class Node:

def \_init\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_init\_(self):

self.head = None

def insert\_at\_beginning(self, data):

new\_node = Node(data)

new\_node.next = self.head

self.head = new\_node

def insert\_at\_end(self, data):

new\_node = Node(data)

if self.head is None:

self.head = new\_node

return

last = self.head

while last.next:

last = last.next

last.next = new\_node

def insert\_at\_position(self, position, data):

if position == 0:

self.insert\_at\_beginning(data)

return

new\_node = Node(data)

current = self.head

for \_ in range(position - 1):

if current is None:

raise IndexError("Position out of bounds")

current = current.next

new\_node.next = current.next

current.next = new\_node

def display(self):

current = self.head

while current:

print(current.data, end=" -> ")

current = current.next

print("None")

if \_name\_ == "\_main\_":

linked\_list = LinkedList()

# Insert at the beginning

linked\_list.insert\_at\_beginning(3)

linked\_list.insert\_at\_beginning(2)

linked\_list.insert\_at\_beginning(1)

# Insert at the end

linked\_list.insert\_at\_end(4)

linked\_list.insert\_at\_end(5)

# Insert at a specific position

linked\_list.insert\_at\_position(3, 10) # Insert 10 at position 3 (0-based index)

# Display the linked list

linked\_list.display()