876. Middle of the Linked list

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

// Definition for singly-linked list.
struct ListNode {
   int val;
   struct ListNode *next;
};

// Function to create a new node
struct ListNode* createNode(int value) {
   struct ListNode* newNode = (struct ListNode*)malloc(sizeof(struct ListNode));
   newNode->val = value;
   newNode->next = NULL;
   return newNode;
}

// Append a node at end of list
void appendNode(struct ListNode** head, int value) {
   struct ListNode* newNode = createNode(value);

   if (*head == NULL) {
      *head = newNode;
      return;
   }

   struct ListNode* temp = *head;
   while (temp->next != NULL) {
```

```
        temp = temp->next;
    }
    temp->next = newNode;
}

struct ListNode* middleNode(struct ListNode* head) {
    struct ListNode* slow = head;
    struct ListNode* fast = head;

    while (fast != NULL && fast->next != NULL) {
        slow = slow->next;
        fast = fast->next->next;
    }

    return slow;  // middle node
}

void printList(struct ListNode* head) {
    struct ListNode* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->val);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct ListNode* head = NULL;

    int n, value;
```

```
    printf("Enter number of nodes: ");
    scanf("%d", &n);

    printf("Enter the values:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &value);
        appendNode(&head, value);
    }

    printf("\nLinked List: ");
    printList(head);

    struct ListNode* mid = middleNode(head);
    printf("Middle node value = %d\n", mid->val);

    return 0;
}
```

```
Enter number of nodes: 4
Enter the values:
1
2
3
4

Linked List: 1 -> 2 -> 3 -> 4 -> NULL
Middle node value = 3
```