

Deletion of middle node

Leet code - 2

___/___/___

19/2/24

```
struct listnode* deleteMiddle(struct listnode*  
head)
```

```
{
```

```
if (head == NULL || head->next == NULL)
```

```
{
```

```
return NULL;
```

```
}
```

```
struct listnode* slow_ptr = head;
```

```
struct listnode* fast_ptr = head;
```

```
struct listnode* prev = NULL;
```

```
while (fast_ptr != NULL && fast_ptr->next !=  
NULL)
```

```
{
```

```
fast_ptr = fast_ptr->next->next;
```

```
prev = slow_ptr;
```

```
slow_ptr = slow_ptr->next;
```

```
}
```

```
if (prev != NULL)
```

```
{
```

```
prev->next = slow_ptr->next;
```

```
free(slow_ptr);
```

```
}
```

```
else
```

```
{
```



```

    head = head->next;
    free (slow->next);
}
return head;
}

```

Case 1

head =
[1, 3, 4, 7, 1, 2, 6]

output

[1, 3, 4, 1, 2, 6]

expected

[1, 3, 4, 1, 2, 6]

Case 2

head =
[1, 2, 3, 4]

output =

[1, 2, 4]

expected

[1, 2, 4]

Case 3

head
[2, 1]

output
[2]

expected
[2]