```
class Node:
    def __init__(self,data):
        self.data = data
        self.prev = prev = None
        self.next = next = None
class Lists:
    def __init__(self):
        self.head = None
  def push(self,data):
        new_node = Node(data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node
 def insert_after(self,prev_node,data):
        if prev_node is None:
            print('No previous node')
            return
        new_node = Node(data)


        new_node.next = prev_node.next
        prev_node.next = new_node
        new_node.prev = prev_node
        if new_node.next is not None:
        new_node.next.prev = new_node
```

```
  def append(self,data):
      new_node = Node(data)
      #set the next of the new node to none since it will be the new tail node
      new_node.next = None
      if self.head is None:
          self.head = new_node


          find_last = self.head
      while(find_last.next):
          find_last = find_last.next
      #point last node to new node
      find_last.next = new_node
      new_node.prev = find_last
def print_list(self):
      node = self.head
      print('In order: ')
      while(node):
          print(node.data)
           reversed order
          find_last = node
          node = node.next
      print('Reversed order: ')


      while(find_last):
```

```
print(find_last.data)

find_last = find_last.prev
```