|  |  |
| --- | --- |
| **Ex. No. 10**  **Date:23.06.2021** | **LIST AND TUPLES – LEVEL 2** |

**AIM:**

To write python program using List and Tuple.

**PROGRAMS:**

**a) Display the duplicate elements**

**Description:**

Given a list of integers with duplicate elements in it.

The objective is to generate another list, which contains only the duplicate elements in sorted order.

Input Format:

Get the input as a string separating each number by space.

Use spilt() to split the numbers and convert it to int.

Output Format:

Resultant list with the duplicate elements in sorted order

Sample Input

1 3 1 2 3

Sample Output

[1, 3]

Sample Input

1 2 3 4 5

Sample Output

There are no duplicates

**Program Template:**

# Function Definition

deffindDuplicate(a):

# Split the input

# Identify the duplicates after converting it to integer

# Sort the duplicate list

# Display the list

# Get the input from the user

a = input()

# Function call

findDuplicate(a)

**Program:**

'''Name:R.sridevi

Roll No:20uit021

Program :Display the duplicate elements'''

# Function Definition

def findDuplicate(a):

d\_list=[]

# Split the input

list1 = a.split(' ')

# Identify the duplicates after converting it to integer

for i in range(len(list1)):

if(list1[i]!=''):

list1[i]=int(list1[i])

else:

pass

for i in range(len(list1)):

for j in range(i+1,len(list1)):

if(list1[i]==list1[j]):

d\_list.append(list1[i])

# Sort the duplicate list

d\_list=list(set(d\_list))

d\_list.sort()

#Display the list

if(len(d\_list)>0):

print(d\_list)

else:

print("There are no duplicates")

# Get the input from the user

a = input()

# Function call

findDuplicate(a)

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 1 3 1 2 3 | [1, 3] |
| 2 | 1 2 3 4 5 | There are no duplicates |
| 3 | 0 0 0 | [0] |
| 4 | 9 -3 -9 -3 8 -3 2 1 1 1 2 2 | [-3, 1, 2] |
| 5 | -1 -3 -8 -3 -8 -8 -9 | [-8, -3] |
| **Total Test Cases** | | **5** |
| **Number of Test Cases Passed** | | **5** |

**b) Right Rotate the Elements in the List**

**Description:**

Given a list, right rotate the list by n times.

Constraint:

1. Number of times (n) cannot be negative

2. Number of times (n) should be equal to or lesser than the total elements.

3. Number of times (n) can be 0

Input Format:

First Line - Get the input as a string separating each number by space.

Use spilt() to split the numbers and convert it to int.

Second Line - n value

Output Format:

Resultant list with the right rotated elements

Sample Input

1 2 3 4 5

2

Sample Output

[4, 5, 1, 2, 3]

Sample Input

1 2 3 4

-9

Sample Output

Sorry! position cannot be a negative value

Sample Input

1

4

Sample Output

Sorry! the total value is less than n value

**Program Template:**

# Function Definition

def rotate(a, n):

# Split the input

# Convert it to integer and then rotate

# Display the list

# Get the input from the user

a = input()

# Get the number of times to rotate

n =

# Function call

rotate(a, n)

**Program:**

'''Name: R.sridevi

Roll No:20uit021

Program:Right Rotate the Elements in the List'''

# Function Definition

def rotate(a, n):

if(n>=0):

# Split the input

a = a.split(' ')

if(len(a)>=n):

# Convert it to integer and then rotate

for i in range(len(a)):

a[i] = int(a[i])

for i in range(n):

temp=a[len(a)-1]

for j in range(len(a)-1,0,-1):

a[j]=a[j-1]

a[0]=temp

# Display the list

print(a)

else:

print("Sorry! the total value is less than n value")

else:

print("Sorry! position cannot be a negative value")

# Get the input from the user

a = input()

# Get the number of times to rotate

n = int(input())

# Function call

rotate(a, n)

**Test Cases:**

|  |  |  |
| --- | --- | --- |
| **Test Case No.** | **Input** | **Expected Output** |
| 1 | 1 2 3 4 5  2 | [4, 5, 1, 2, 3] |
| 2 | 1 2 3 4  -9 | Sorry! position cannot be a negative value |
| 3 | 1  4 | Sorry! the total value is less than n value |
| 4 | 1 2 3 4  4 | [1, 2, 3, 4] |
| 5 | 1 2 3 4  0 | [1, 2, 3, 4] |
| 6 | 1 2 3 4 5  4 | [2, 3, 4, 5, 1] |
| **Total Test Cases** | | **6** |
| **Number of Test Cases Passed** | | **6** |

**RESULT:**

Thus, the Python programs are executed successfully.