



King Abdul Aziz University

Faculty of Engineering

2023

EE-463

Operating Systems

Lab-6

M/No	Name	ID
1	Anwar Shukri Shawli	1937123

Instructor: Dr. Abdulghani Al-Qasimi

The fixed code:

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

int number_instantiated = 0;

struct Node {
    int value;
    struct Node* next;
};

struct LinkedList {
    struct Node* head;
};

void insert(struct LinkedList* list, int new_item) {
    struct Node* new_node = (struct Node*)malloc(sizeof(struct Node));
    new_node->value = new_item;
    new_node->next = list->head;
    list->head = new_node;
    printf("Creating Node, %d are in existence right now\n", ++number_instantiated);
}

int remove_item(struct LinkedList* list, int item_to_remove) {
    struct Node* marker = list->head;
    struct Node* temp = NULL; // temp points to one behind as we iterate

    while (marker != NULL) {
        if (marker->value == item_to_remove) {
            if (temp == NULL) { // marker is the first element in the list
                list->head = marker->next;
            } else {
                temp->next = marker->next;
            }
            free(marker);
            printf("Destroying Node, %d are in existence right now\n", --
number_instantiated);
            return 0;
        }
        temp = marker;
        marker = marker->next;
    }
    return -1; // failure
}

void print(struct LinkedList* list) {
    struct Node* marker = list->head;
    while (marker != NULL) {
        printf("%d\n", marker->value);
        marker = marker->next;
    }
}

void delete_nodes(struct LinkedList* list) {
    struct Node* marker = list->head;
    while (marker != NULL) {
        struct Node* temp = marker;
```

```
        marker = marker->next;
        free(temp);
    }
    list->head = NULL; // Set head_ to NULL after deleting all nodes
}

int main(int argc, char** argv) {
    struct LinkedList list;
    list.head = NULL;

    insert(&list, 1);
    insert(&list, 2);
    insert(&list, 3);
    insert(&list, 4);

    printf("The fully created list is:\n");
    print(&list);

    printf("\nNow removing elements:\n");
    remove_item(&list, 4);
    print(&list);
    printf("\n");
    remove_item(&list, 1);
    print(&list);
    printf("\n");
    remove_item(&list, 2);
    print(&list);
    printf("\n");
    remove_item(&list, 3);
    print(&list);

    delete_nodes(&list);

    return 0;
}
```

The output for this code:

```
anwar@lamp ~$ gcc main2.c
anwar@lamp ~$ ./a.out
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Destroying Node, 3 are in existence right now
3
2
1

Destroying Node, 2 are in existence right now
3
2

Destroying Node, 1 are in existence right now
3

Destroying Node, 0 are in existence right now
```