

## PRACTICAL 2

<b>Name:</b>	Harsh Shah	<b>Roll No.:</b>	21BCP359
<b>Division:</b>	6	<b>Batch:</b>	G11
<b>Aim:</b>	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

```
PS C:\Users\harsh\OneDrive - pdpu.ac.in\HARSH\_PDEU\SEM 7\Blockchain\Blockchain Lab> p  
ac.in\HARSH\_PDEU\SEM 7\Blockchain\Blockchain Lab\assignments\2.py"  
SHA-1 Hash of the Doubly Linked List data: 36b2c9ebaccd2ddfdb0892792043a174a494b0a8
```