

```

typedef struct ListNode lnode;

int get_len(lnode *head)
{
    int n = 0;
    while(head)
    {
        n++;
        head = head->next;
    }
    return n;
}

struct ListNode** splitListToParts(struct ListNode* head, int k, int*
returnSize)
{
    int n = get_len(head), elems, i, j;
    *returnSize = k;
    lnode **list = (lnode**)calloc(k, sizeof(lnode*)), *t = head;
    if(n>k)
    {
        for(i=0;i<k;i++)
        {
            elems = i<n%k?n/k+1:n/k;
            j=0;
            list[i] = head;
            t = head;
            while(j++<elems)
            {
                t = head;
                head = head->next;
            }
            t->next = NULL;
        }
    }
    else
    {
        for(i=0;i<n;i++)
        {
            list[i] = head;
            head = head->next;
            list[i]->next = NULL;
        }
    }
}

```

```
    return list;  
}
```

Accepted Runtime: 5 ms

• Case 1 • Case 2

Input

head =

[1,2,3]

k =

5

Output

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

Expected

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



♥ [Contribute a testcase](#)

Accepted Runtime: 5 ms

• Case 1 • Case 2

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

 Rushila submitted at Feb 15, 2024 20:45


 Editorial

 Solution


 Runtime

4 ms

 Beats 53.42% of users with C

 Memory

6.58 MB

 Beats 63.98% of users with C

