Assignment functions

1) What is the difference between a function and a method in Python?

(i)Function : A function is a structured , reusable block of code utilized for carrying out a specific task . exp -> print(),etc.

(ii) Method : Method is a function that is associated with an object and is called on that object. exp -> 'Ravi'.upper() ,etc.

2) Explain the concept of function arguments and parameters in Python.

(i) Parameters : Variables defined in the function definition to accept inputs. exp -> def add(a,b).

(ii) Arguments : Values passed to the function when called . exp -> add(5+5).

3) What are the different ways to define and call a function in Python?

1. Defining and Calling a Basic Function.The simplest way to define a function is by using the def keyword, followed by the function name, parameters (if any), and a block of code that defines the function's behavior.

2. Defining a Function with Default Arguments .You can provide default values for parameters in case no value is passed when calling the function

3. Defining a Function with Variable Number of Arguments (*args and **kwargs)

(i) *args allows the function to accept a variable number of positional arguments.

(ii) **kwargs allows the function to accept a variable number of keyword arguments.

4. Lambda Functions (Anonymous Functions) .Lambda functions are small anonymous functions that can have any number of arguments but only one expression. They are often used for short, throwaway functions.

4) What is the purpose of the `return` statement in a Python function?

The return statement sends a value from a function back to caller . Without return, the function returns 'None' by default.

5) What are iterators in Python and how do they differ from iterables?

(i) Iterable : An object that can be looped over or iterated . exp -> lists,strings ,etc.

(ii) Iterator: An object produced by calling 'iter()' on an iterable. It fetches one element at a time using 'next()'.

6) Explain the concept of generators in Python and how they are defined.

A generator in Python is a special type of iterator that allows you to iterate over a sequence of values, but it does so lazily, meaning it generates values one at a time as needed. This is different from a standard iterator, which computes and stores all values at once, potentially consuming a lot of memory. Generators, on the other hand, only produce values when requested, making them memory-efficient, especially for large data sets.

A generator function is defined just like a regular function, but instead of return, it uses the yield keyword to return a value. This allows the function to generate values one at a time when iterated over.

7) What are the advantages of using generators over regular functions?

(i) Memory efficient as they generate values lazily.

(ii) Simplifies code for large datasets or infinite sequences.

8) What is a lambda function in Python and when is it typically used?

(i) Anonymous, single-line functions defined with lambda keyword. Example: lambda x: x + 1.

(ii) Typically used for short, simple functions.

9) Explain the purpose and usage of the `map()` function in Python.

The map function applies a given function to all items in an input iterable (like a list) and returns an iterator with the results.

Usage : Used to transform each item in an iterable by applying the specified function.

Syntax: map(function,iterable)

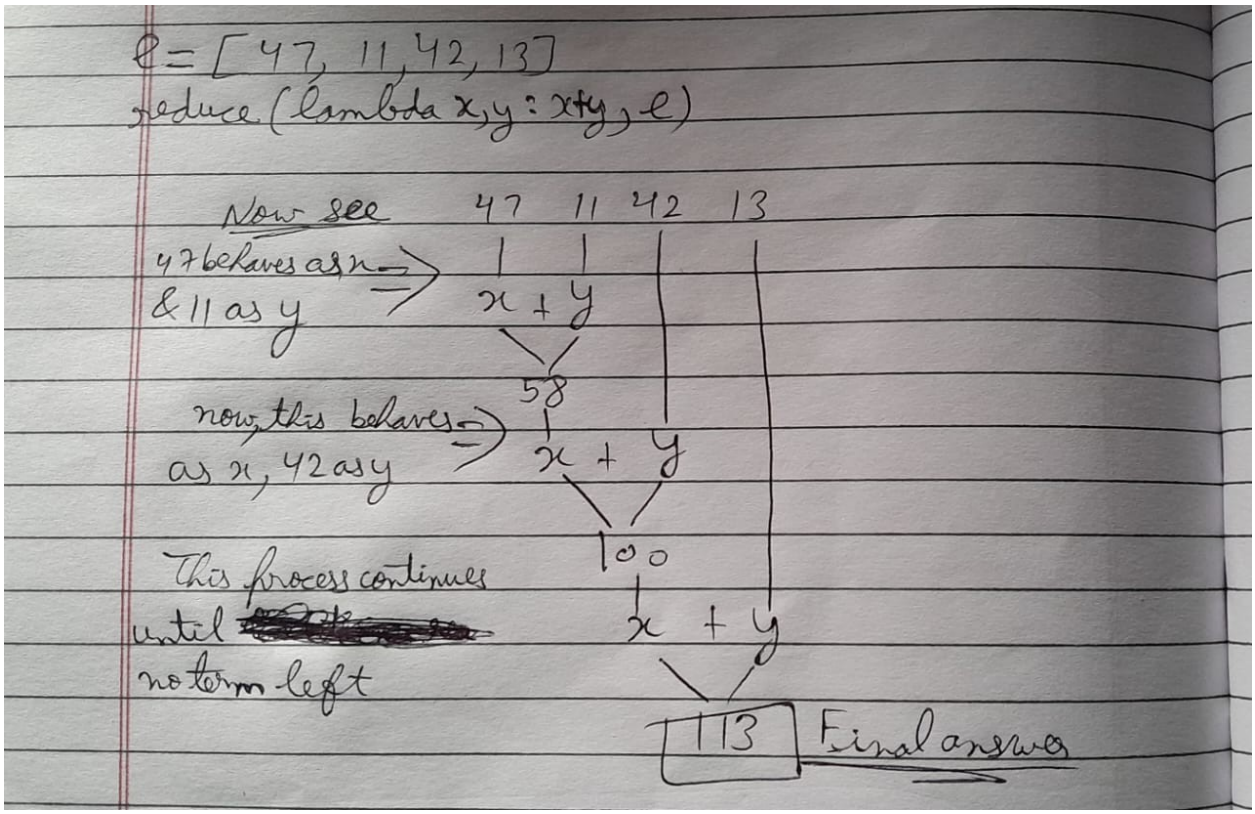
10) What is the difference between `map()`, `reduce()`, and `filter()` functions in Python?

(i) Map : The 'map()' function apllies a given function to all the iterables(like lsit) and returns an iterator with the result. Syntax : mao(function,iterable).

(ii) Reduce : The 'reduce()' function from the functools module applies a given function cumulatively to the items of a sequence, from left to right , to reduce the sequence to a single value. Synatx : reduce(function,iterable).

(iii) Filter : The 'filter()' function constructs a list or iterator from elements of an iterable for which a specified function returns True.

11) Using pen & Paper write the internal mechanism for sum operation using reduce function on this given list: [47,11,42,13];



Practical question

1. Write a Python function that takes a list of numbers as input and returns the sum of all even numbers in the list.

```
from functools import reduce
lst = [int(input('Enter element'))  for i in range(5) ]
even = list(filter(lambda x :  x%2==0 ,lst))
sum = reduce(lambda x,y:x+y,even)
print('Sum of all even numbers in list is  : ',sum)
```

```
Enter element 1
Enter element 2
Enter element 3
Enter element 4
Enter element 6

Sum of all even numbers in list is  :  12
```

2. Create a Python function that accepts a string and returns the reverse of that string.

```
str = [input('Enter string') for i in range(1)]
list(map(lambda x : x[::-1],str))
```

```
Enter string vaday ramuk ivar
```

```
['ravi kumar yadav']
```

3. Implement a Python function that takes a list of integers and returns a new list containing the squares of each number.

```
list1= [int(input('Enter integers : ')) for i in range(5)]
square = list(map(lambda x : x**2,list1))
print('Square of all elements of list is : ',square)
```

```
Enter integers :  1
Enter integers :  2
Enter integers :  3
Enter integers :  4
Enter integers :  5
```

```
Square of all elements of list is :  [1, 4, 9, 16, 25]
```

4. Write a Python function that checks if a given number is prime or not from 1 to 200.

```
list2 = [ i for i in range(1,201)]
checks = lambda x : all( x%i !=0  for i in range(2,int(x**0.5)+1))
prime = list(filter(checks,list2))
print('Prime numbers from 1 to 200 are : ',prime)
```

```
Prime numbers from 1 to 200 are :  [1, 2, 3, 5, 7, 11, 13, 17, 19, 23,
29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101,
103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173,
179, 181, 191, 193, 197, 199]
```

5. Create an iterator class in Python that generates the Fibonacci sequence up to a specified number of terms.

```
def fibonacci_generator(n):
    current, next = 0, 1
    for _ in range(n):
        yield current
```

```
        current, next = next, current + next

num_terms = 10
for num in fibonacci_generator(num_terms):
    print(num, end="    ")
```

```
0    1    1    2    3    5    8    13    21    34
```

6. Write a generator function in Python that yields the powers of 2 up to a given exponent.

```
def expo(n):
    for i in range(n+1):
        yield 2**i
for i in expo(4):
    print(i,end="   ")
```

```
1   2   4   8   16
```

7. Implement a generator function that reads a file line by line and yields each line as a string.

```
def read_file(file_path):
    with open(file_path, 'r') as file:
        for line in file:
            yield line.strip()
for line in read_file(r"C:\\Users\\kumar\\Documents\\It's me.txt"):
    print(line)
```

```
Hi Everyone my name is 'Ravi kumar yadav' and i belongs to 'Barnala , Punjab' . I am 18 years old and i have done my schooling from Dr: R.P.S.D Sen Sec School , Barnala with very good score. Currently , i am pursuing my B.tech degree in Computer Science with specialisation of A.I. & M.L. from a private college in ,Punjab. I am a very quick learner . I adapts and learns the things very quickly.I like to play Cricket , Football and do Yoga - Exercises.
Thank you
```

8. Use a lambda function in Python to sort a list of tuples based on the second element of each tuple.

```
listt = [(1,3),(4,1),(2,2),(5,1),(100,1/2),(56,5),(0,0)]
sorted_list = sorted(listt,key = lambda x :x[1])
print
```

```
[(0, 0), (100, 0.5), (4, 1), (5, 1), (2, 2), (1, 3), (56, 5)]
```

9. Write a Python program that uses `map()` to convert a list of temperatures from Celsius to Fahrenheit.

```
Celsius = [-40,37.1,40,30,19,7]
Fahrenheit = list(map(lambda x: (9/5)*x+32,Celsius))
print('List of temperatures conversion to Fahrenheit : ',Fahrenheit)
```

```
List of temperatures conversion to Fahrenheit :  [-40.0, 98.78, 104.0, 86.0, 66.2, 44.6]
```

10. Create a Python program that uses `filter()` to remove all the vowels from a given string.

```
vowels = 'aeiouAEIOU'
strig = input('Enter string : ')
s = list(filter(lambda x: x  in vowels,strig))
print(f'Vowels in {strig} is/are : {s}')
```

```
Enter string :  Ravi kumar yadav

Vowels in Ravi kumar yadav is/are : ['a', 'i', 'u', 'a', 'a', 'a']
```

11) Imagine an accounting routine used in a book shop. It works on a list with sublists, which look like this:

Order Number	Book Title and Author	Quantity	Price per Item
34587	Learning Python, Mark Lutz	4	40.95
98762	Programming Python, Mark Lutz	5	56.80
77226	Head First Python, Paul Barry	3	32.95
88112	Einführung in Python3, Bernd Klein	3	24.99

Write a Python program, which returns a list with 2-tuples. Each tuple consists of the order number and the product of the price per item and the quantity. The product should be increased by 10,- € if the value of the order is smaller than 100,00 €.

Write a Python program using lambda and map.

```
Orders = [
    [34587,'Learning Python, Mark Lutz', 4,40.95],
    [98762,'Programming Python,Mark Lutz',5,56.80],
    [77226,'Head First Python, Paul Barry',3,32.95],
    [88112,'Einfuhrung in Python3, Bernd Klein',3,24.99]
]
list(map(lambda x : (x[0] , x[2] , x[2] * x[3] + 10 if x[2] * x[3] < 100 else x[2] * x[3]) , Orders))
```

```
[(34587, 4, 163.8),
 (98762, 5, 284.0),
```

```
(77226, 3, 108.85000000000001),
(88112, 3, 84.97)]
```