**PRACTICAL 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Roll No.:** | 21BCP359 |
| **Division:** | 6 | **Batch:** | G11 |
| **Aim:** | Implement SHA-1 and apply it on Doubly Linked List data. | | |

**Program**

import hashlib

class Node:

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

self.data = data

self.prev = None

self.next = None

class DoublyLinkedList:

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

self.head = None

self.tail = None

def append(self, data):

new\_node = Node(data)

if self.head is None:

self.head = self.tail = new\_node

else:

new\_node.prev = self.tail

self.tail.next = new\_node

self.tail = new\_node

def get\_concatenated\_data(self):

current = self.head

data\_str = ""

while current:

data\_str += str(current.data)

current = current.next

return data\_str

def apply\_sha1(self):

concatenated\_data = self.get\_concatenated\_data()

sha1\_hash = hashlib.sha1(concatenated\_data.encode())

return sha1\_hash.hexdigest()

dll = DoublyLinkedList()

dll.append("Node1")

dll.append("Node2")

dll.append("Node3")

hash\_result = dll.apply\_sha1()

print("SHA-1 Hash of the Doubly Linked List data:", hash\_result)

**Output**

