DOUBLY LINKED LIST

ARUNTHATHI N

23102009

BTECH.AI&DS

1.Doubly linked list full implementation.

Aim:

To find the implementation of Doubly linked list.

Algorithem:

Step1:Start

Step2: Definition of a Node in a singly linked list

Step3: Data part of the node

Step4: Constructor to initialize the node with data

Step5: Function to print the linked list

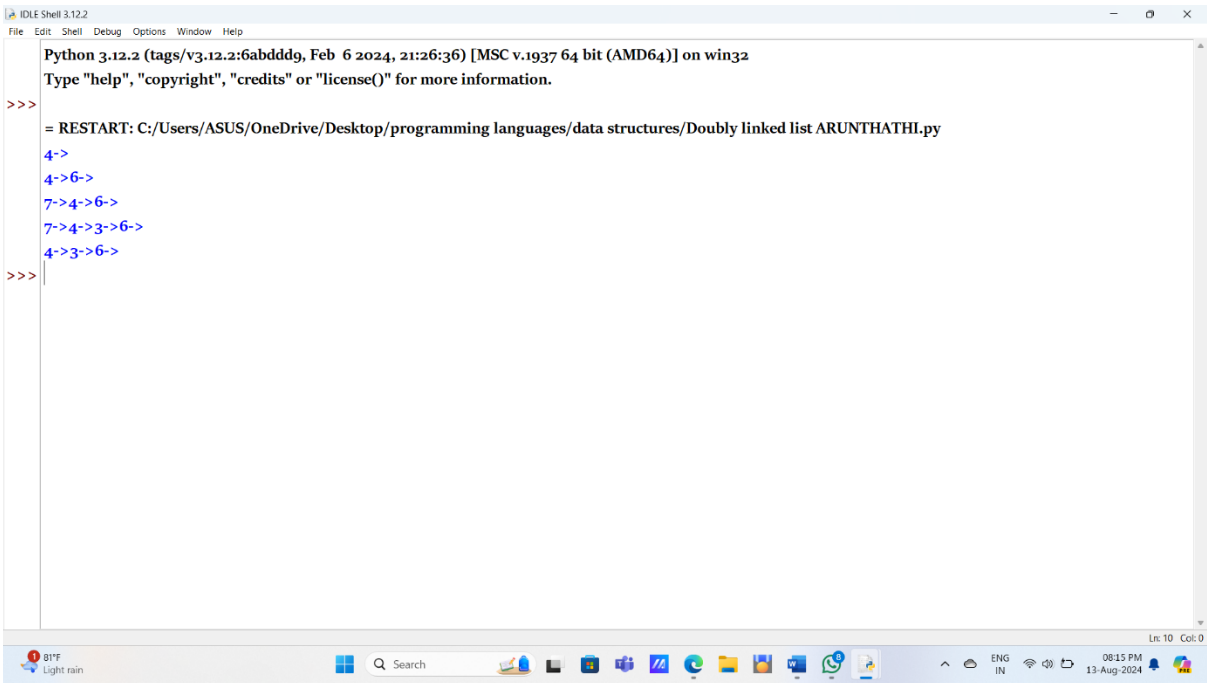
Step6: Printing the above list

Step7:End

Program:

```
#DoublyLinked
class Node:
   def _init_(self,data):
      self.Prev = None
      self.data = data
      self.Next = None
class DoublyLinkedList:
   def _init_(self):
      self.head = None
      self.tail = None
   def InsertEnd(self,data):
      NewNode = Node(data)
      if self.head is None:
         self.head = NewNode
         self.tail = NewNode
      else:
         NewNode.Prev = self.tail
         self.tail.Next = NewNode
         self.tail = NewNode
   def InsertHead(self,data):
      NewNode = Node(data)
      if self.head is None:
         self.head = NewNode
         self.tail = NewNode
      else:
         NewNode.Next  = self.head
         self.head.Prev = NewNode
         self.head = NewNode
   def InsertPos(self,data,pos):
      NewNode = Node(data)
      if self.head is None:
         self.head = NewNode
         self.tail = NewNode
      else:
         c = 0
         temp = self.head
         while(temp.Next!=None):
             if (c == pos-1):
                 break
             c = c+1
             temp = temp.Next
         NewNode.Next = temp.Next
         NewNode.Prev = temp
         temp.Next= NewNode
         NewNode.Next.Prev = NewNode
   def Display(self):
      temp = self.head
      while(temp!=None):
         print(temp.data,end = '->')
         temp = temp.Next
      print()
   def DeleteEnd(self):
      temp = self.tail.Prev
      temp.Next = self.tail.Next
      tempDel = self.tail
      self.tail = temp
      del(tempDel)
   def DeletePos(self,Pos):
      c = 0
      temp = self.head
      while(temp.Next!=None):
         if (c == Pos):
             break
         c = c+1
         temp = temp.Next
      prev = temp.Prev
      next1 = temp.Next
      prev.Next = next1
      next1.Prev = prev
   def DeleteHead(self):
      temp = self.head.Next
      temp.Prev = self.head.Prev
      tempDel = self.head
      self.head = temp
      del(tempDel)
Doub = DoublyLinkedList()
Doub.InsertEnd(4)
Doub.Display()
Doub.InsertEnd(6)
Doub.Display()
Doub.InsertHead(7)
Doub.Display()
Doub.InsertPos(3,2)
Doub.Display()
Doub.DeleteHead()
Doub.Display()
```

Output:



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/OneDrive/Desktop/programming languages/data structures/Doubly linked list ARUNTHATHI.py
4->
4->6->
7->4->6->
7->4->3->6->
4->3->6->
>>>
```

Result:

Thus full implementation of Doubly linked list is performed.

2.Circular Doubly linked list

Aim:

To find the implementation of Doubly linked list.

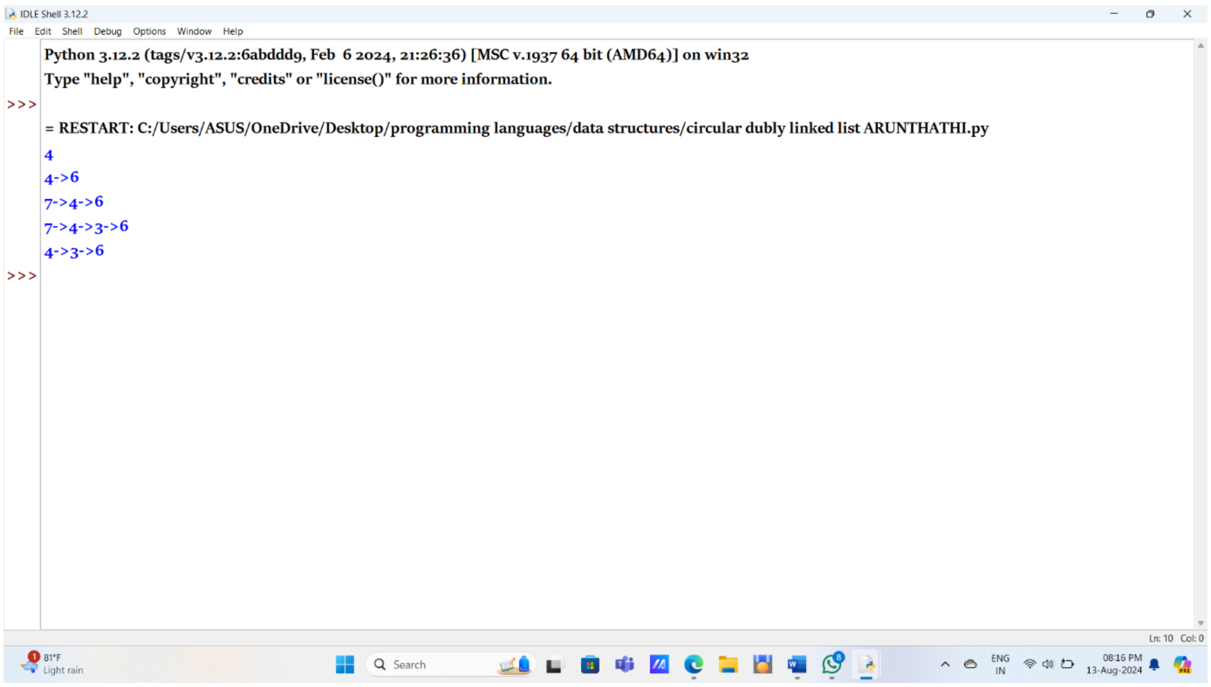
Algorithem:

```
Step1:Start
Step2: Definition of a Node in a singly linked list
Step3: Data part of the node
Step4: Constructor to initialize the node with data
Step5: Function to print the linked list
Step6: Printing the above list
Step7:End
```

Program:

```
#CircularDoubly
class Node:
    def _init_(self,data):
        self.Prev = None
        self.data = data
        self.Next = None
class DoublyLinkedList:
    def _init_(self):
        self.head = None
        self.tail = None
    def InsertEnd(self,data):
        NewNode = Node(data)
        if self.head is None:
            self.head = NewNode
            self.tail = NewNode
            self.head.Next = self.head
            self.tail.Prev = self.head
        else:
            NewNode.Next = self.tail.Next
            NewNode.Prev = self.tail
            self.tail.Next = NewNode
            self.tail = NewNode
    def InsertHead(self,data):
        NewNode = Node(data)
        if self.head is None:
            self.head = NewNode
            self.tail = NewNode
            self.head.Next = self.head
            self.tail.Prev = self.head
        else:
            NewNode.Prev = self.head.Prev
            NewNode.Next  = self.head
            self.head.Prev = NewNode
            self.head = NewNode
    def InsertPos(self,data,pos):
        NewNode = Node(data)
        if self.head is None:
            self.head = NewNode
            self.tail = NewNode
        else:
            c = 0
            temp = self.head
            while(temp.Next!=None):
                if (c == pos-1):
                    break
                c = c+1
                temp = temp.Next
            NewNode.Next = temp.Next
            NewNode.Prev = temp
            temp.Next= NewNode
            NewNode.Next.Prev = NewNode
    def Display(self):
        if self.head == self.tail:
            print(self.head.data)
        else:
            temp = self.head
            while(temp!=self.tail):
                print(temp.data,end = '->')
                temp = temp.Next
            print(self.tail.data)
    def DeleteEnd(self):
        temp = self.tail.Prev
        temp.Next = self.tail.Next
        tempDel = self.tail
        self.tail = temp
        del(tempDel)
    def DeletePos(self,Pos):
        c = 0
        temp = self.head
        while(temp.Next!=None):
            if (c == Pos):
                    break
                c = c+1
                temp = temp.Next
            prev = temp.Prev
            next1 = temp.Next
            prev.Next = next1
            next1.Prev = prev
        def DeleteHead(self):
            temp = self.head.Next
            temp.Prev = self.head.Prev
            tempDel = self.head
            self.head = temp
            del(tempDel)
    Doub = DoublyLinkedList()
    Doub.InsertEnd(4)
    Doub.Display()
    Doub.InsertEnd(6)
    Doub.Display()
    Doub.InsertHead(7)
    Doub.Display()
    Doub.InsertPos(3,2)
    Doub.Display()
    Doub.DeleteHead()
    Doub.Display()
```

Output:



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ASUS/OneDrive/Desktop/programming languages/data structures/circular dubly linked list ARUNTHATHI.py
4
4->6
7->4->6
7->4->3->6
4->3->6
>>>
```

Result:

Thus full implementation of Circular doubly linked list is performed.