Code:

"""

First Exercise E#1 - Using LinkedList and Node classes

Use the LinkedList and Node class to manipulate a LinkedList doing the following operations:

- Include in this order the following numbers at the beginning of the list (they will be in reverse order because of it):

 - 76, 88, 11, 34, 56, 91;

- Print out the current status of the list;

- Push the Current to the third element of the list;

- Remove the next to the current element;

- Insert 23 next to the current element of the list;

- Print out the current status of the list.

You have to submit a .pdf file with your main code, plus the output for the executions above.

"""

```
class Node:
  def __init__(self, d):
    self.Data = d
    self.Next = None

class Linkedlist:
  def __init__(self, d=None):
    if (d == None): # an empty list
      self.Header = None
```

```
        self.Current = None
    else:
        self.Header = Node(d)
        self.Current = self.Header

def nextCurrent(self):
    if (self.Current.Next is not None):
        self.Current = self.Current.Next
    else:
        self.Current = self.Header

def resetCurrent(self):
    self.Current = self.Header

def getCurrent(self):
    if (self.Current is not None):
        return self.Current.Data
    else:
        return None

def insertBeginning(self, d):
    if (self.Header is None): # if list is empty
        self.Header = Node(d)
        self.Current = self.Header
    else:               # if list not empty
        Tmp = Node(d)
```

```
        Tmp.Next = self.Header
        self.Header = Tmp

    def insertCurrentNext(self, d):
        if (self.Header is None): # if list is empty
            self.Header = Node(d)
            self.Current = self.Header
        else:                   # if list not empty
            Tmp = Node(d)
            Tmp.Next = self.Current.Next
            self.Current.Next = Tmp

    def removeBeginning(self):
        if (self.Header is None): # if list is empty
            return None
        else:                   # if list not empty
            ans = self.Header.Data
            self.Header = self.Header.Next
            self.Current = self.Header
            return ans

    def removeCurrentNext(self):
        if (self.Current.Next is None): # if there is no node
            return None             #       after Current
        else:                       # if there is
            ans = self.Current.Next.Data
```

```
            self.Current.Next = self.Current.Next.Next
            return ans

    def printList(self,msg="====="):
        p = self.Header
        print("====",msg)
        while (p is not None):
            print(p.Data, end=" ")
            p = p.Next
        if (self.Current is not None):
            print("Current:", self.Current.Data)
        else:
            print("Empty Linked List")
        input("----------------")

def main():
    mylist = Linkedlist()
    mylist.printList("List created")
    mylist.insertBeginning(76)
    mylist.printList("Inserting 76 at Beginning")
    mylist.insertBeginning(88)
    mylist.printList("Inserting 88 at Beginning")
    mylist.insertBeginning(11)
    mylist.printList("Inserting 11 at Beginning")
    mylist.insertBeginning(34)
    mylist.printList("Inserting 34 at Beginning")
```

```
    mylist.insertBeginning(56)
    mylist.printList("Inserting 56 at Beginning")
    mylist.insertBeginning(91)
    mylist.printList("Inserting 91 at Beginning")
    mylist.resetCurrent()
    mylist.printList("Reseting the Current")
    mylist.nextCurrent()
    mylist.printList("Moving the Current to the next (circularly)")
    mylist.nextCurrent()
    mylist.printList("Moving the Current to the next (circularly)")
    print("The current is:",mylist.getCurrent())
    print(mylist.removeCurrentNext())
    mylist.printList("Removing next the Current")
    mylist.insertCurrentNext(23)
    mylist.printList("Inserting 23 next the Current")
    mylist.printList("Current status of the list")


main()
```

Output:

```
==== List created
Empty Linked List
----------------
==== Inserting 76 at Beginning
76 Current: 76
```

==== Inserting 88 at Beginning

88 76 Current: 76

==== Inserting 11 at Beginning

11 88 76 Current: 76

==== Inserting 34 at Beginning

34 11 88 76 Current: 76

==== Inserting 56 at Beginning

56 34 11 88 76 Current: 76

==== Inserting 91 at Beginning

91 56 34 11 88 76 Current: 76

==== Reseting the Current

91 56 34 11 88 76 Current: 91

==== Moving the Current to the next (circularly)

91 56 34 11 88 76 Current: 56

==== Moving the Current to the next (circularly)

91 56 34 11 88 76 Current: 34

The current is: 34

11

==== Removing next the Current

91 56 34 88 76 Current: 34

==== Inserting 23 next the Current

91 56 34 23 88 76 Current: 34

==== Current status of the list

91 56 34 23 88 76 Current: 34