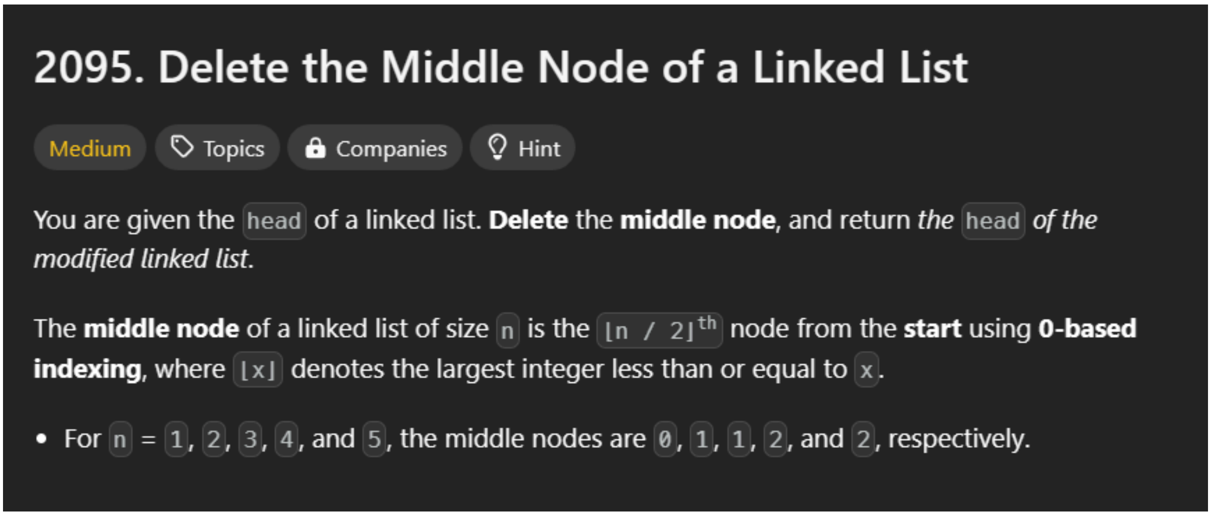
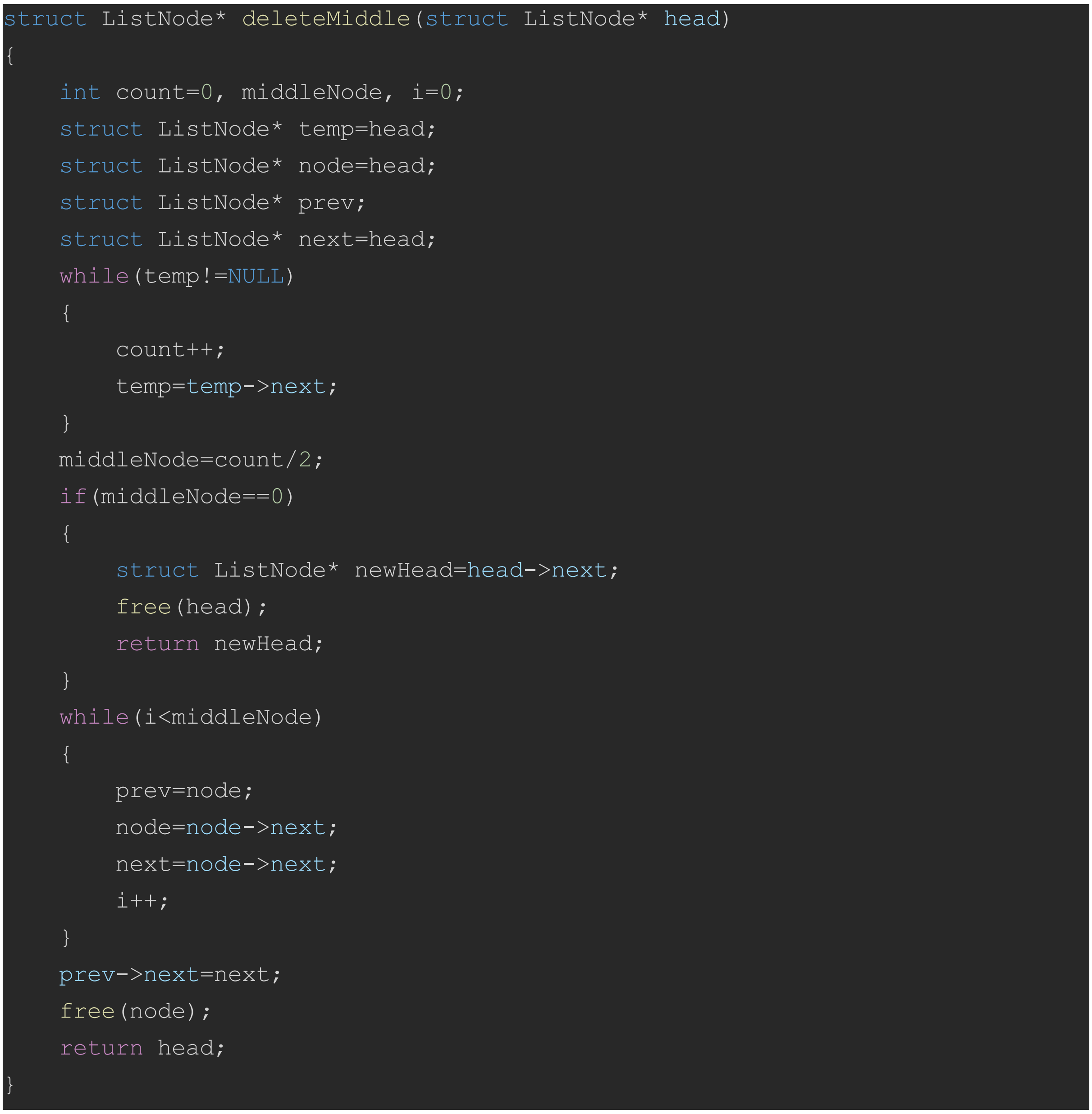


2095. Delete the Middle Node of a Linked List

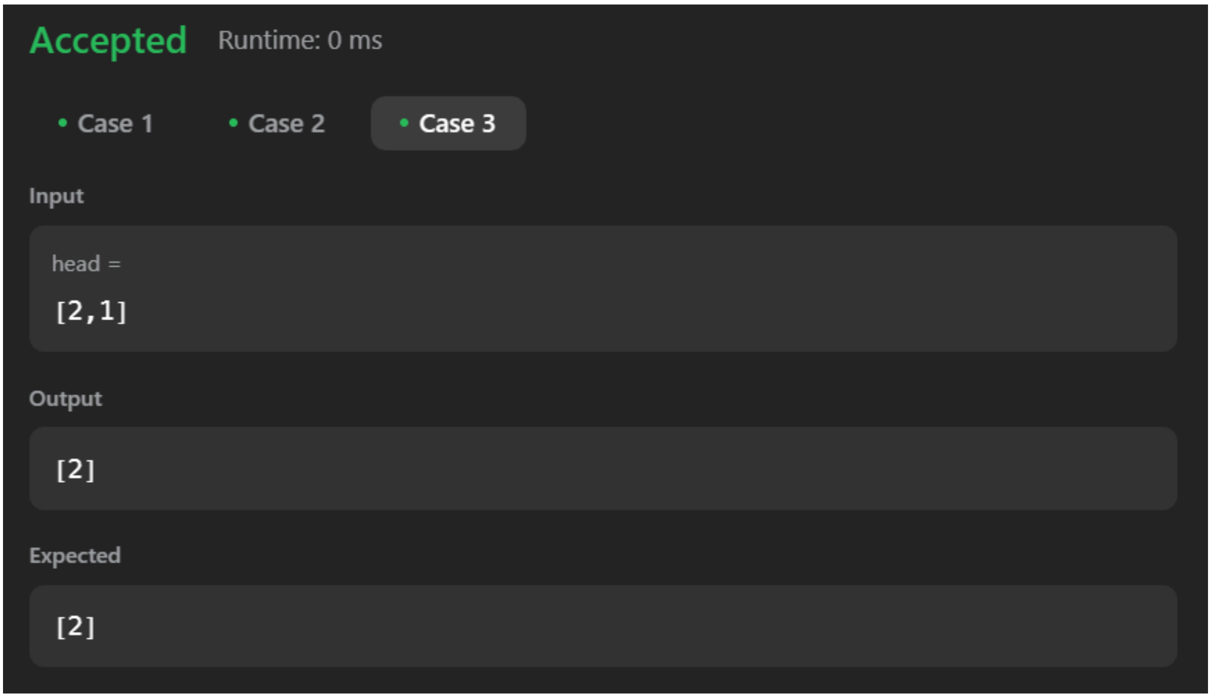
You are given the `head` of a linked list. **Delete** the **middle node**, and return *the* `head` *of the modified linked list.*

The **middle node** of a linked list of size `n` is the $\lfloor n / 2 \rfloor^{th}$ node from the **start** using **0-based indexing**, where $\lfloor x \rfloor$ denotes the largest integer less than or equal to `x`.

- For `n` = `1`, `2`, `3`, `4`, and `5`, the middle nodes are `0`, `1`, `1`, `2`, and `2`, respectively.



```c
struct ListNode* deleteMiddle(struct ListNode* head)
{
    int count=0, middleNode, i=0;
    struct ListNode* temp=head;
    struct ListNode* node=head;
    struct ListNode* prev;
    struct ListNode* next=head;
    while(temp!=NULL)
    {
        count++;
        temp=temp->next;
    }
    middleNode=count/2;
    if(middleNode==0)
    {
        struct ListNode* newHead=head->next;
        free(head);
        return newHead;
    }
    while(i<middleNode)
    {
        prev=node;
        node=node->next;
        next=node->next;
        i++;
    }
    prev->next=next;
    free(node);
    return head;
}
```



Input

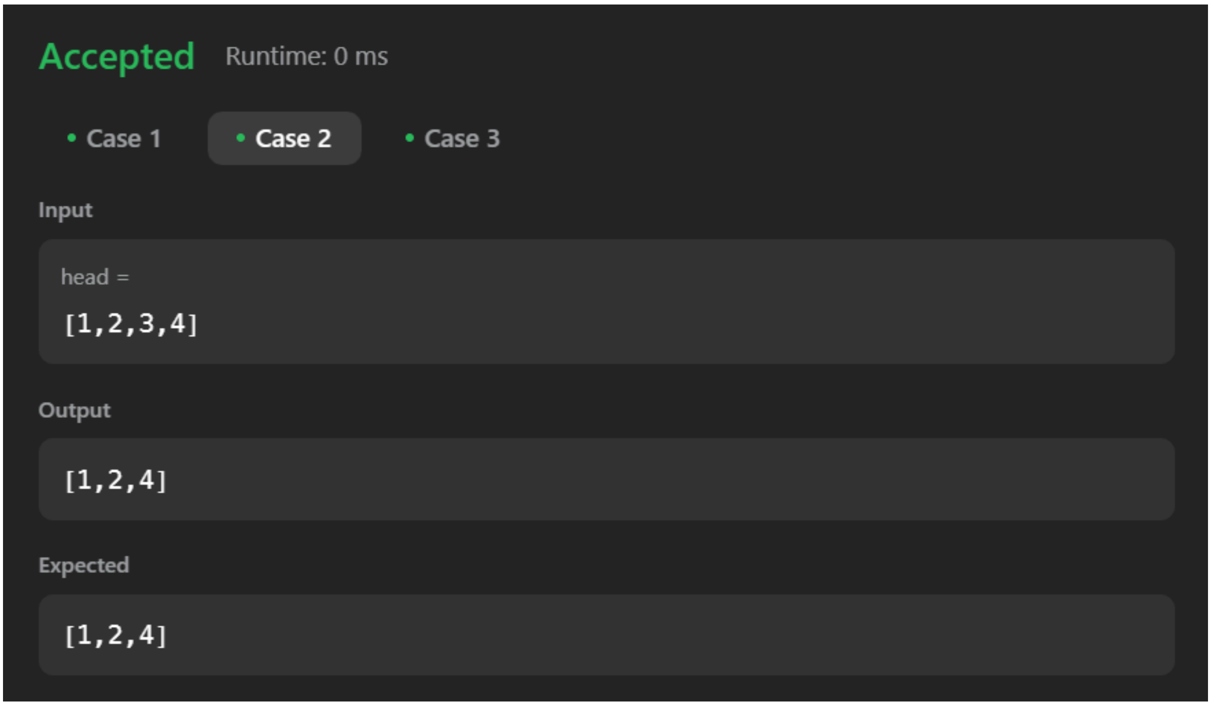
```
head =
[2,1]
```

Output

```
[2]
```

Expected

```
[2]
```



Input

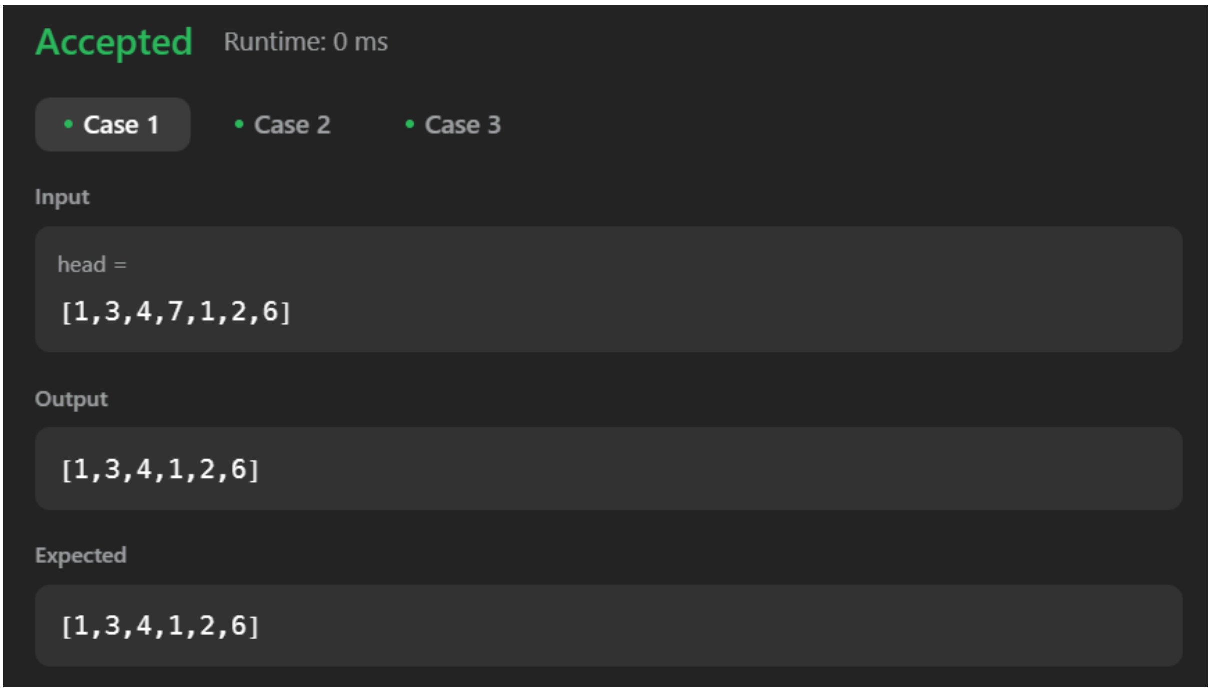
```
head =
[1,2,3,4]
```

Output

```
[1,2,4]
```

Expected

```
[1,2,4]
```



Accepted Runtime: 0 ms

Input

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

Output

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

Expected

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

26/2/24