**Name: Omerullah Ansari (65584)**

**Task 7.1**

class Node:

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

self.Info = value

self.Prev = None

self.Next = None

class DoublyLinked:

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

self.Start = Node(value)

def PrintList(self):

if(self.Start == None):

print("List is Empty")

else:

ptr = self.Start

while(ptr != None):

print(ptr.Info)

ptr = ptr.Next

**Task 7.2**

def InsertionStart(self, value):

New = Node(value)

if(self.Start == None):

self.Start = New

else:

self.Start.Prev = New

New.Next = self.Start

self.Start = New

**Task 7.3**

def InsertionEnd(self, value):

New = Node(value)

if(self.Start == None):

self.Start = New

else:

ptr = self.Start

while(ptr.Next != None):

ptr = ptr.Next

ptr.Next = New

New.Prev = ptr

**Task 7.4**

def DeleteFromBegin(self):

if(self.Start == None):

print("List is Already Empty")

elif(self.Start.Next == None):

self.Start = None

else:

self.Start = self.Start.Next

self.Start.Prev = None

**Task 7.5**

def DeleteFromEnd(self):

if(self.Start == None):

print("List is Already Empty")

elif (self.Start.Next == None):

self.Start = None

else:

ptr = self.Start

while(ptr.Next != None):

ptr = ptr.Next

ptr.Prev.Next = None

**Task 7.6**

class Node:

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

self.Info = value

self.Next = None

def Print(self, node):

if(node != None):

print(node.Info)

self.Print(node.Next)

one = Node(1)

one = Node(2)

one = Node(3)

one.Next = two

two.Next = three

three.Next = two

one.Print(one)