

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Definition for singly-linked list.
```

```
struct ListNode {
```

```
    int val;
```

```
    struct ListNode *next;
```

```
};
```

```
// Function to get the length of the linked list
```

```
int getListLength(struct ListNode* head) {
```

```
    int length = 0;
```

```
    struct ListNode* current = head;
```

```
    while (current != NULL) {
```

```
        length++;
```

```
        current = current->next;
```

```
    }
```

```
    return length;
```

```
}
```

```
// Function to split the linked list into k consecutive parts
```

```
struct ListNode** splitListToParts(struct ListNode* head, int k, int* returnSize) {
```

```
    int length = getListLength(head);
```

```
    int minPartLength = length / k;
```

```
    int extraNodes = length % k;
```

```
    struct ListNode** result = (struct ListNode**)malloc(k * sizeof(struct ListNode*));
```

```
    struct ListNode* current = head;
```

```
    for (int i = 0; i < k && current != NULL; i++) {
```

```
        result[i] = current;
```

```

    int partLength = minPartLength + (i < extraNodes ? 1 : 0);
    for (int j = 1; j < partLength; j++) {
        current = current->next;
    }

    struct ListNode* nextNode = current->next;
    current->next = NULL;
    current = nextNode;
}

*returnSize = k;
return result;
}

// Function to print the linked list
void printList(struct ListNode* head) {
    struct ListNode* current = head;
    while (current != NULL) {
        printf("%d -> ", current->val);
        current = current->next;
    }
    printf("NULL\n");
}

int main() {
    // Create a sample linked list
    struct ListNode* head = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->val = 1;
    head->next = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->next->val = 2;
    head->next->next = (struct ListNode*)malloc(sizeof(struct ListNode));
    head->next->next->val = 3;

```

```

head->next->next->next = (struct ListNode*)malloc(sizeof(struct ListNode));

head->next->next->next->val = 4;

head->next->next->next->next = (struct ListNode*)malloc(sizeof(struct ListNode));

head->next->next->next->next->val = 5;

head->next->next->next->next->next = NULL;


printf("Original Linked List:\n");

printList(head);


int k = 3;

int returnSize;

struct ListNode** parts = splitListToParts(head, k, &returnSize);


printf("\nSplit Linked List into %d parts:\n", k);

for (int i = 0; i < returnSize; i++) {

    printf("Part %d: ", i + 1);

    printList(parts[i]);

}


return 0;

}

```

```

Original Linked List:
1 -> 2 -> 3 -> 4 -> 5 -> NULL

Split Linked List into 3 parts:
Part 1: 1 -> 2 -> NULL
Part 2: 3 -> 4 -> NULL
Part 3: 5 -> NULL

...Program finished with exit code 0
Press ENTER to exit console.

```