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

// Define a structure for a node in the linked list
struct Node {
    int data;
    struct Node* next;
};
void insertAtEnd(struct Node** head, int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    if (newNode == NULL) {
        perror("Memory allocation failed");
        exit(EXIT_FAILURE);
    }
    newNode->data = data;
    newNode->next = NULL;

    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node* current = *head;
        while (current->next != NULL) {
            current = current->next;
        }
        current->next = newNode;
    }
}
void removeOddIndices(struct Node** head) {
    if (*head == NULL || (*head)->next == NULL) {
        return;
    }

    struct Node* current = *head;
    struct Node* nextNode;

    while (current != NULL && current->next != NULL) {
        nextNode = current->next;
        current->next = nextNode->next;
        free(nextNode);
        current = current->next;
    }
}
void printList(struct Node* head) {
    struct Node* current = head;
    while (current != NULL) {
        printf("%d -> ", current->data);
        current = current->next;
    }
    printf("NULL\n");
}
void freeList(struct Node* head) {
    struct Node* current = head;
    struct Node* nextNode;
    while (current != NULL) {
        nextNode = current->next;
        free(current);
        current = nextNode;
    }
}

int main() {
    struct Node* head = NULL;
    insertAtEnd(&head, 1);
    insertAtEnd(&head, 2);
    insertAtEnd(&head, 3);
    insertAtEnd(&head, 4);
```

```c
    insertAtEnd(&head, 5);
    insertAtEnd(&head, 6);
    printf("Original Linked List:\n");
    printList(head);
    removeOddIndices(&head);
    printf("Linked List after Removing Elements with Odd Indices:\n");
    printList(head);
    freeList(head);

    return 0;
}
```