Связанный список

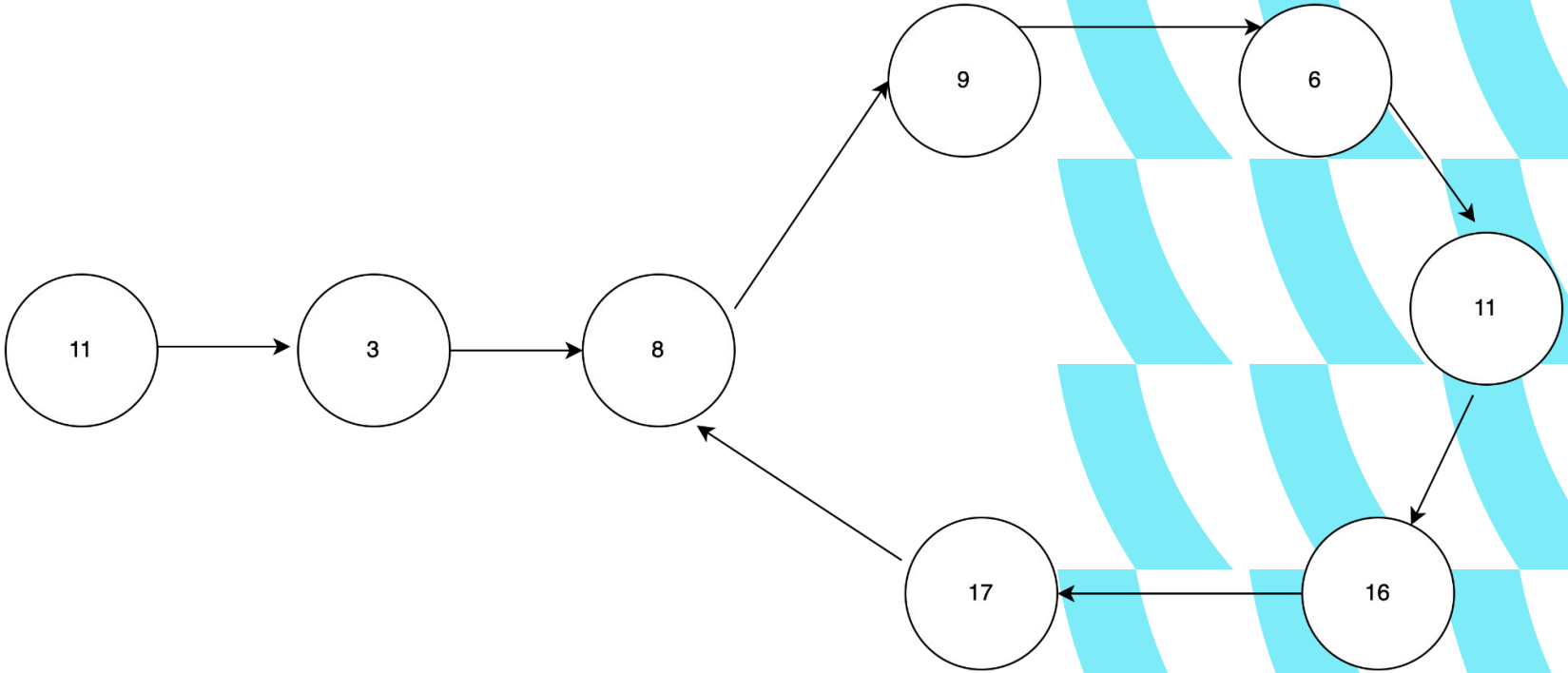
Семинар

Проверить, является ли список циклическим

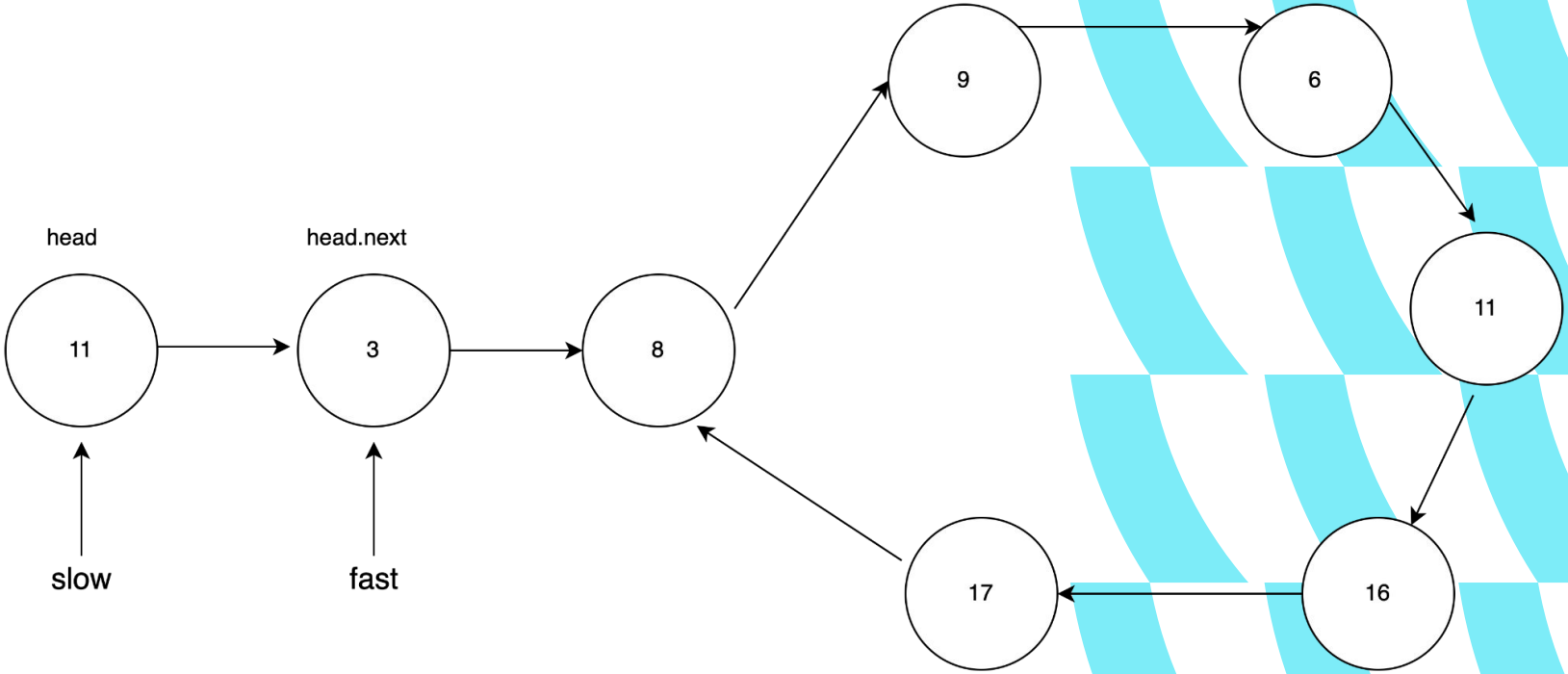
Дан односвязный список. Необходимо проверить, является ли этот список циклическим.

Циклическим (кольцевым) списком называется список у которого последний узел ссылается на один из предыдущих узлов.

Циклический список

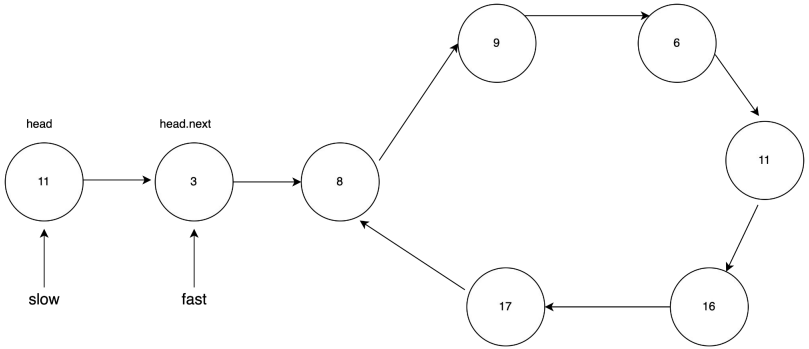


Циклический список



Является ли список циклическим

Проходимся в цикле по нашему списку. Если быстрый указатель дошел до null, значит список не циклический. Если медленный указатель в какой-то момент стал равен быстрому, значит мы нашли цикл.

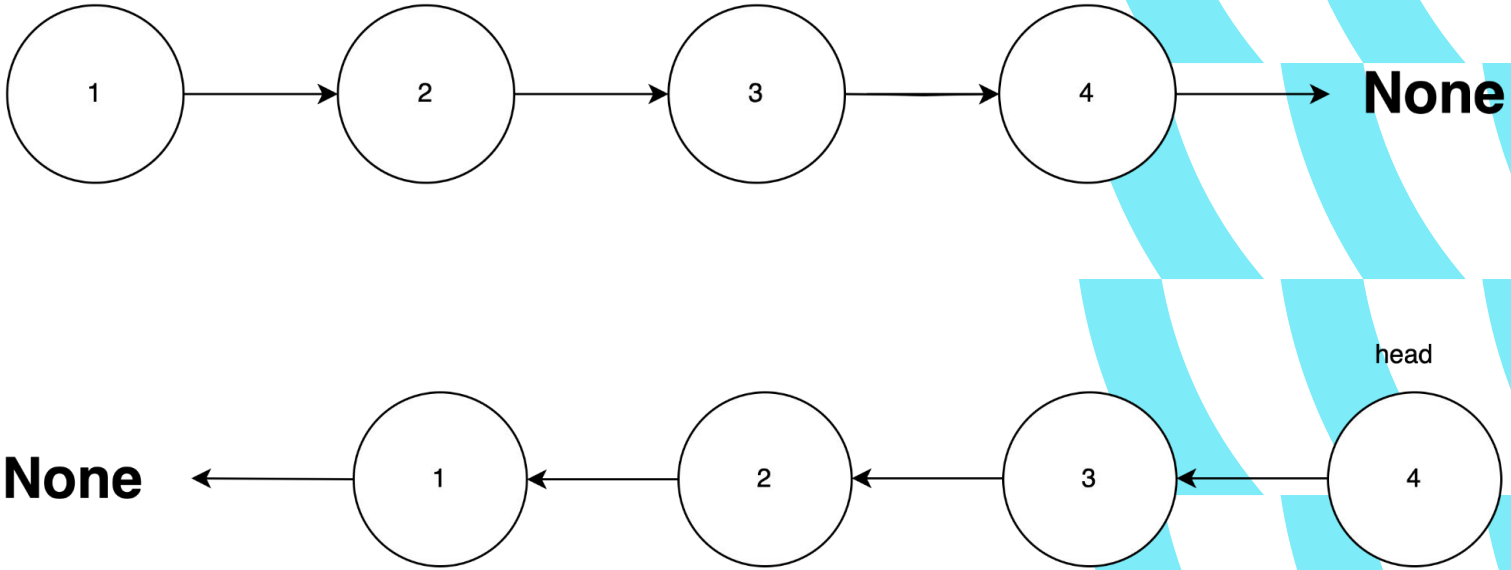



```
function hasCycle(head) {
    if head == null || head.next == null {
        return false
    }

    slow = head
    fast = head.next

    while slow != fast {
        if fast == null || fast.next == null {
            return false
        }
        slow = slow.next
        fast = fast.next.next
    }
    return true
}
```

Развернуть односвязный список

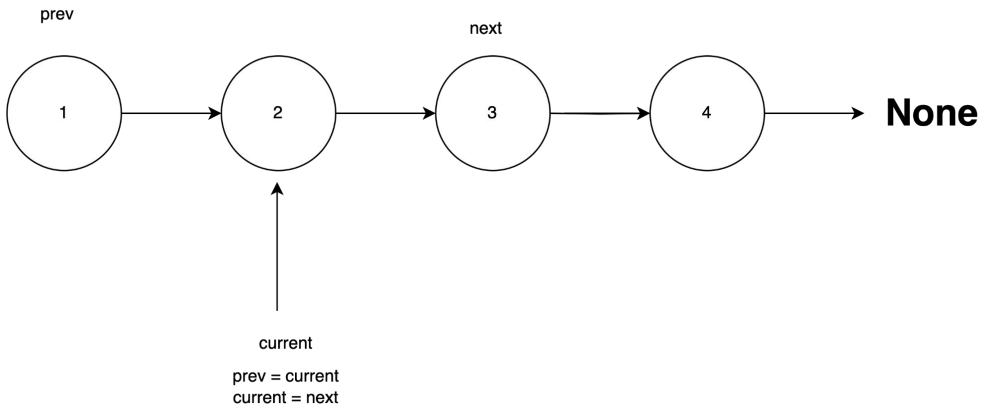
Необходимо написать функцию, которая принимает на вход односвязный список и разворачивает его.

Reverse linked list



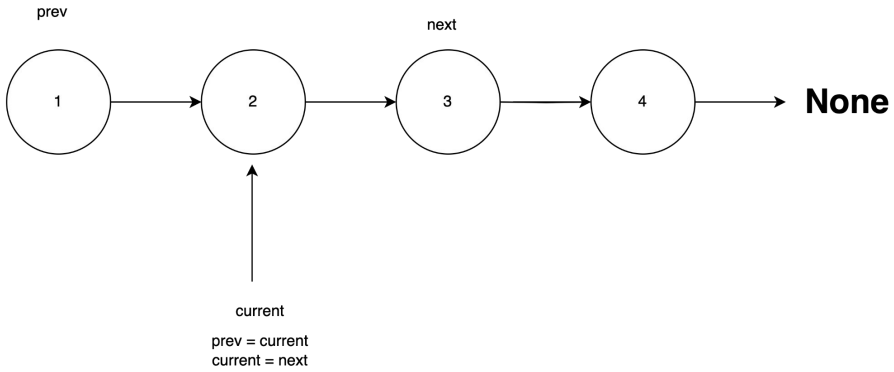
reverseLinkedList

Проходим последовательно в цикле по всему списку. На каждой итерации меняем указатели у узлов



reverseLinkedList

Проходим последовательно в цикле по всему списку. На каждой итерации меняем указатели у узлов



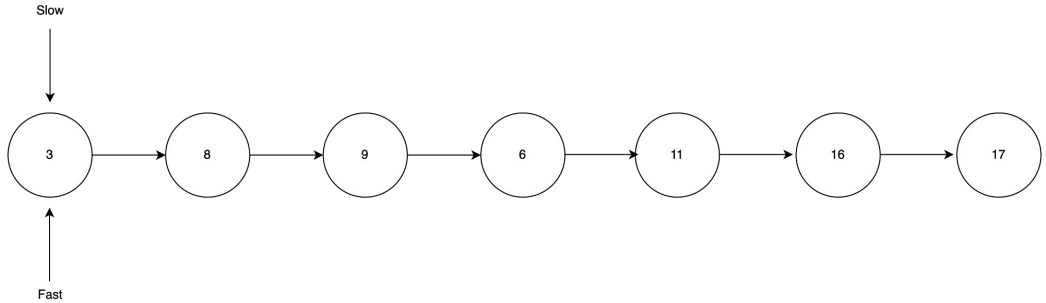

```
function reverseLinkedList(head) {
    prev = null
    current = head

    while (current !== null) {
        next = current.next
        current.next = prev
        prev = current
        current = next
    }

    head = prev
    return head
}
```

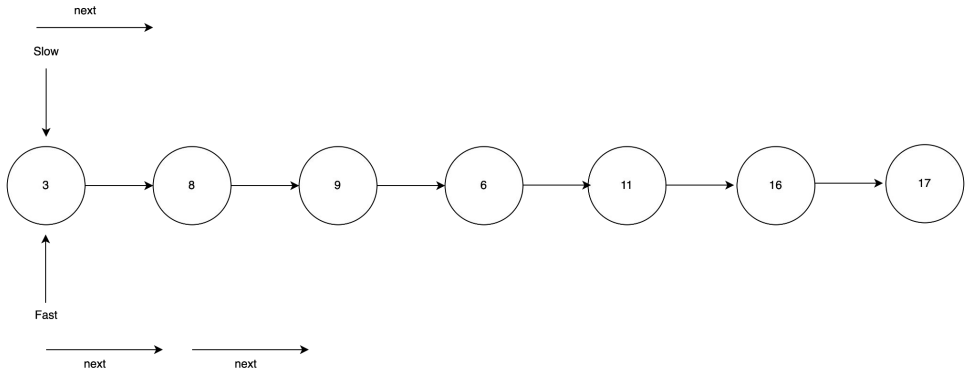
Найти середину списка

Дан связный список. Необходимо найти середину списка. Сделать это необходимо за O(n) без дополнительных аллокаций

Середина связного списка

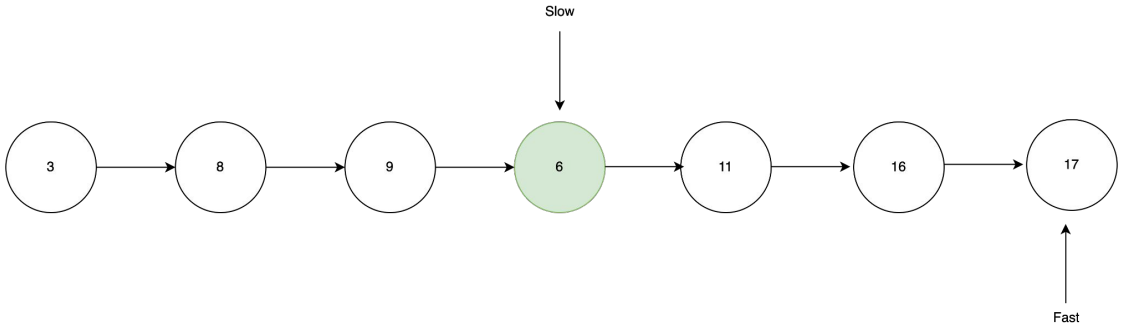
Применим два указателя.
Один будет двигаться вперед на next, а другой на next.next





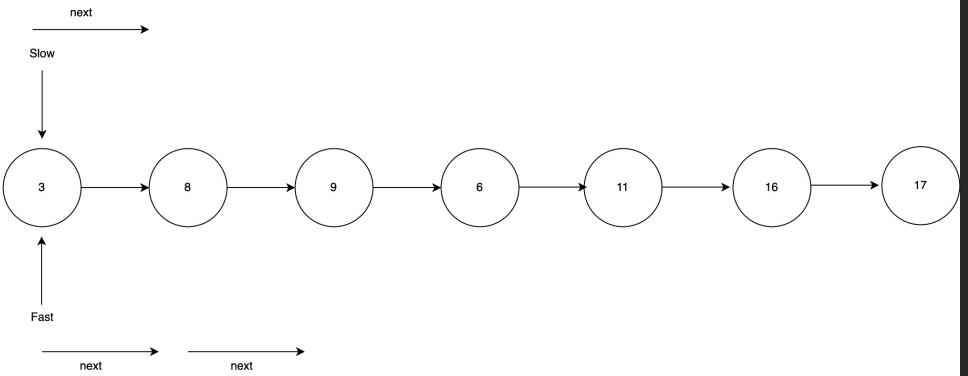
Середина связного списка

Когда быстрый указатель достигнет конца списка, указатель **slow** будет указывать на середину



Середина связного списка

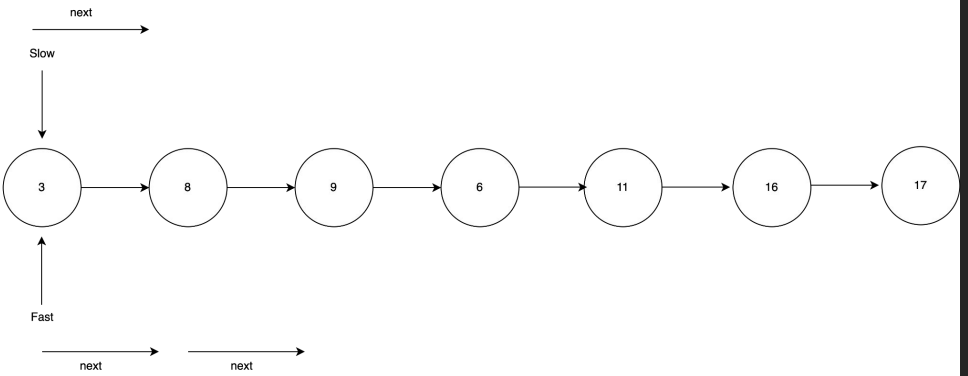
Когда быстрый указатель достигнет конца списка, указатель **slow** будет указывать на середину




```
function middleNode(head) {
    slow = fast = head
    // . . .?
    return slow
}
```

Середина связного списка

Когда быстрый указатель достигнет конца списка, указатель **slow** будет указывать на середину

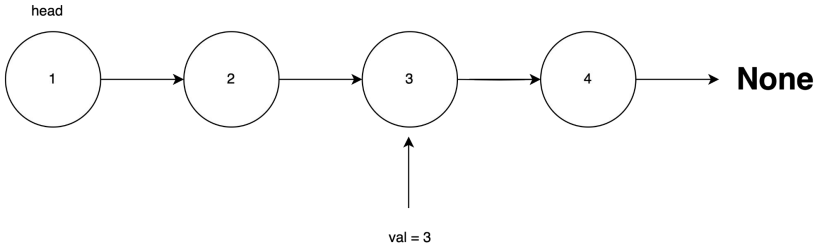


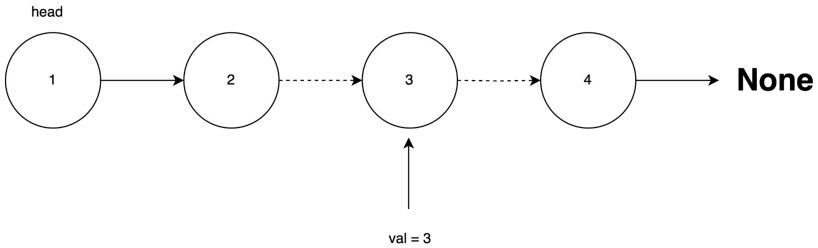

```
function middleNode(head) {
    slow = fast = head
    while fast != null and fast.next != null {
        slow = slow.next
        fast = fast.next.next
    }
    return slow
}
```

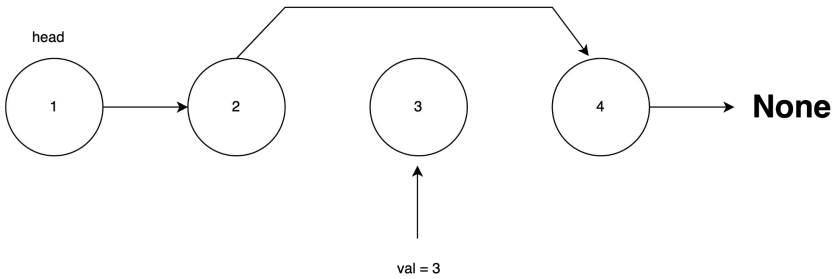
Удалить элемент из односвязного списка

Необходимо написать функцию, которая принимает на вход односвязный список и некоторое целое число **val**. Необходимо удалить узел из списка, значение которого равно val.

Удаление элемента из связного списка

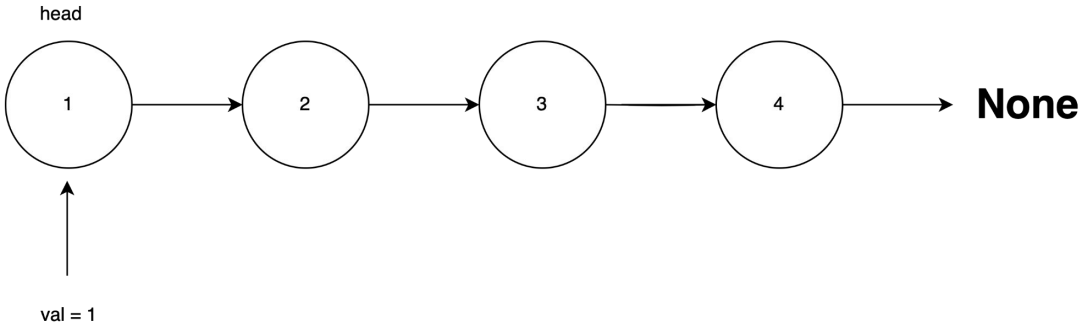






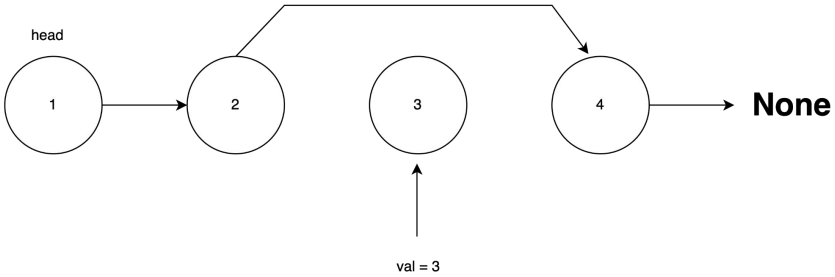
Удаление элемента из связного списка

А как быть если, мы хотим удалить первый элемент списка?



removeElement

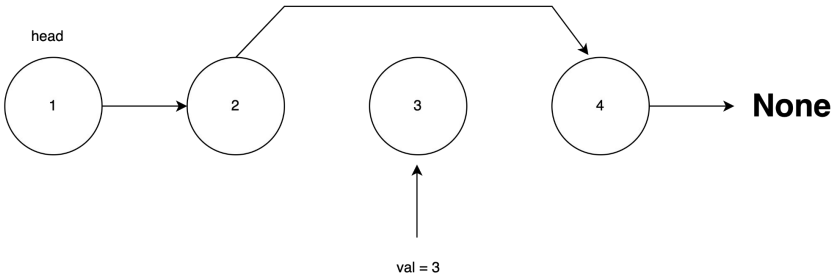
Создаем переменную dummy, основная задача которой - быть заглушкой на случай если придется удалять первый элемент.




```
class ListNode {
    constructor(val) {
        this.val = val
        this.next = null
    }
}

function removeElements(head, val) {
    dummy = new ListNode()
    dummy.next = head
    prev = dummy
    cur = head

    while (cur != null) {
    // . . . ?
    }

    return dummy.next
}
```

removeElement

Создаем переменную dummy, основная задача которой - быть заглушкой на случай если придется удалять первый элемент.




```
class ListNode {
    constructor(val) {
        this.val = val
        this.next = null
    }
}

function removeElements(head, val) {
    dummy = new ListNode()
    dummy.next = head
    prev = dummy
    cur = head

    while (cur != null) {
        if (cur.val == val) {
            prev.next = cur.next
        } else {
            prev = cur
        }
        cur = cur.next
    }

    return dummy.next
}
```

Слияние двух отсортированных списков

Написать функцию, которая принимает на вход два отсортированных односвязных списка и объединяет их в один отсортированный список. При этом затраты по памяти должны быть O(1)

Входные данные: list1 = [3, 6, 8], list2 = [4, 7, 9, 11]
Выходные: [3,4,6,7,8,9,11]

Всем спасибо:)

И хорошего вечера:))