***Delete node from singly linked list***

#include <stdio.h>

#include <stdlib.h>

struct node{

int val;

struct node \*next;

};

void print\_list(struct node \*head)

{

printf("H->");

while(head)

{

printf("%d->", head->val);

head = head->next;

}

printf("|||\n");

}

void insert\_front(struct node \*\*head, int value)

{

struct node \* new\_node = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if (new\_node == NULL)

{

printf("Failed to insert element. Out of memory");

}

new\_node->val = value;

new\_node->next = \*head;

\*head = new\_node;

}

void delete\_node(struct node \*\*head, int val)

{

struct node \*tmp = NULL;

struct node \*tmp1 = NULL;

if (head == NULL || \*head == NULL) return;

if((\*head)->val == val)

{

/\*Delete the head node\*/

tmp = \*head;

\*head = (\*head)->next;

free(tmp);

return;

}

tmp = \*head;

while(tmp->next && tmp->next->val != val) tmp = tmp->next;

if(tmp->next)

{

tmp1 = tmp->next;

tmp->next = tmp1->next;

free(tmp1);

}

else

{

printf("%d not found in the list.\n", val);

}

}

void main()

{

int count = 0, i, val;

struct node \* head = NULL;

printf("Enter number of elements: ");

scanf("%d", &count);

for (i = 0; i < count; i++)

{

printf("Enter %dth element: ", i);

scanf("%d", &val);

insert\_front(&head, val);

}

printf("Initial List: ");

print\_list(head);

printf("Enter a value to delete from the list: ");

scanf("%d", &val);

delete\_node(&head, val);

printf("List after deletion: ");

print\_list(head);

}

***Output***

Enter number of elements: 3

Enter 0th element: 1

Enter 1th element: 2

Enter 2th element: 3

Initial List: H->3->2->1->|||

Enter a value to delete from the list: 3

List after deletion: H->2->1->|||