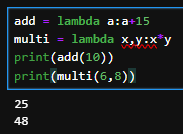
**Assignment 4**

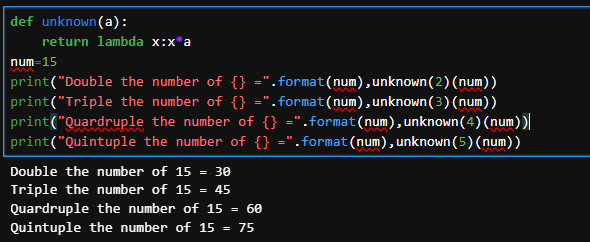
1. Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and prints the result.  
   Sample Output:  
   25  
   48

**add = lambda a:a+15**

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

**print(add(10))**

**print(multi(6,8))**

1. 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

**def unknown(a):**

**return lambda x:x\*a**

**num=15**

**print("Double the number of {} =".format(num),unknown(2)(num))**

**print("Triple the number of {} =".format(num),unknown(3)(num))**

**print("Quardruple the number of {} =".format(num),unknown(4)(num))**

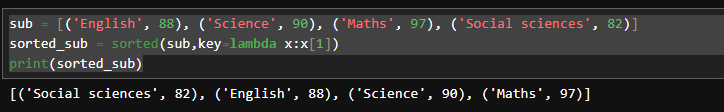
**print("Quintuple the number of {} =".format(num),unknown(5)(num))**

1. 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)]

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

**sorted\_sub = sorted(sub,key=lambda x:x[1])**

**print(sorted\_sub)**

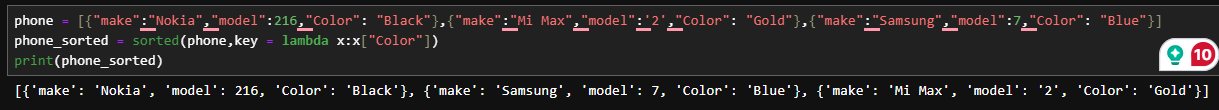


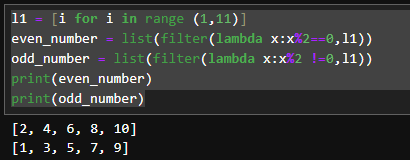
1. Write a Python program to sort a list of dictionaries using Lambda.  
   Original list of dictionaries :  
   [{'make': 'Nokia', 'model': 216, 'color': 'Black'}, {'make': 'Mi Max', 'model': '2', 'color': 'Gold'}, {'make': 'Samsung', 'model': 7, 'color': 'Blue'}]  
   Sorting the List of dictionaries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

**phone = [{"make":"Nokia","model":216,"Color": "Black"},{"make":"Mi Max","model":'2',"Color": "Gold"},{"make":"Samsung","model":7,"Color": "Blue"}]**

**phone\_sorted = sorted(phone,key = lambda x:x["Color"])**

**print(phone\_sorted)**



1. 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]

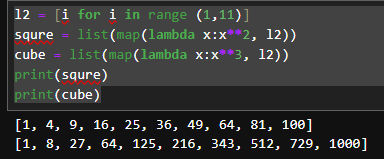
**l1 = [i for i in range (1,11)]**

**even\_number = list(filter(lambda x:x%2==0,l1))**

**odd\_number = list(filter(lambda x:x%2 !=0,l1))**

**print(even\_number)**

**print(odd\_number)**

1. 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]

**l2 = [i for i in range (1,11)]**

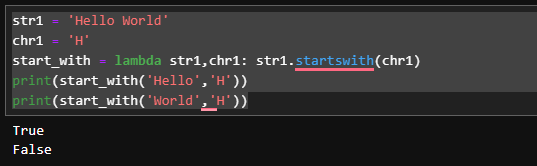
**squre = list(map(lambda x:x\*\*2, l2))**

**cube = list(map(lambda x:x\*\*3, l2))**

**print(squre)**

**print(cube)**

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



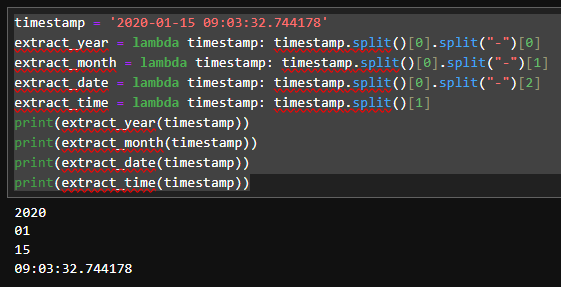
**str1 = 'Hello World'**

**chr1 = 'H'**

**start\_with = lambda str1,chr1: str1.startswith(chr1)**

**print(start\_with('Hello','H'))**

**print(start\_with('World','H'))**

1. Write a Python program to extract year, month, date and time using Lambda.  
   Sample Output:  
   2020-01-15 09:03:32.744178  
   2020  
   1  
   15  
   09:03:32.744178

**timestamp = '2020-01-15 09:03:32.744178'**

**extract\_year = lambda timestamp: timestamp.split()[0].split("-")[0]**

**extract\_month = lambda timestamp: timestamp.split()[0].split("-")[1]**

**extract\_date = lambda timestamp: timestamp.split()[0].split("-")[2]**

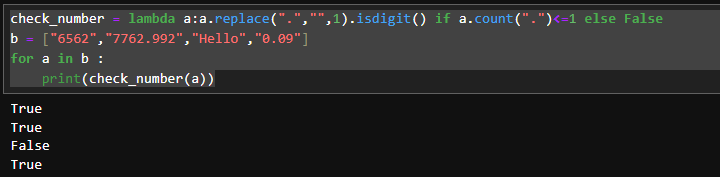
**extract\_time = lambda timestamp: timestamp.split()[1]**

**print(extract\_year(timestamp))**

**print(extract\_month(timestamp))**

**print(extract\_date(timestamp))**

**print(extract\_time(timestamp))**

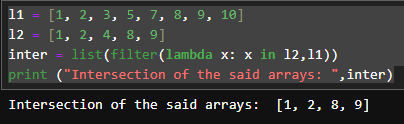
1. Write a Python program to check whether a given string is a number or not using Lambda.  
   Sample Output:  
   True  
   True  
   False  
   True  
   False  
   True  
   Print checking numbers:  
   True  
   True

**check\_number = lambda a:a.replace(".","",1).isdigit() if a.count(".")<=1 else False**

**b = ["6562","7762.992","Hello","0.09"]**

**for a in b :**

**print(check\_number(a))**

1. 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]

**l1 = [1, 2, 3, 5, 7, 8, 9, 10]**

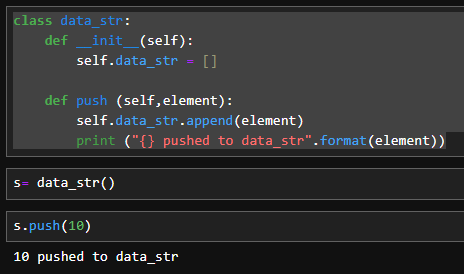
**l2 = [1, 2, 4, 8, 9]**

**inter = list(filter(lambda x: x in l2,l1))**

**print ("Intersection of the said arrays: ",inter)**

**Assignment 3**

1. Write a Python program to create a class representing a stack data structure. Include methods for pushing and popping elements.

**class data\_str:**

**def \_\_init\_\_(self):**

**self.data\_str = []**

**def push (self,element):**

**self.data\_str.append(element)**

**print ("{} pushed to data\_str".format(element))**