**1.** Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and prints the result.  
Sample Output:  
25  
48

**Answer)**

**r = lambda a: a + 15**

**print(r(10))**

**r = lambda x, y: x \* y**

**print(r(12, 4))**

**2.** Write a Python program to create a function that takes one argument, and that argument will be multiplied with an unknown given number.  
Sample Output:  
Double the number of 15 = 30  
Triple the number of 15 = 45  
Quadruple the number of 15 = 60  
Quintuple the number 15 = 75

**Answer)** Here in below code I have tried with number 7

**def fn\_calc(n):**

**return lambda x: x \* n**

**result = fn\_calc (2)**

**print("Double the number of 7 =", result(7))**

**result = fn\_calc (3)**

**print("Triple the number of 7 =", result(7))**

**result = fn\_calc (4)**

**print("Quadruple the number of 7 =", result(7))**

**result = fn\_calc (5)**

**print("Quintuple the number 7 =", result(7))**

**Output will be:**

Double the number of 7 = 14

Triple the number of 7 = 21

Quadruple the number of 7 = 28

Quintuple the number 7 = 35

**3.** Write a Python program to sort a list of tuples using Lambda.  
Original list of tuples:  
[('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]  
Sorting the List of Tuples:  
[('Social sciences', 82), ('English', 88), ('Science', 90), ('Maths', 97)]

**Answer)**

**marks = [('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]**

**print("Original list of tuples:")**

**print(marks)**

**marks.sort(key=lambda x: x[1])**

**print("\nSorting the List of Tuples:")**

**print(marks)**

**Output will be:**

**Original list of tuples:**

[('English', 88), ('Science', 90), ('Maths', 97), ('Social sciences', 82)]

**Sorting the List of Tuples:**

[('Social sciences', 82), ('English', 88), ('Science', 90), ('Maths', 97)]

**4.** Write a Python program to sort a list of dictionaries using Lambda.  
Original list of dictionaries :  
[{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Mi Max', 'model': '2', 'color': 'Gold'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}]  
Sorting the List of dictionaries :  
[{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}, {'make': 'Mi Max', 'model': '2', 'color': 'Gold'}]

**Answer)**

**models = [**

**{'make': 'Nokia', 'model': 216, 'color': 'Black'},**

**{'make': 'Mi Max', 'model': '2', 'color': 'Gold'},**

**{'make': 'Samsung', 'model': 7, 'color': 'Blue'}**

**]**

**print("Original list of dictionaries:")**

**print(models)**

**sorted\_models = sorted(models, key=lambda x: x[make])**

**print("\nSorting the List of dictionaries:")**

**print(sorted\_models)**

**Output will be:**

**Original list of dictionaries:**

[{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Mi Max', 'model': '2', 'color': 'Gold'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}]

**Sorting the List of dictionaries:**

[{'make': 'Mi Max', 'model': '2', 'color': 'Gold'}, {'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}]

**5.** Write a Python program to filter a list of integers using Lambda.  
Original list of integers:  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
Even numbers from the said list:  
[2, 4, 6, 8, 10]  
Odd numbers from the said list:  
[1, 3, 5, 7, 9]

**Answer)**

**nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

**print("Original list of integers:")**

**print(nums)**

**print("\nEven numbers from the list:")**

**even\_nums = list(filter(lambda x: x % 2 == 0, nums))**

**print(even\_nums)**

**print("\nOdd numbers from the list:")**

**odd\_nums = list(filter(lambda x: x % 2 != 0, nums))**

**print(odd\_nums)**

**6.** Write a Python program to square and cube every number in a given list of integers using Lambda.  
Original list of integers:  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
Square every number of the said list:  
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]  
Cube every number of the said list:  
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]  
**Answer)**

**nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

**print("Original list of integers:")**

**print(nums)**

**print("\nSquare every number of the list:")**

**square\_nums = list(map(lambda x: x \*\* 2, nums))**

**print(square\_nums)**

**print("\nCube every number of the list:")**

**cube\_nums = list(map(lambda x: x \*\* 3, nums))**

**print(cube\_nums)**

**7.** Write a Python program to find if a given string starts with a given character using Lambda.  
Sample Output:  
True  
False

**Answer)**

**starts\_with = lambda x: True if x.startswith('H') else False**

**print(starts\_with('Hashtag'))**

**print(starts\_with('SemiColon'))**

**8.** Write a Python program to extract year, month, date and time using Lambda.

**Answer)**

**import datetime**

**now = datetime.datetime.now()**

**print(now)**

**year = lambda x: x.year**

**month = lambda x: x.month**

**day = lambda x: x.day**

**t = lambda x: x.time()**

**print(year(now))**

**print(month(now))**

**print(day(now))**

**print(t(now))**

**Output will be:**

2024-09-17 17:54:27.017000

2024

9

17

17:54:27.017000

**9.** Write a Python program to check whether a given string is a number or not using Lambda.  
Sample Output:  
True  
True  
False  
True  
False  
True  
Print checking numbers:  
True  
True

**Answer)**

**is\_num = lambda q: q.replace('.', '', 1).isdigit()**

**print(is\_num('12345'))**

**print(is\_num('50'))**

**print(is\_num('-1'))**

**print(is\_num('8'))**

**print(is\_num('YA01'))**

**print(is\_num('19'))**

**print("\nPrint checking numbers:")**

**is\_num1 = lambda r: is\_num(r[1:]) if r[0] == '-' else is\_num(r)**

**print(is\_num1('-34'))**

**print(is\_num1('-8.98'))**

**10.** Write a Python program to create Fibonacci series up to n using Lambda.  
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]

**Answer)**

**from functools import reduce**

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

**print("Fibonacci series upto 2:")**

**print(val(2))**

**print("\nFibonacci series upto 5:")**

**print(val(5))**

**print("\nFibonacci series upto 6:")**

**print(val(6))**

**print("\nFibonacci series upto 9:")**

**print(val(9))**

**11.** Write a Python program to find the intersection of two given arrays using Lambda.  
Original arrays:  
[1, 2, 3, 5, 7, 8, 9, 10]  
[1, 2, 4, 8, 9]  
Intersection of the said arrays: [1, 2, 8, 9]

**Answer)**

**array\_nums1 = [1, 2, 3, 5, 7, 8, 9, 10]**

**array\_nums2 = [1, 2, 4, 8, 9]**

**print("Original arrays:")**

**print(array\_nums1)**

**print(array\_nums2)**

**result = list(filter(lambda x: x in array\_nums1, array\_nums2))**

**print("\nArrays intersected: ", result)**