

### LeetCode - 3 (Split Linked List)

```
typedef struct ListNode {node;
```

```
int getLen(ListNode *head)
{
```

```
    int n = 0;
```

```
    while(head)
```

```
    {
```

```
        n++;
```

```
        head = head->next;
```

```
    }
```

```
    return n;
```

```
}
```

```
struct ListNode** splitListToParts
```

```
(struct ListNode* head, int k, int*
```

```
&returnSize)
```

```
{
```

```
    int n = getLen(head), elems, i, j;
```

```
    *returnSize = k;
```

```
    ListNode** list = (ListNode**) calloc(k, sizeof(ListNode*));
```

```
    *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;
    }
    k → next = NULL;
}
else
{
    list[i] = head;
    head = head → next;
    (list[i]) → next = NULL;
}
}
return list;
}

```

### Output

Case 1:

head = [1, 2, 3]

k = 5

Case 2:

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

k = 3

Get

22/2/24.