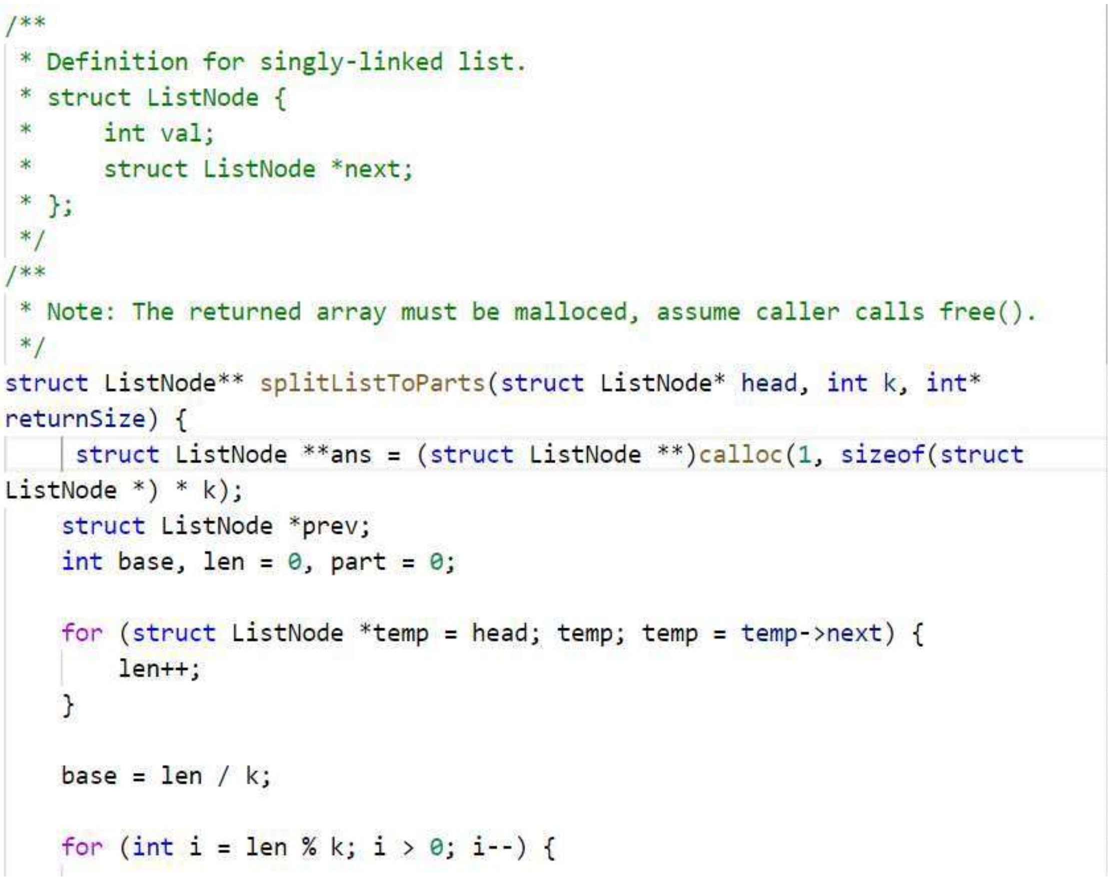
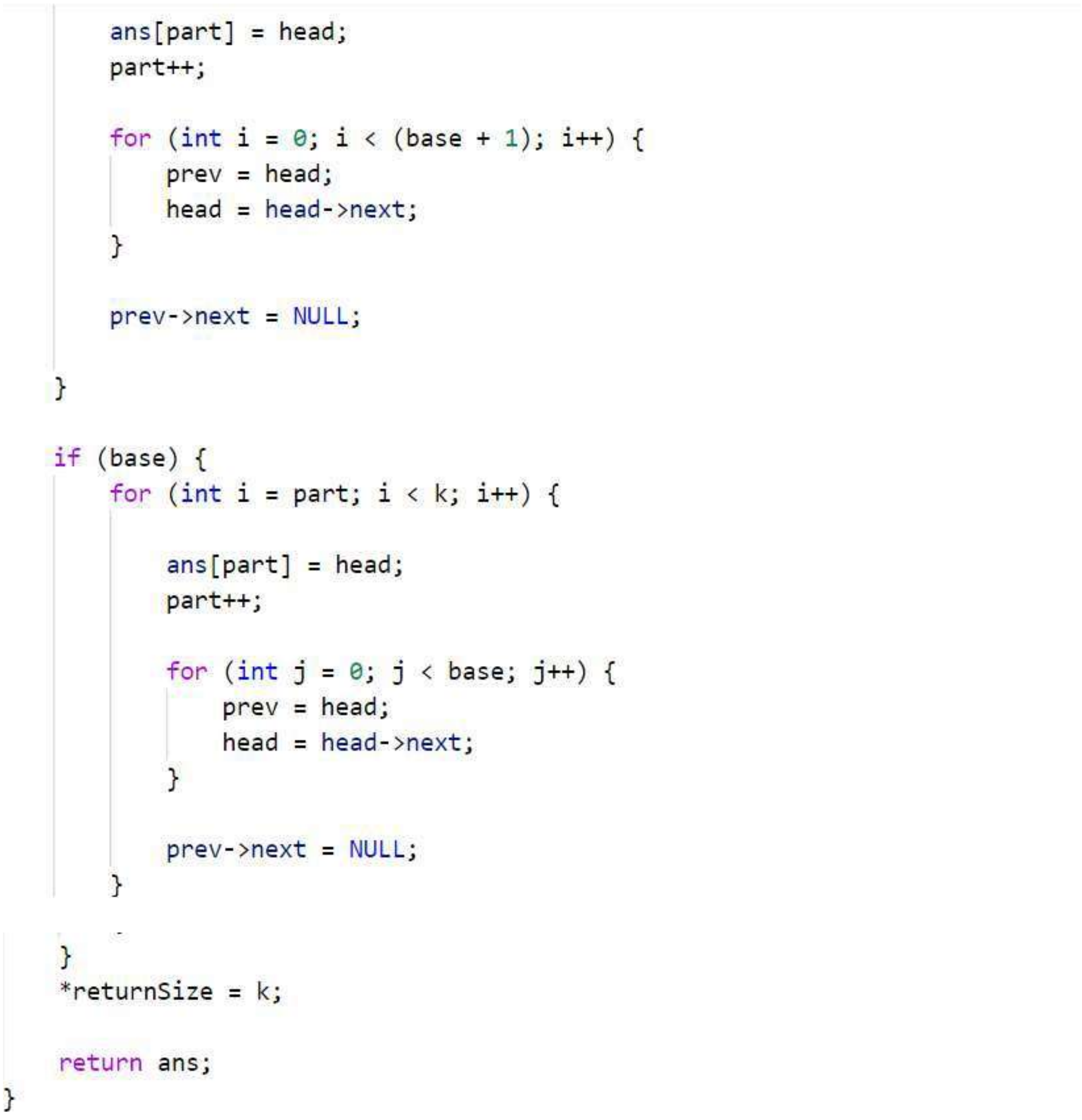
1.Split Linked List in Parts.

Shashank Patel C J

1BM22CS255

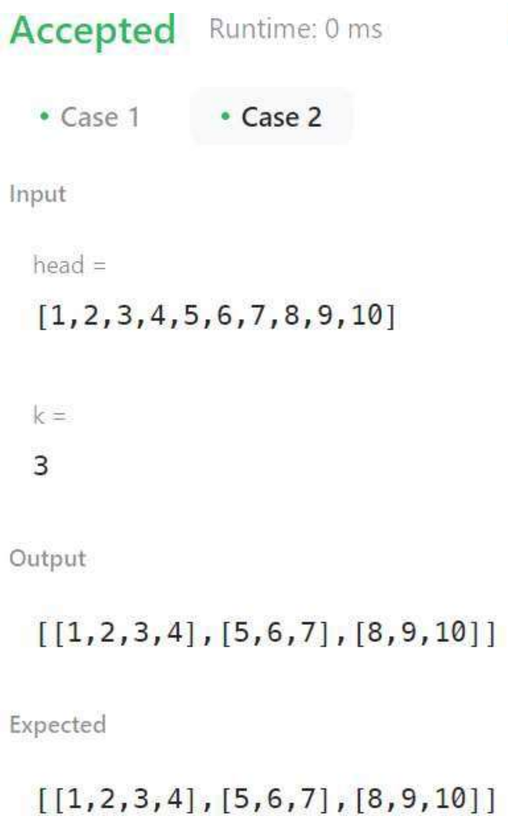


```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
struct ListNode** splitListToParts(struct ListNode* head, int k, int*
returnSize) {
    struct ListNode **ans = (struct ListNode **)calloc(1, sizeof(struct
ListNode *) * k);
    struct ListNode *prev;
    int base, len = 0, part = 0;

    for (struct ListNode *temp = head; temp; temp = temp->next) {
        len++;
    }

    base = len / k;

    for (int i = len % k; i > 0; i--) {
```



```
        ans[part] = head;
        part++;

        for (int i = 0; i < (base + 1); i++) {
            prev = head;
            head = head->next;
        }

        prev->next = NULL;

    }

    if (base) {
        for (int i = part; i < k; i++) {

            ans[part] = head;
            part++;

            for (int j = 0; j < base; j++) {
                prev = head;
                head = head->next;
            }

            prev->next = NULL;
        }

    }
    *returnSize = k;

    return ans;
}
```

Output:



Accepted Runtime: 0 ms

Input

head =

```
[1,2,3,4,5,6,7,8,9,10]
```

k =

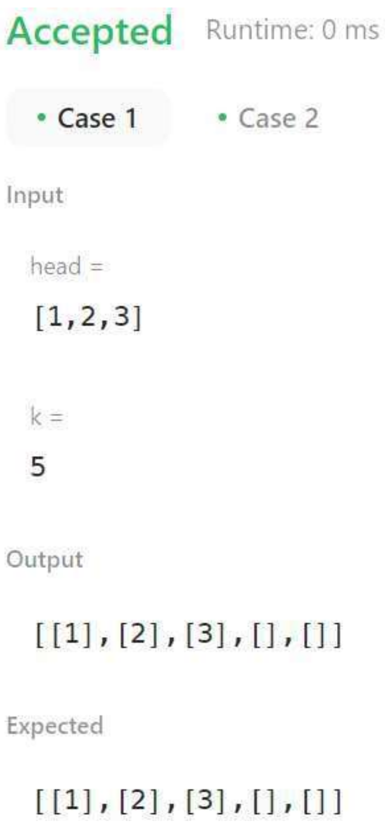
```
3
```

Output

```
[[1,2,3,4],[5,6,7],[8,9,10]]
```

Expected

```
[[1,2,3,4],[5,6,7],[8,9,10]]
```



Accepted Runtime: 0 ms

Input

head =

```
[1,2,3]
```

k =

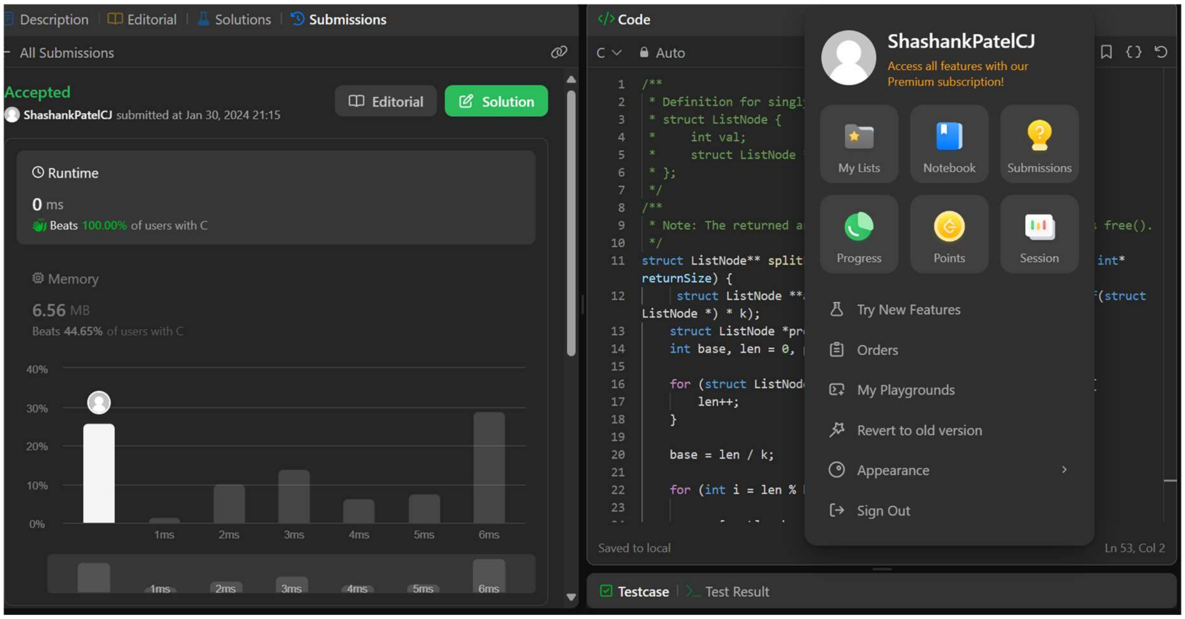
```
5
```

Output

```
[[1],[2],[3],[],[]]
```

Expected

```
[[1],[2],[3],[],[]]
```



Accepted

ShashankPatelCJ submitted at Jan 30, 2024 21:15

Runtime

0 ms

Beats 100.00% of users with C

Memory

6.56 MB

Beats 44.65% of users with C

```
/**
 * Definition for singl
 * struct ListNode {
 *     int val;
 *     struct ListNode
 * };
 */
/**
 * Note: The returned a                                    free().
 */
struct ListNode** split                                    int*
returnSize) {
    struct ListNode **                                     (struct
ListNode *) * k);
    struct ListNode *pr
    int base, len = 0,

    for (struct ListNod
        len++;
    }

    base = len / k;

    for (int i = len %
```