1. multi = lambda x, y : x * y
   print(multi(5, 20))

   print('\nResult from a multi Function')
   def multi_func(x, y):
   return x * y

   print(multi_func(5, 20))

O/P:

100

Result from a multi Function

100

2. from functools import reduce

fib_series = lambda n: reduce(lambda x, _: x+[x[-1]+x[-2]],
                range(n-2), [0, 1])

print('Fibonacci series upto 2:')
print(fib_series(2))
print('\nFibonacci series upto 5:')
print(fib_series(5))
print('\nFibonacci series upto 6:')
print(fib_series(6))
print('\nFibonacci series upto 9:')
print(fib_series(9))

O/P:

Fibonacci series upto 2:

[0, 1]

Fibonacci series upto 5:

[0, 1, 1, 2, 3]

Fibonacci series upto 6:

[0, 1, 1, 2, 3, 5]

Fibonacci series upto 9:

[0, 1, 1, 2, 3, 5, 8, 13, 21]

3. nums = [2, 4, 6, 9 , 11]

n = 2

print('Original list: ', nums)

print('Given number: ', n)

filtered_numbers=list(map(lambda number:number*n,nums))

print('Result:')

print(' '.join(map(str,filtered_numbers)))

O/P:

Original list: [2, 4, 6, 9, 11]

Given number: 2

Result:

4 8 12 18 22

4. # Take a list of numbers

my_list = [12, 65, 54, 39, 102, 339, 221,]

use anonymous function to filter

result = list(filter(lambda x: (x % 9 == 0), my_list))

display the result

print('Numbers divisible by 9 are',result)

O/P:

Numbers divisible by 9 are [54]

5. list1 = [21,3,4,6,33,2,3,1,3,76]

#even numbers

```
even_count = len(list(filter(lambda x: (x%2 == 0) , list1)))
print('Even numbers available in the list: ', even_count)
```

O/p:

```
Even numbers available in the list: 4
```