1: map()

my\_pets = ['alfred', 'tabitha', 'william', 'arla']

uppered\_pets = list(map(str.upper, my\_pets))

print(uppered\_pets)

o/p : ['ALFRED', 'TABITHA', 'WILLIAM', 'ARLA']

2: map() – two iterable passed

circle\_areas = [3.56773, 5.57668, 4.00914, 56.24241, 9.01344, 32.00013]

result = list(map(round, circle\_areas, range(1,7)))

print(result)

o/p:

[3.6, 5.58, 4.009, 56.2424, 9.01344, 32.00013]

3:map() two iterable of different length – It will executer one of the iterables has elements

circle\_areas = [3.56773, 5.57668, 4.00914, 56.24241, 9.01344, 32.00013]

result = list(map(round, circle\_areas, range(1,3)))

print(result)

o/p;

[3.6, 5.58]

4)map() ,To implement zip() using map()

my\_strings = ['a', 'b', 'c', 'd', 'e']

my\_numbers = [1,2,3,4,5]

results = list(map(lambda x , y : (x ,y) , my\_strings, my\_numbers))

print (results)

o/p:

[('a', 1), ('b', 2), ('c', 3), ('d', 4), ('e', 5)]

5)filter()

dromes = ("demigod", "rewire", "madam", "freer", "anutforajaroftuna", "kiosk")

palindromes = list(filter(lambda word: word == word[::-1], dromes))

print(palindromes)

o/p:

['madam', 'anutforajaroftuna']

6) reduce()

from functools import reduce

numbers = [3, 4, 6, 9, 34, 12]

def custom\_sum(first, second):

return first + second

result = reduce(custom\_sum, numbers, 10)

print(result)

Output:

78