**LINKED LIST IMPLEMENTATION OF STACK**

**CODE:**

**class Node:**

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

**self.data = data**

**self.next = None**

**class Stack:**

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

**self.top = None**

**def is\_empty(self):**

**return self.top is None**

**def push(self, data):**

**new\_node = Node(data)**

**new\_node.next = self.top**

**self.top = new\_node**

**def pop(self):**

**if self.is\_empty():**

**return None**

**popped\_data = self.top.data**

**self.top = self.top.next**

**return popped\_data**

**def peek(self):**

**if self.is\_empty():**

**return None**

**return self.top.data**

**stack = Stack()**

**stack.push(1)**

**stack.push(2)**

**stack.push(3)**

**print("Peek:", stack.peek())**

**print("Pop :", stack.pop())**

**print("Pop :", stack.pop())**

**print("Peek:", stack.peek())**

**print("Is empty:", stack.is\_empty())**

**OUTPUT:**

**Peek: 3**

**Pop : 3**

**Pop : 2**

**Peek: 1**

**Is empty: False**