

<MIDTERM>	
Course Code: CPE 201L	Program: BSCpE
Course Title: Data Structure and Algorithm	Date Performed: September 6, 2025
Section: 2-A	Date Submitted: September 6, 2025
Name: Directo, Hannah Thea B.	Instructor: Engr. Maria Rizette H. Sayo
1.Objectives	
<ol style="list-style-type: none"> 1. To create a singly linked list that contains odd integers from 1 to 29. 2. To show linked list operations: display, append, and delete. 	
2. Discussion	
<p>A LinkedList is a linear data structure made up of node. Each node contains two parts: the data and the reference to the next node. The first node is called the head, and it serves as the starting point of the list. The last node points to null, which marks the end of the list. Linked lists are useful for storing and managing data. It mainly allows efficient insertion and deletion operations compared to arrays.</p>	
3. Materials and Equipment	
<ol style="list-style-type: none"> 1. Windows Operating System 2. MS Word 3. Google Colab 4. GitHub 	
4. Procedure	
<ul style="list-style-type: none"> • Define a node class with data and next fields. • Initialize the head node with the first odd number. • Use a loop to insert the remaining odd numbers (3–29) into the linked list. • Traverse the list to display all nodes. • Append a new node at the end of the linked list. Traverse the list again to confirm the new node is added. • Locate a specific node and delete it by adjusting the next pointer of the previous node. • Display the final linked list. 	

5. Output

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

def display(head):
    current = head
    while current:
        print(current.data, end=" ")
        current = current.next
    print()

def append(head, value):
    if value % 2 == 0:
        print("Please type an odd number")
        return
    new_node = Node(value)
    current = head
    while current.next:
        current = current.next
    current.next = new_node

def delete(head, value):
    if value % 2 == 0:
        print("Please type an odd number")
        return
    current = head
    prev = None
    while current and current.data != value:
        prev = current
        current = current.next
    if current:
        if prev is None:
            head = current.next
        else:
            prev.next = current.next
    return head

ODD_NO = Node(1)
current = ODD_NO
for i in range(3, 30, 2):
    current.next = Node(i)
    current = current.next

print("Odd Numbers from 1 - 30:\n", end=" ")
display(ODD_NO)

append(ODD_NO, (int(input("\nType an odd number to add: "))))
print("\nAfter appending:\n", end=" ")
display(ODD_NO)

head = delete(ODD_NO, int(input("\nType a number to delete: ")))
print("\nAfter deleting:\n", end=" ")
display(ODD_NO)
```

Output:

```
Odd Numbers from 1 - 30:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29

Type an odd number to add: 31

After appending:
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31

Type a number to delete: 21

After deleting:
1 3 5 7 9 11 13 15 17 19 23 25 27 29 31
```

Refer to this link: [MIDTERM.ipynb - Colab](#)

6. Conclusion

The skill test demonstrates a singly linked list program that shows the odd numbers from 1 to 30. The program showed how to display the elements of linked list, add a new number at the end using the append operation, and remove a chosen odd number from the list. The program shows how the nodes are connected and how the links are updated when changes are made. This task helped me understand the basic idea of linked list as a simple way to store and manage data.