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

// Structure to represent a Todo task
struct Task {
    char description[100];
    struct Task* next;
};

// Function to create a new task
struct Task* createTask(char description[])
{
    struct Task* newTask = (struct
Task*)malloc(sizeof(struct Task));
    strcpy(newTask->description,
description);
    newTask->next = NULL;
    return newTask;
}
```

```
// Function to add a task to the list
void addTask(struct Task** head, char
description[]) {
    struct Task* newTask =
createTask(description);
    if (*head == NULL) {
        *head = newTask;
    } else {
        struct Task* temp = *head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newTask;
    }
    printf("Task added: %s\n", description);
}

// Function to display the list of tasks
void displayTasks(struct Task* head) {
    if (head == NULL) {
        printf("No tasks to display!\n");
        return;
    }
    struct Task* temp = head;
    printf("Todo List:\n");
    int taskNo = 1;
    while (temp != NULL) {
        printf("%d. %s\n", taskNo++, temp->description);
        temp = temp->next;
    }
}

// Function to delete a task by its number
void deleteTask(struct Task** head, int taskNo) {
    if (*head == NULL) {
        printf("No tasks to delete!\n");
        return;
    }

    struct Task* temp = *head;
    if (taskNo == 1) {
        *head = temp->next;
        printf("Task deleted: %s\n", temp->description);
        free(temp);
        return;
    }

    for (int i = 1; temp != NULL && i < taskNo - 1; i++) {
        temp = temp->next;
    }

    if (temp == NULL || temp->next == NULL) {
        printf("Invalid task number!\n");
        return;
    }

    struct Task* nextTask = temp->next->next;
    printf("Task deleted: %s\n", temp->next->description);
    free(temp->next);
    temp->next = nextTask;
}

// Main function
int main() {
    struct Task* head = NULL;
    int choice, taskNo;
    char description[100];

    while (1) {
        printf("\n1. Add Task\n");
        printf("2. Display Tasks\n");
        printf("3. Delete Task\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        getchar();  // To consume the newline character
        switch (choice) {
            case 1:
                printf("Enter task description: ");
                fgets(description, 100, stdin);
                description[strcspn(description, "\n")] = 0; // Remove newline character
                addTask(&head, description);
                break;

            case 2:
                displayTasks(head);
                break;

            case 3:
                printf("Enter task number to delete: ");
                scanf("%d", &taskNo);
                deleteTask(&head, taskNo);
                break;
            case 4:
                printf("Exiting program.\n");
                exit(0);

            default:
                printf("Invalid choice! Try again.
\n");
        }
    }
    return 0;
}
```