FILTER, MAP, REDUCE METHODS

1. filter()
    a. this method is used to filter out the items from the sequence of items/iterable object based on the condition
    b. syntax:

```
filter(function,iterable)
```

⇨ function – a function that returns True or False

⇨ iterable – like list, tuple or any sequence

⇨ it returns the item, whenever the condition is True

ex by using normal function:

```
def is_even(li):
    return li%2==0
li=[1,2,3,4,5,6,7,8,9]
evenno=list(filter(is_even,li))
print(evenno)
```

ex by using lambda function:

```
li=[1,2,3,4,5,6,7,8,9]
evenno=list(filter(lambda num:num%2==0,li))
print(evenno)
```

2. map()

a. this method is used to apply a function i.e., operation to the each item in the sequence and return the results

b. syntax:

```
map(function,iterable)
```

⇨ function – a function that performs operation on each item in the sequence/iterable

⇨ iterable – like list, tuple or any sequence

ex by using lambda function:

```
evenno=[2,4,6,8]
square=list(map(lambda num:num*num,evenno))
print(square)
```

3. reduce()

a. this method is used to repeatedly apply a function to elements of a sequence, reducing the sequence in to a single final value

b. this method is available under the “functools” module

c. syntax:

```
from functools import reduce
reduce(function,iterable)
```

⇨ function – a function that takes two inputs returns one output

⇨ iterable – like list, tuple or any sequence

in simple words, it combines the first two elements, then that result will be added to the third element and so on….

Ex:

```
from functools import reduce
nums=[1,2,3,4,5]
add_result=reduce(lambda num1,num2:num1+num2,nums)
print(add_result)
```