UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 6

Singly Linked Lists

Submitted by:
Maringal, Czer Justine D.

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:

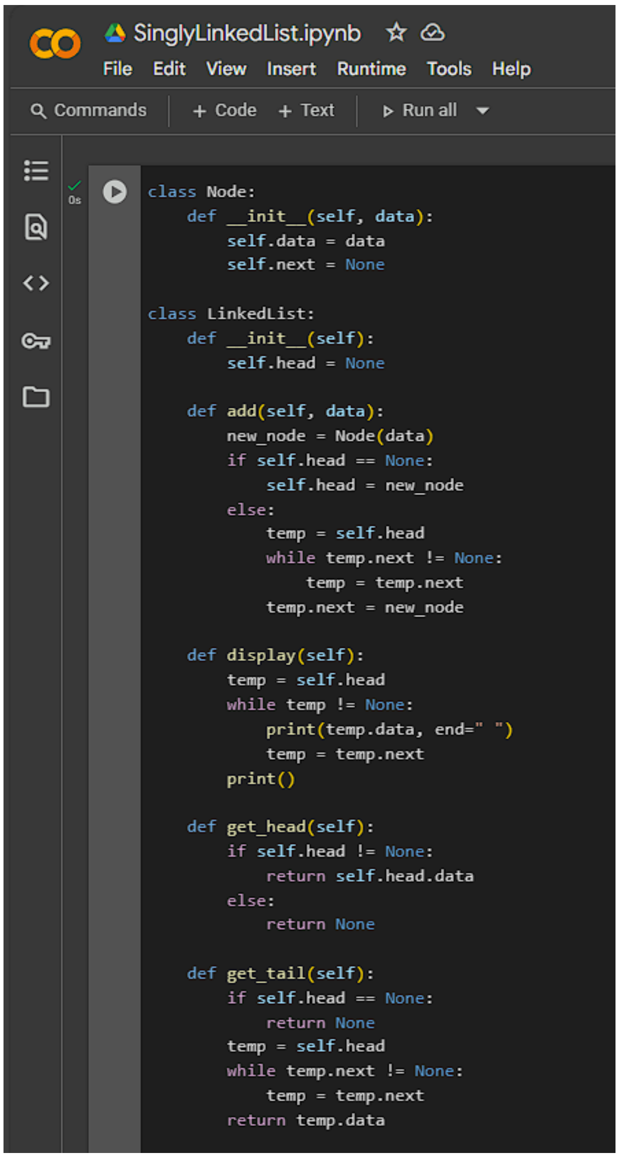
- Writing algorithms using Linked list
- Writing a python program that will perform the common operations in a singly linked list

II. Methods

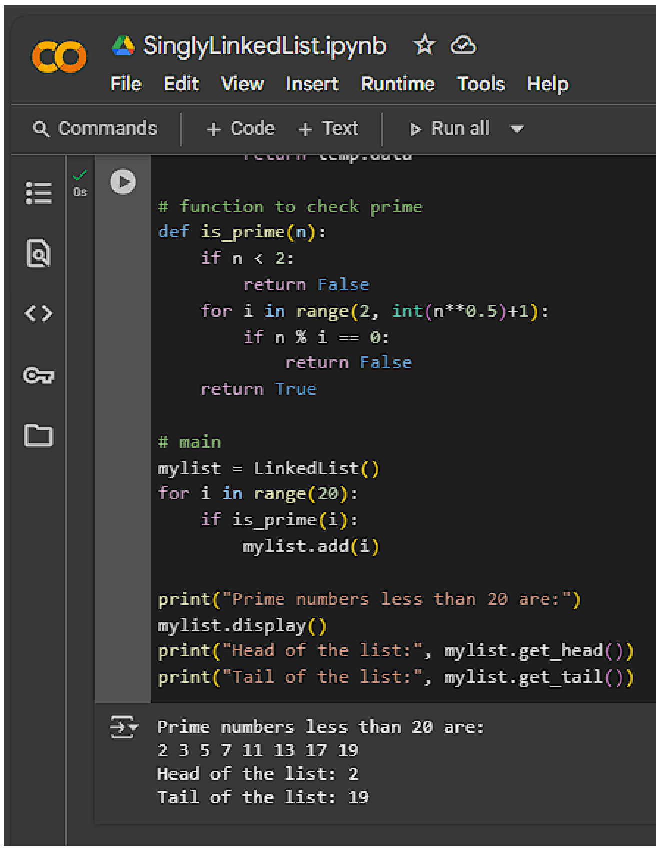
- Write a Python program to create a singly linked list of prime numbers less than 20. By iterating through the list, display all the prime numbers, the head, and the tail of the list. (using Google Colab)
- Save your source codes to GitHub

III. Results

In my program, I created a singly linked list to store all the prime numbers less than 20. First i made a **Node** class with two parts, one for storing the data and another for pointing to the next node. Then i made a **LinkedList** class with functions to add numbers into the list, display all the numbers, and get the head and tail values. I also did a function called **is_prime** that checks if a number is prime or not. In the main part of the program, I looped from 0 to 19 and added only the prime numbers into the linked list. After that i displayed all the prime numbers in the list and printed out the head, which is the first prime number, and the tail, which is the last prime number.



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

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

    def add(self, data):
        new_node = Node(data)
        if self.head == None:
            self.head = new_node
        else:
            temp = self.head
            while temp.next != None:
                temp = temp.next
            temp.next = new_node

    def display(self):
        temp = self.head
        while temp != None:
            print(temp.data, end=" ")
            temp = temp.next
        print()

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

    def get_tail(self):
        if self.head == None:
            return None
        temp = self.head
        while temp.next != None:
            temp = temp.next
        return temp.data
```



```python
        return temp.data

# function to check prime
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5)+1):
        if n % i == 0:
            return False
    return True

# main
mylist = LinkedList()
for i in range(20):
    if is_prime(i):
        mylist.add(i)

print("Prime numbers less than 20 are:")
mylist.display()
print("Head of the list:", mylist.get_head())
print("Tail of the list:", mylist.get_tail())
```

```
Prime numbers less than 20 are:
2 3 5 7 11 13 17 19
Head of the list: 2
Tail of the list: 19
```

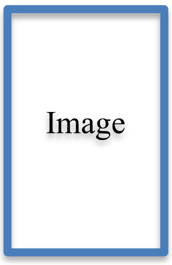


Figure 1 Screenshot of program

IV. Conclusion

In this lesson, I learned how to create and work with a singly linked list. I understand how nodes are connected to form a list and how to add, display, and find the head and tail of the list. I also practiced using a function to check for prime numbers and applied it to store only prime numbers less than 20. This activity helped me how linked lists can be used to organize data and how important it is to understand basic data structures in programming.

References

[1] **W3Schools.** (n.d.). *Python Data Structures.* from https://www.w3schools.com/python/

[2] **GeeksforGeeks.** (n.d.). *Linked List in Python.* from https://www.geeksforgeeks.org/linked-list-set-1-introduction/

[3] **TutorialsPoint.** (n.d.). *Python - Linked Lists.* from https://www.tutorialspoint.com/python_data_structure/python_linked_lists.htm