

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // Define a structure for a node in the linked list
5  struct Node {
6      int data;
7      struct Node* next;
8  };
9  void insertAtEnd(struct Node** head, int data) {
10     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
11     if (newNode == NULL) {
12         perror("Memory allocation failed");
13         exit(EXIT_FAILURE);
14     }
15     newNode->data = data;
16     newNode->next = NULL;
17
18     if (*head == NULL) {
19         *head = newNode;
20     } else {
21         struct Node* current = *head;
22         while (current->next != NULL) {
23             current = current->next;
24         }
25         current->next = newNode;
26     }
27 }
28 void removeOddIndices(struct Node** head) {
29     if (*head == NULL || (*head)->next == NULL) {
30         return;
31     }
32
33     struct Node* current = *head;
34     struct Node* nextNode;
35
36     while (current != NULL && current->next != NULL) {
37         nextNode = current->next;
38         current->next = nextNode->next;
39         free(nextNode);
40         current = current->next;
41     }
42 }
43 void printList(struct Node* head) {
44     struct Node* current = head;
45     while (current != NULL) {
46         printf("%d -> ", current->data);
47         current = current->next;
48     }
49     printf("NULL\n");
50 }
51 void freeList(struct Node* head) {
52     struct Node* current = head;
53     struct Node* nextNode;
54     while (current != NULL) {
55         nextNode = current->next;
56         free(current);
57         current = nextNode;
58     }
59 }
60
61 int main() {
62     struct Node* head = NULL;
63     insertAtEnd(&head, 1);
64     insertAtEnd(&head, 2);
65     insertAtEnd(&head, 3);
66     insertAtEnd(&head, 4);

```

```
67     insertAtEnd(&head, 5);
68     insertAtEnd(&head, 6);
69     printf("Original Linked List:\n");
70     printList(head);
71     removeOddIndices(&head);
72     printf("Linked List after Removing Elements with Odd Indices:\n");
73     printList(head);
74     freeList(head);
75
76     return 0;
77 }
```