**Problem Definition: Python program to implement linked list Iterators.**

**Theory:**

An iterator is an object that contains a countable number of values. An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.

**#Expt11**

**#Python program to implement linked list Iterators.**

class Node:

# Singly linked node

def \_\_init\_\_(self, data=None):

self.data = data

self.next = None

class singly\_linked\_list:

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

# Createe an empty list

self.head = None

self.tail = None

self.count = 0

def iterate\_item(self):

# Iterate the list.

current\_item = self.head

while current\_item:

val = current\_item.data

current\_item = current\_item.next

yield val

def append\_item(self, data):

#Append items on the list

node = Node(data)

if self.tail:

self.tail.next = node

self.tail = node

else:

self.head = node

self.tail = node

self.count += 1

items = singly\_linked\_list()

items.append\_item('PHP')

items.append\_item('Python')

items.append\_item('C#')

items.append\_item('C++')

items.append\_item('Java')

for val in items.iterate\_item():

print(val)

print("\nhead.data: ",items.head.data)

print("tail.data: ",items.tail.data)

OUTPUT:

PHP

Python

C#

C++

Java

head.data: PHP

tail.data: Java