



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 6

---

# Singly Linked Lists

---

*Submitted by:*  
Barbas, Steven Jade P.

*Instructor:*  
Engr. Maria Rizette H. Sayo

August, 23, 2025

# I. Objectives

## Introduction

A linked list is an organization of a list where each item in the list is in a separate node. Linked lists look like the links in a chain. Each link is attached to the next link by a reference that points to the next link in the chain. When working with a linked list, each link in the chain is called a Node. Each node consists of two pieces of information, an item, which is the data associated with the node, and a link to the next node in the linked list, often called next.

This laboratory activity aims to implement the principles and techniques in:

- Writing algorithms using Linked list
- Writing a python program that will perform the common operations in a singly linked list

# II. Methods

- Write a Python program to create a singly linked list of prime numbers less than 20. By iterating through the list, display all the prime numbers, the head, and the tail of the list. (using Google Colab)
- Save your source codes to GitHub

# III. Results

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

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

    def append(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        current = self.head
        while current.next:
            current = current.next
        current.next = new_node

    def display(self):
        current = self.head
        while current:
            print(current.data, end=" -> ")
            current = current.next
        print("None")

    def get_head(self):
        return self.head.data if self.head else None

    def get_tail(self):
        current = self.head
        if not current:
            return None
        while current.next:
            current = current.next
        return current.data
```

```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

ll = LinkedList()
for i in range(20):
    if is_prime(i):
        ll.append(i)

print("Prime numbers in the linked list:")
ll.display()

print("Head of the list:", ll.get_head())
print("Tail of the list:", ll.get_tail())
```

Prime numbers in the linked list:  
2 -> 3 -> 5 -> 7 -> 11 -> 13 -> 17 -> 19 -> None  
Head of the list: 2  
Tail of the list: 19

Figure 1 Screenshot of program

This program shows how a singly linked list can be used to keep prime numbers less than 20. Each node in the list stores a prime number and points to the next node. The first node is called the head, and the last node is the tail. A function checks which numbers are prime, and each prime number is added to the list one by one. By going through the list, we can display all the prime numbers in order, as well as show the head (2) and the tail (19).

## IV. Conclusion

In conclusion, the program shows how a singly linked list can be used to store and display prime numbers in order. It proves that linked lists are useful for organizing data step by step, starting from the head and ending at the tail. Overall, this laboratory help me to have a better understanding of how single linked lists function useful for organizing data.

## References

- [1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.