1. Python – Join Tuples if similar initial element

test\_list = [(5, 6), (5, 7), (6, 8), (6, 10), (7, 13)]

print("The original list is : " + str(test\_list))

res = []

for sub in test\_list:

if res and res[-1][0] == sub[0]:

res[-1].extend(sub[1:])

else:

res.append([ele for ele in sub])

res = list(map(tuple, res))

print("The extracted elements : " + str(res))

1. Python – Extract digits from Tuple list

from itertools import chain

test\_list = [(15, 3), (3, 9), (1, 10), (99, 2)]

print("The original list is : " + str(test\_list))

temp = map(lambda ele: str(ele), chain.from\_iterable(test\_list))

res = set()

for sub in temp:

for ele in sub:

res.add(ele)

print("The extracted digits : " + str(res))

1. Python – All pair combinations of 2 tuples

test\_tuple1 = (4, 5)

test\_tuple2 = (7, 8)

print("The original tuple 1 : " + str(test\_tuple1))

print("The original tuple 2 : " + str(test\_tuple2))

res = [(a, b) for a in test\_tuple1 for b in test\_tuple2]

res = res + [(a, b) for a in test\_tuple2 for b in test\_tuple1]

print("The filtered tuple : " + str(res))

1. Python – Remove Tuples of Length K

test\_list = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]

print("The original list : " + str(test\_list))

K = 1

res = [ele for ele in test\_list if len(ele) != K]

print("Filtered list : " + str(res))

1. Sort a list of tuples by second Item

def Sort\_Tuple(tup):

lst = len(tup)

for i in range(0, lst):

for j in range(0, lst-i-1):

if (tup[j][1] > tup[j + 1][1]):

temp = tup[j]

tup[j]= tup[j + 1]

tup[j + 1]= temp

return tup

tup =[('for', 24), ('is', 10), ('Geeks', 28),

('Geeksforgeeks', 5), ('portal', 20), ('a', 15)]

print(Sort\_Tuple(tup))

1. Python program to Order Tuples using external List

my\_list = [('Mark', 34), ('Will', 91), ('Rob', 23)]

print("The list of tuple is : ")

print(my\_list)

ordered\_list = ['Will', 'Mark', 'Rob']

print("The ordered list is :")

print(ordered\_list)

temp = dict(my\_list)

my\_result = [(key, temp[key]) for key in ordered\_list]

print("The ordered tuple list is : ")

print(my\_result)

1. Python – Flatten tuple of List to tuple

test\_tuple = ([5, 6], [6, 7, 8, 9], [3])

print("The original tuple : " + str(test\_tuple))

res = tuple(sum(test\_tuple, []))

print("The flattened tuple : " + str(res))

1. Python – Convert Nested Tuple to Custom Key Dictionary

test\_tuple = ((4, 'Gfg', 10), (3, 'is', 8), (6, 'Best', 10))

print("The original tuple : " + str(test\_tuple))

res = [{'key': sub[0], 'value': sub[1], 'id': sub[2]}

for sub in test\_tuple]

print("The converted dictionary : " + str(res))

1. Python Program for Binary Search (Recursive and Iterative

def binary\_search(arr, low, high, x):

if high >= low:

mid = (high + low) // 2

if arr[mid] == x:

return mid

elif arr[mid] > x:

return binary\_search(arr, low, mid - 1, x)

else:

return binary\_search(arr, mid + 1, high, x)

else:

# Element is not present in the array

return -1

arr = [ 2, 3, 4, 10, 40 ]

x = 10

result = binary\_search(arr, 0, len(arr)-1, x)

if result != -1:

print("Element is present at index", str(result))

else:

print("Element is not present in array")

1. Python Program for Linear Search

def search(arr, x):

for i in range(len(arr)):

if arr[i] == x:

return i

return -1

1. Python Program for Insertion Sort

def insertionSort(arr):

for i in range(1, len(arr)):

key = arr[i]

j = i-1

while j >=0 and key < arr[j] :

arr[j+1] = arr[j]

j -= 1

arr[j+1] = key

arr = [12, 11, 13, 5, 6]

insertionSort(arr)

lst = []

print("Sorted array is : ")

for i in range(len(arr)):

lst.append(arr[i])

print(lst)