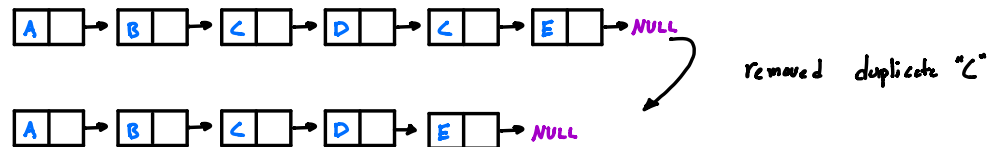


## Singular Linked List- Remove Duplicates



```
def remove_duplicates(self):  
    current = self.head  
    previous = None  
    duplicate_values = dict()  
    while current:  
        if current.data in duplicate_values:  
            # Remove node:  
            previous.next = current.next  
            current = None  
        else:  
            duplicate_values[current.data] = 1  
            previous = current  
        current = previous.next
```

Annotations for the code:

- Keep track of current and previous node (points to `current = self.head` and `previous = None`)
- Create dictionary for keeping track of values (points to `duplicate_values = dict()`)
- loop through the linked list (points to `while current:`)
- Check if our value in the linked list is already in our dictionary (points to `if current.data in duplicate_values:`)
- Change Pointer (points to `previous.next = current.next`)
- Remove duplicate node (points to `current = None`)
- Add to dictionary (count of one) (points to `duplicate_values[current.data] = 1`)
- Proceed to next value (points to `current = previous.next`)