

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None

    def add_to_end(self, data):
        new_node = Node(data)
        if self.head is None:
            self.head = new_node
        else:
            current = self.head
            while current.next is not None:
                current = current.next
            current.next = new_node

    def print_list(self):
        if self.head is None:
            print("List is empty.")
        else:
            current = self.head
            while current is not None:
                print(current.data, end=" -> ")
                current = current.next
            print("None")

    def delete_nth_node(self, n):
        if self.head is None:
            print("Cannot delete from an empty list.")
            return

        if n == 1:
            print(f"Deleted: {self.head.data}")
            self.head = self.head.next
            return

        current = self.head
```

```
count = 1

while current is not None and count < n - 1:
    current = current.next
    count += 1

if current is None or current.next is None:
    print("Index out of range.")
else:
    print(f"Deleted: {current.next.data}")
    current.next = current.next.next

l1 = LinkedList()
l1.add_to_end(10)
l1.add_to_end(20)
l1.add_to_end(30)
l1.add_to_end(40)

print("Original List:")
l1.print_list()

l1.delete_nth_node(3)
print("After deleting 3rd node:")
l1.print_list()

l1.delete_nth_node(10)
l1.delete_nth_node(1)
l1.print_list()

l1.delete_nth_node(1)
l1.delete_nth_node(1)
l1.delete_nth_node(1)
```

```
ll.delete_nth_node(1)
```

```
ll.delete_nth_node(1)...
```



Original List:

10 -> 20 -> 30 -> 40 -> None

Deleted: 30

After deleting 3rd node:

10 -> 20 -> 40 -> None

Index out of range.

Deleted: 10

20 -> 40 -> None

Deleted: 20

Deleted: 40

Cannot delete from an empty list.