main.c

Share Run

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node *next;
} Node;
void insert(Node **tail, int val) {
    Node *n = malloc(sizeof(Node));
    n->data = val;
    if (!*tail) {
        n->next = n;
        *tail = n;
    } else {
        n->next = (*tail)->next;
        (*tail)->next = n;
        *tail = n;
    }
}
void delete(Node **tail, int val) {
    if (!*tail) return;
    Node *curr = (*tail)->next, *prev = *tail;
    do {
        if (curr->data == val) {
            if (curr == *tail && curr->next == *tail) {
                free(curr);
                *tail = NULL;
```

Output Clear

```
1.Insert  2.Display  3.Delete  4.Exit
Choice: 2
List is empty.

1.Insert  2.Display  3.Delete  4.Exit
Choice: 1
Enter value: 10

1.Insert  2.Display  3.Delete  4.Exit
Choice:
```

main.c

Share | Run

```c
            return;
        }
        if (curr == *tail) *tail = prev;
        prev->next = curr->next;
        free(curr);
        return;
    }
    prev = curr;
    curr = curr->next;
  } while (curr != (*tail)->next);
}
void display(Node *tail) {
  if (!tail) {
    printf("List is empty.\n");
    return;
  }
  Node *p = tail->next;
  printf("List: ");
  do {
    printf("%d -> ", p->data);
    p = p->next;
  } while (p != tail->next);
  printf("(back to head)\n");
}
int main() {
  Node *tail = NULL;
```

Output | Clear

```
1.Insert  2.Display  3.Delete  4.Exit
Choice: 2
List is empty.

1.Insert  2.Display  3.Delete  4.Exit
Choice: 1
Enter value: 10

1.Insert  2.Display  3.Delete  4.Exit
Choice:
```

main.c

Share | Run

```
    do {
        printf("%d -> ", p->data);
        p = p->next;
    } while (p != tail->next);
    printf("(back to head)\n");
}
int main() {
    Node *tail = NULL;
    int ch, val;
    while (1) {
        printf("\n1.Insert  2.Display  3.Delete  4.Exit\nChoice: ");
        scanf("%d", &ch);
        if (ch == 1) {
            printf("Enter value: ");
            scanf("%d", &val);
            insert(&tail, val);
        } else if (ch == 2) {
            display(tail);
        } else if (ch == 3) {
            printf("Enter value to delete: ");
            scanf("%d", &val);
            delete(&tail, val);
        } else break;
    }
    return 0;
}
```

Output | Clear

```
1.Insert  2.Display  3.Delete  4.Exit
Choice: 2
List is empty.

1.Insert  2.Display  3.Delete  4.Exit
Choice: 1
Enter value: 10

1.Insert  2.Display  3.Delete  4.Exit
Choice:
```

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node *prev, *next;
} Node;
void insert(Node **h, int v) {
    Node *n = malloc(sizeof(Node)), *t = *h;
    n->data = v; n->next = NULL;
    if (!t) { n->prev = NULL; *h = n; return; }
    while (t->next) t = t->next;
    t->next = n; n->prev = t;
}
void del(Node **h, int v) {
    Node *t = *h;
    while (t && t->data != v) t = t->next;
    if (!t) return;
    if (t->prev) t->prev->next = t->next; else *h = t->next;
    if (t->next) t->next->prev = t->prev;
    free(t);
}
void show(Node *t) {
    printf("List: ");
    while (t) { printf("%d <-> ", t->data); t = t->next; }
    printf("NULL\n");
}
```

Output | Clear

```
1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 10

1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 20

1.Insert 2.Display 3.Delete 4.Exit
Choice:
```

main.c

Share Run

```c
}
void show(Node *t) {
    printf("List: ");
    while (t) { printf("%d <-> ", t->data); t = t->next; }
    printf("NULL\n");
}
int main() {
    Node *head = NULL;
    int ch, val;
    while (1) {
        printf("\n1.Insert 2.Display 3.Delete 4.Exit\nChoice: ");
        scanf("%d", &ch);
        if (ch == 1) {
            printf("Enter value: ");
            scanf("%d", &val);
            insert(&head, val);
        } else if (ch == 2) {
            show(head);
        } else if (ch == 3) {
            printf("Enter value to delete: ");
            scanf("%d", &val);
            del(&head, val);
        } else break;
    }
    return 0;
}
```

Output

Clear

```
1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 10

1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 20

1.Insert 2.Display 3.Delete 4.Exit
Choice:
```

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node *prev, *next;
};
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node
        ));
    newNode->data = value;
    newNode->prev = newNode->next = NULL;
    return newNode;
}
void insertEnd(struct Node** head, int value) {
    struct Node* newNode = createNode(value);
    if (!*head) {
        *head = newNode;
        return;
    }
    struct Node* temp = *head;
    while (temp->next) temp = temp->next;
    temp->next = newNode;
    newNode->prev = temp;
}
void deleteValue(struct Node** head, int value) {
    struct Node* temp = *head;
```

Output | Clear

```
1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 10

1.Insert 2.Display 3.Delete 4.Exit
Choice:
```

main.c

Share | Run

```
    while (temp && temp->data != value) temp = temp->next;
    if (!temp) return;
    if (temp->prev) temp->prev->next = temp->next;
    else *head = temp->next;
    if (temp->next) temp->next->prev = temp->prev;
    free(temp);
}
void display(struct Node* head) {
    while (head) {
        printf("%d <-> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
int main() {
    struct Node* head = NULL;
    int ch, val;
    while (1) {
        printf("\n1.Insert 2.Display 3.Delete 4.Exit\nChoice: ");
        scanf("%d", &ch);
        if (ch == 1) {
            printf("Enter value: ");
            scanf("%d", &val);
            insertEnd(&head, val);
        } else if (ch == 2) {
            display(head);
```

Output | Clear

```
1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 10

1.Insert 2.Display 3.Delete 4.Exit
Choice:
```

main.c

Share Run

```
    while (head) {
        printf("%d <-> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}
int main() {
    struct Node* head = NULL;
    int ch, val;
    while (1) {
        printf("\n1.Insert 2.Display 3.Delete 4.Exit\nChoice: ");
        scanf("%d", &ch);
        if (ch == 1) {
            printf("Enter value: ");
            scanf("%d", &val);
            insertEnd(&head, val);
        } else if (ch == 2) {
            display(head);
        } else if (ch == 3) {
            printf("Enter value to delete: ");
            scanf("%d", &val);
            deleteValue(&head, val);
        } else break;
    }
    return 0;
}
```

Output Clear

```
1.Insert 2.Display 3.Delete 4.Exit
Choice: 1
Enter value: 10

1.Insert 2.Display 3.Delete 4.Exit
Choice:
```