```
Linked List: 2 -> 3 -> 4 -> NULL

--- Singly Linked List Deletion Menu ---
1. Insert at End
2. Delete at Beginning
3. Delete at End
4. Delete by Value
5. Display
6. Exit
Enter your choice: 4
Enter value to delete: 3
Deleted element: 3

--- Singly Linked List Deletion Menu ---
1. Insert at End
2. Delete at Beginning
3. Delete at End
4. Delete by Value
5. Display
6. Exit
Enter your choice: 5
Linked List: 2 -> 4 -> NULL

--- Singly Linked List Deletion Menu ---
1. Insert at End
2. Delete at Beginning
3. Delete at End
4. Delete by Value
5. Display
6. Exit
Enter your choice: 6
Exiting program...
```

```
/*ENROLLMENT :ET24IMCA002
NAME : AREEBA KAGZI
CLASS :IMCA13301 - DATA STRUCTURES
PROBLEM DEFINITION : 17)Write a program to merge two list Singly Linked
*/
#include <stdio.h>
#include <stdlib.h>

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

struct Node* first = NULL;
struct Node* second = NULL;

void insert(struct Node** head, int value) {
    struct Node* newNode = malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    struct Node* temp = *head;
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = newNode;
}
void merge() {
    if (first == NULL) {
        first = second;
        second = NULL;
        printf("Merged.\n");
        return;
    }
    struct Node* temp = first;
    while (temp->next != NULL)
        temp = temp->next;
    temp->next = second;
    second = NULL;
    printf("Merged.\n");
}

void show(struct Node* head) {
    if (head == NULL) {
        printf("Empty.\n");
        return;
    }
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
int main() {
    int ch, value;
    while (1) {
        printf("\n--- Merge Linked Lists ---\n");
        printf("1. Insert into List 1\n");
        printf("2. Insert into List 2\n");
        printf("3. Show List 1\n");
        printf("4. Show List 2\n");
        printf("5. Merge List 2 into List 1\n");
        printf("6. Exit\n");
        printf("Choice: ");
        scanf("%d", &ch);

        if (ch == 1) {
            printf("Enter number: ");
            scanf("%d", &value);
            insert(&first, value);
        }
        else if (ch == 2) {
            printf("Enter number: ");
            scanf("%d", &value);
            insert(&second, value);
        }
        else if (ch == 3) {
            show(first);
        }
        else if (ch == 4) {
            show(second);
        }
        else if (ch == 5) {
            merge();
        }
        else if (ch == 6) {
            printf("Exit.\n");
            break;
        }
        else {
            printf("Invalid.\n");
        }
    }

    return 0;
}
```

OUTPUT

```
--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
```

```
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 1
Enter number: 11

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 1
Enter number: 22

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 2
Enter number: 1

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 2
Enter number: 2

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 3
11 -> 22 -> NULL

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 4
```

```
1 -> 2 -> NULL

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 5
Merged.

--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 3
11 -> 22 -> 1 -> 2 -> NULL
--- Merge Linked Lists ---
1. Insert into List 1
2. Insert into List 2
3. Show List 1
4. Show List 2
5. Merge List 2 into List 1
6. Exit
Choice: 6
Exit.
```

```
/*ENROLLMENT :ET24IMCA002
NAME : AREEBA KAGZI
CLASS :IMCA13301 - DATA STRUCTURES
PROBLEM DEFINITION : 18)Write a program to Sort an element from a Singly Linked
*/
#include <stdio.h>
#include <stdlib.h>

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

struct Node* head = NULL;

void addNode(int value) {
        struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
  newNode->data = value;
  newNode->next = head;
```