725. Split Linked List in Parts

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */
typedef struct ListNode* node;
struct ListNode** splitListToParts(struct ListNode* head, int k, int*
returnSize){
    int cnt = 0;
    node temp = head;
    while (temp != NULL){
        temp = temp->next;
        cnt++;
    }
    int part = cnt / k;
    int rem = cnt % k;
    temp = head;
    node* arr = (node*)malloc(sizeof(node) * k);
    for (int i = 0; i < k; i++){
        node sub = NULL;
        node st = sub;
        for (int j = 0; j < (rem > 0 ? part+1 : part); j++){
            if (sub == NULL) {
                sub = (node)malloc(sizeof(struct ListNode));
                st = sub;
            }
            else{
                st->next = (node)malloc(sizeof(struct ListNode));
                st = st->next;
```

```
            }
            st->val = temp->val;
            st->next = NULL;
            temp = temp->next;
        }
        arr[i] = sub;
        if (rem > 0) {
            rem--;
        }
    }
    *returnSize = k;
    return arr;
}
```

OUTPUT :