```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node*next;
};
struct Node*insert_at_beginning(struct Node*head, int data) {
    struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->data=data;
    newNode->next=head;
    head=newNode;
    return head;
}

struct Node*insert_at_end(struct Node*head, int data) {
    struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
    newNode->data=data;
    newNode->next=NULL;

    if (head == NULL) {
        return newNode;
    }
    struct Node*temp=head;
    while (temp->next!=NULL) {
        temp = temp->next;
    }
    temp->next=newNode;

    return head;
}

struct Node*delete_node(struct Node*head, int data) {
    struct Node*temp=head;
    struct Node*prev=NULL;

    if (head == NULL) {
        printf("List is empty\n");
        return head;
    }

    if(temp!=NULL&&temp->data==data) {
        head=temp->next;
        free(temp);
        return head;
    }

    while(temp!=NULL&&temp->data !=data) {
        prev=temp;
        temp=temp->next;
    }

    if (head == NULL) {
        printf("Element not found\n");
        return head;
    }
prev->next=temp->next;
  free(temp);
  return head;
}

void traverse(struct Node* head) {
    struct Node* temp = head;

    if (head == NULL) {
        printf("List is empty\n");
        return;
    }

    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

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

    head = insert_at_beginning(head, 10);
    head = insert_at_beginning(head, 5);
    head = insert_at_end(head, 20);
    head = insert_at_end(head, 30);

    printf("Linked List:\n");
    traverse(head);

    head = delete_node(head, 20);
    printf("After deleting 20:\n");
    traverse(head);

    return 0;
}
```