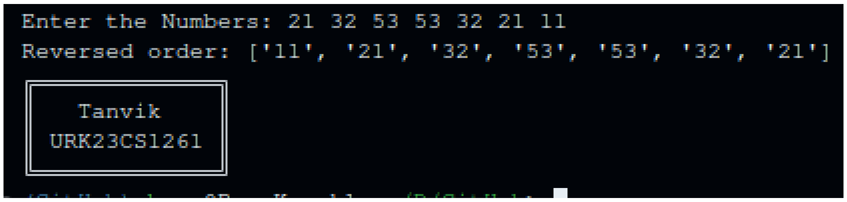
Program:

```
list1 = input("Enter the Numbers: ").split(" ")

reverse = list1[::-1]

print(f"Reversed order: {reverse}")

print("╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝")
```

Output:



```
Enter the Numbers: 1 2 3 21 1 2
Reversed order: ['2', '1', '21', '3', '2', '1']
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```



```
Enter the Numbers: 21 32 53 53 32 21 11
Reversed order: ['11', '21', '32', '53', '53', '32', '21']
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.

2 C) Develop a Python program to input and store the provided five student details such as name, regno, cgpa as tuples in a list. Provide a menu driven option to sort the student details by name , regno and cgpa and display the appropriate sorted student details.

Aim: The objective of this program is to manage student information for five students. The program will store details like name, registration number (regno), and CGPA (Cumulative Grade Point Average) as tuples within a list.

Algorithm:

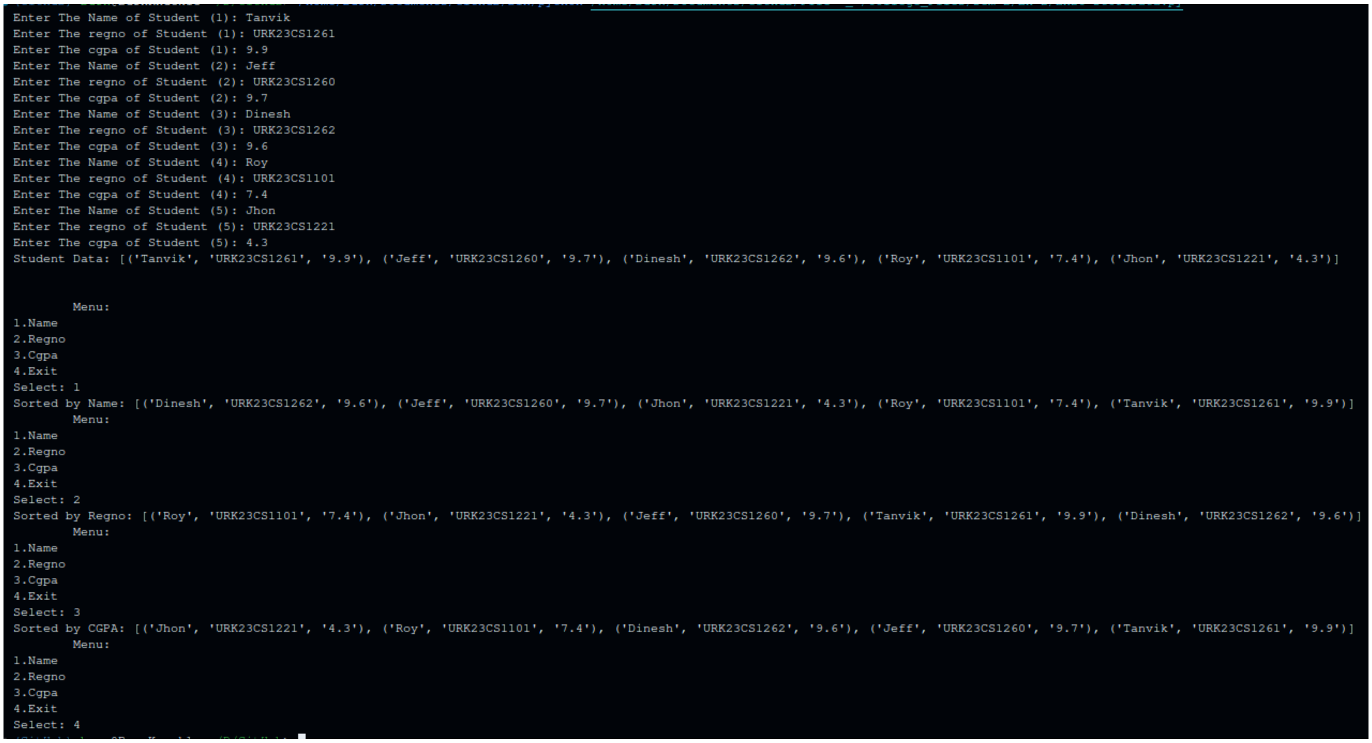
```
Step 1: Start the program.
Step 2: Create an empty list called student_data.
Step 3: For each x in the range from 1 to 5:
        a. Prompt for Name, Regno, and CGPA for a student.
        b. Create a tuple (name, regno, cgpa) with the collected data.
        c. Add the tuple to the student_data list.
Step 4: Display the student_data list.
Step 5: Start an indefinite loop using while True:
        a. Prompt the user to select a sorting option using a numerical menu.
        b. If the user picks 1, sort student_data by Name and display the sorted list.
        c. If the user picks 2, sort by Regno and display the sorted list.
        d. If the user picks 3, sort by CGPA as a float and display the sorted list.
        e. If the user picks 4, exit the loop and end the program.
        f. Handle incorrect menu inputs by catching a ValueError and asking for a valid selection.
        g. Catch any other exceptions, displaying an error message if encountered.
Step 6: Stop.
```

Program:

```
student_data = []
for x in range(1,6):
    name = input(f"Enter The Name of Student ({x}): ")
    regno = input(f"Enter The regno of Student ({x}): ")
    cgpa = input(f"Enter The cgpa of Student ({x}): ")
    tuple_data = (name, regno, cgpa)
    student_data.append(tuple_data)
print(f"Student Data: {student_data}\n\n")
try :
    while True:
        input_sort = int(input("\tMenu: \n1.Name \n2.Regno \n3.Cgpa\n4.Exit\nSelect: "))
        if input_sort == 1:
            student_data.sort(key=lambda x: x[0])
            print(f"Sorted by Name: {student_data}")
        elif input_sort == 2:
            student_data.sort(key=lambda x: x[1])
            print(f"Sorted by Regno: {student_data}")
```

```
elif input_sort == 3:
            student_data.sort(key=lambda x: float(x[2]))
            print(f"Sorted by CGPA: {student_data}")
        elif input_sort == 4:
            exit()
except ValueError:
    print("Please enter a valid menu option.")
except Exception as e:
    print(f“Error : {e}”)

print("╔══════════════════╗\n║    Tanvik    ║\n║ URK23CS1261 ║\n╚══════════════════╝")
```

Output:



```
Enter The Name of Student (1): Tanvik
Enter The regno of Student (1): URK23CS1261
Enter The cgpa of Student (1): 9.9
Enter The Name of Student (2): Jeff
Enter The regno of Student (2): URK23CS1260
Enter The cgpa of Student (2): 9.7
Enter The Name of Student (3): Dinesh
Enter The regno of Student (3): URK23CS1262
Enter The cgpa of Student (3): 9.6
Enter The Name of Student (4): Roy
Enter The regno of Student (4): URK23CS1101
Enter The cgpa of Student (4): 7.4
Enter The Name of Student (5): Jhon
Enter The regno of Student (5): URK23CS1221
Enter The cgpa of Student (5): 4.3
Student Data: [('Tanvik', 'URK23CS1261', '9.9'), ('Jeff', 'URK23CS1260', '9.7'), ('Dinesh', 'URK23CS1262', '9.6'), ('Roy', 'URK23CS1101', '7.4'), ('Jhon', 'URK23CS1221', '4.3')]


        Menu:
1.Name
2.Regno
3.Cgpa
4.Exit
Select: 1
Sorted by Name: [('Dinesh', 'URK23CS1262', '9.6'), ('Jeff', 'URK23CS1260', '9.7'), ('Jhon', 'URK23CS1221', '4.3'), ('Roy', 'URK23CS1101', '7.4'), ('Tanvik', 'URK23CS1261', '9.9')]
        Menu:
1.Name
2.Regno
3.Cgpa
4.Exit
Select: 2
Sorted by Regno: [('Roy', 'URK23CS1101', '7.4'), ('Jhon', 'URK23CS1221', '4.3'), ('Jeff', 'URK23CS1260', '9.7'), ('Tanvik', 'URK23CS1261', '9.9'), ('Dinesh', 'URK23CS1262', '9.6')]
        Menu:
1.Name
2.Regno
3.Cgpa
4.Exit
Select: 3
Sorted by CGPA: [('Jhon', 'URK23CS1221', '4.3'), ('Roy', 'URK23CS1101', '7.4'), ('Dinesh', 'URK23CS1262', '9.6'), ('Jeff', 'URK23CS1260', '9.7'), ('Tanvik', 'URK23CS1261', '9.9')]
        Menu:
1.Name
2.Regno
3.Cgpa
4.Exit
Select: 4
```

Result: Thus, The program has successfully produced the desired output.

2 D) Write a Python program to find all the unique words and count the frequency of occurrence from a given list of strings. Use Python set data type.

Aim: The objective of this program is to find all the unique words and count the frequency of occurrence from a given list of strings. Using Python set data type.

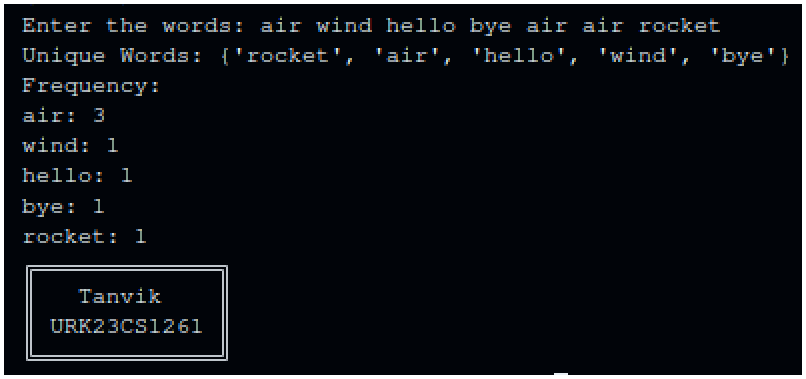
Algorithm:

Step 1: Start the program.
Step 2: Prompts the user to input words separated by spaces.
Step 3: Split the input into a list of words.
Step 4: Create a set to store unique words and a dictionary to track word frequencies.
Step 5: Calculate the frequency of each word and store it in the dictionary.
Step 6: Display unique words, their frequencies.
Step 7: Stop.

Program:

```
words = input("Enter the words: ")
words = words.split(" ")
unq_words = set(words)
word_freq = {}
for word in words:
   if word in word_freq:
      word_freq[word] += 1
   else:
      word_freq[word] = 1
print(f"Unique Words: {unq_words}")
print("Frequency:")
for word, freq in word_freq.items():
   print(f"{word}: {freq}")

print("╔═════════════╗\n║   Tanvik    ║\n║ URK23CS1261 ║\n╚═════════════╝")
```

Output:



```
Enter the words: air wind hello bye air air rocket
Unique Words: {'rocket', 'air', 'hello', 'wind', 'bye'}
Frequency:
air: 3
wind: 1
hello: 1
bye: 1
rocket: 1
╔═════════════╗
║   Tanvik    ║
║ URK23CS1261 ║
╚═════════════╝
```

Result: Thus, The program has successfully produced the desired output.