TO DO LIST

A C-PROGRAMMING PROJECT

SUBMITTED BY:

Akarsh khare

590024681

Batch - 61

SUBMITTED TO :

Dr. Srinivasan Ramachandran

Associate Professor
School Of Computer Science

ABSTRACT

This project presents a **To-Do List Management System** developed in the **C programming language**. The system provides basic functionalities such as adding tasks, viewing the list of tasks, and deleting completed or unwanted tasks. It is implemented using structured programming concepts including **arrays, structures, pointers, and functions** to manage task data efficiently. The application follows a **menu-driven approach**, allowing users to interact with the system easily through the console. This project demonstrates how simple C programs can be used to simulate real-world task management applications and helps in understanding fundamental programming concepts while providing an effective solution for organizing daily activities.

PROBLEM DEFINITON

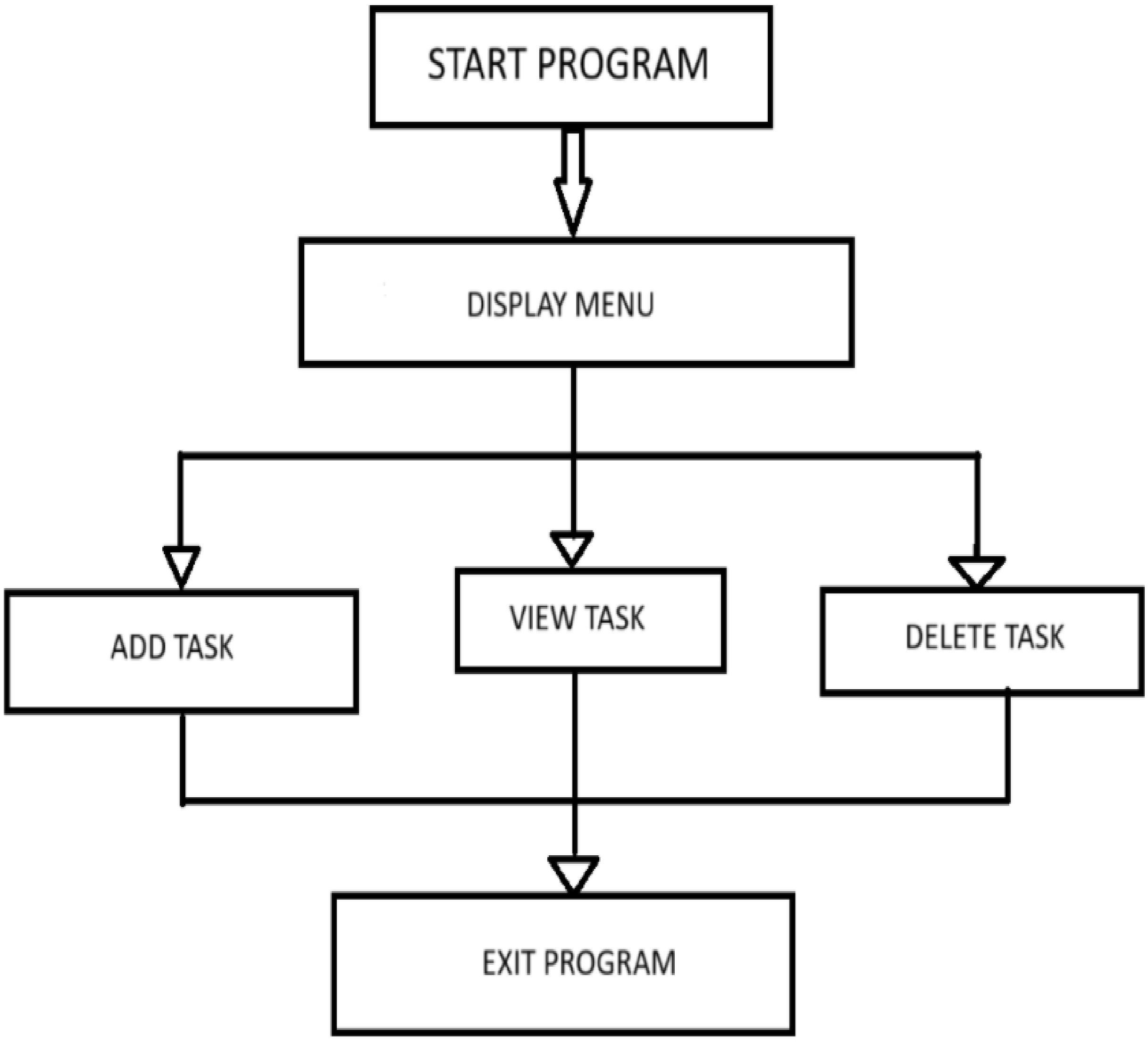
In daily life, individuals often face difficulties in organizing tasks, remembering deadlines, and managing multiple activities efficiently. Traditional methods such as writing tasks on paper or relying on memory are prone to errors, loss of information, and inefficiency. Without a proper system, tasks may be forgotten or improperly prioritized.

The problem is to design a **computerized To-Do List Management System** that:

- Allows users to **add tasks** with ease.
- Displays the **list of stored tasks** in an organized manner.
- Enables users to **delete completed or unwanted tasks**.

The goal is to create a **simple, menu-driven program** using the **C programming language** that users **organize, track, and manage their daily tasks efficiently**, reducing the chances of forgetting important activities and improving overall productivity through a simple To-Do List application.

SYSTEM DESIGN FLOWCHART

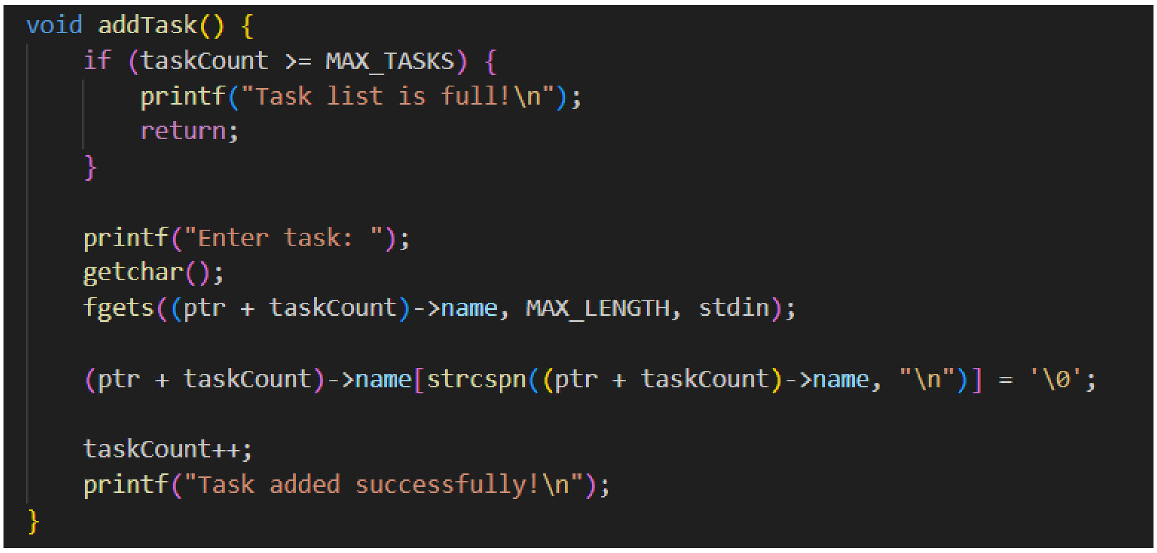


ALGORITHM

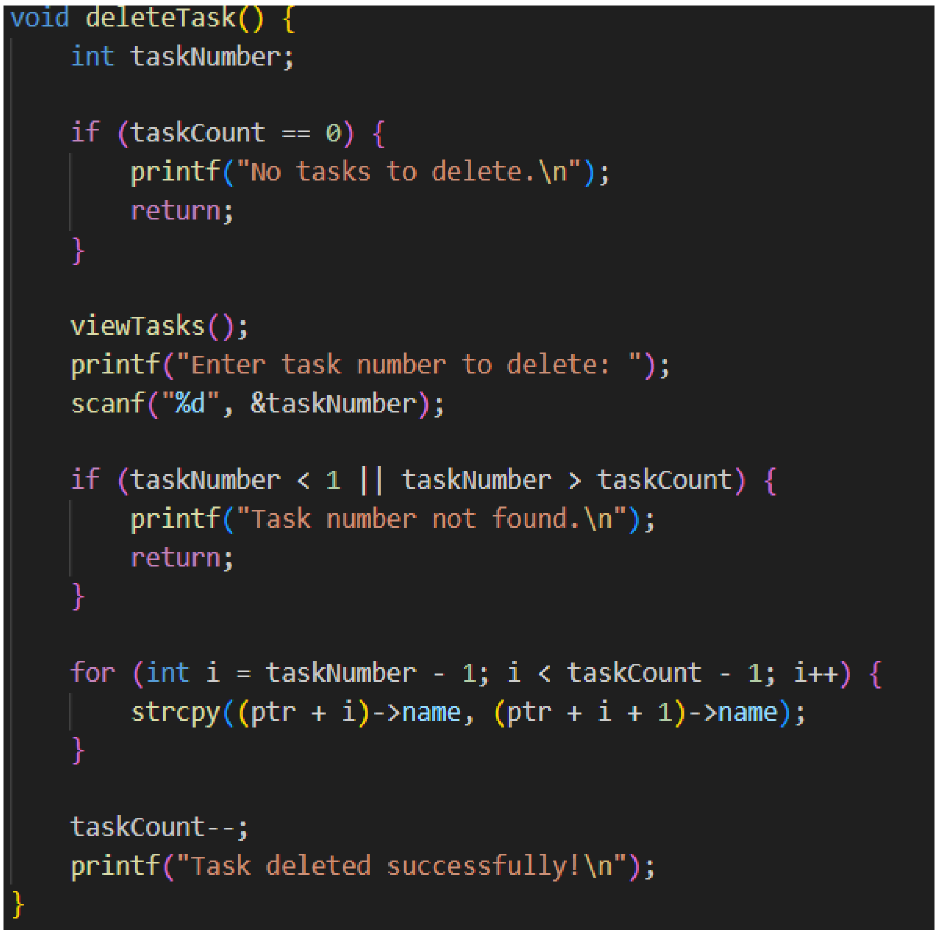
1. Input user’s choice.
2. If the choice is Add Task, add a new task to the list.
3. If the choice is View Tasks, display all tasks.
4. If the choice is Delete Task, remove the selected task.

IMPLEMENTATION WITH SNIPPETS

1.Add and Delete Task

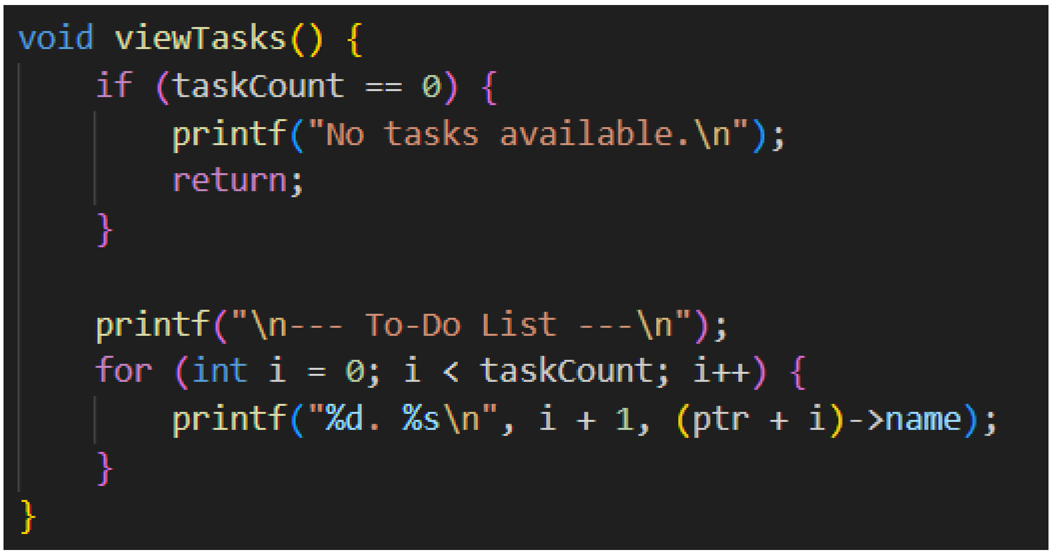


```c
void addTask() {
    if (taskCount >= MAX_TASKS) {
        printf("Task list is full!\n");
        return;
    }

    printf("Enter task: ");
    getchar();
    fgets((ptr + taskCount)->name, MAX_LENGTH, stdin);

    (ptr + taskCount)->name[strcspn((ptr + taskCount)->name, "\n")] = '\0';

    taskCount++;
    printf("Task added successfully!\n");
}
```



```c
void deleteTask() {
    int taskNumber;

    if (taskCount == 0) {
        printf("No tasks to delete.\n");
        return;
    }

    viewTasks();
    printf("Enter task number to delete: ");
    scanf("%d", &taskNumber);

    if (taskNumber < 1 || taskNumber > taskCount) {
        printf("Task number not found.\n");
        return;
    }

    for (int i = taskNumber - 1; i < taskCount - 1; i++) {
        strcpy((ptr + i)->name, (ptr + i + 1)->name);
    }

    taskCount--;
    printf("Task deleted successfully!\n");
}
```

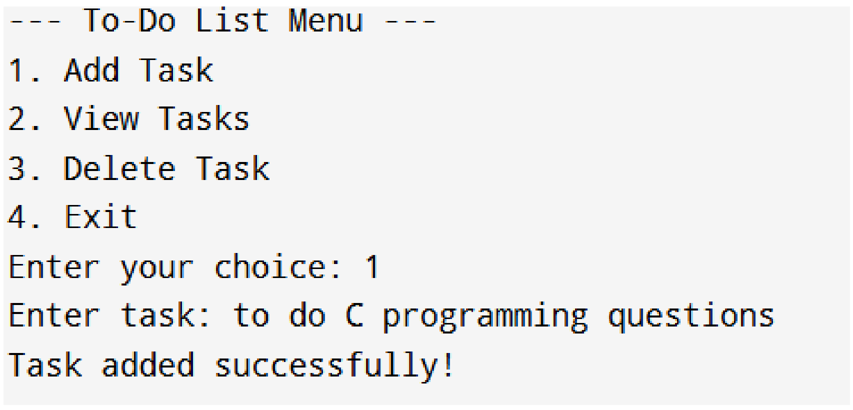
2.View Task



```c
void viewTasks() {
    if (taskCount == 0) {
        printf("No tasks available.\n");
        return;
    }

    printf("\n--- To-Do List ---\n");
    for (int i = 0; i < taskCount; i++) {
        printf("%d. %s\n", i + 1, (ptr + i)->name);
    }
}
```

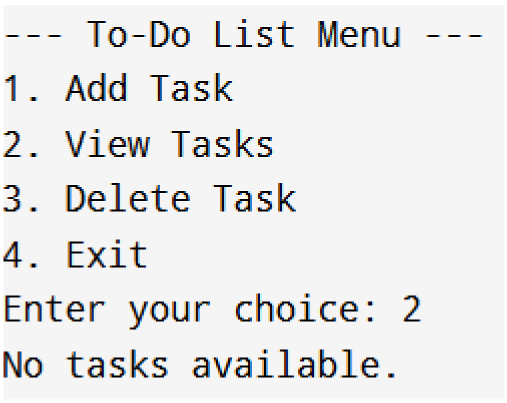
TEST CASES

1. Add Task – Task Added Successfully



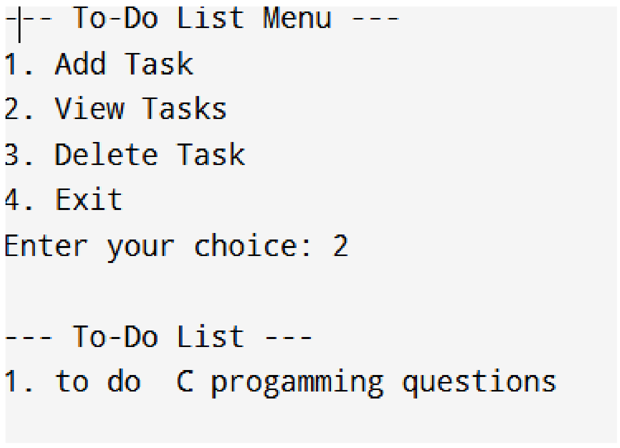
```
--- To-Do List Menu ---
1. Add Task
2. View Tasks
3. Delete Task
4. Exit
Enter your choice: 1
Enter task: to do C programming questions
Task added successfully!
```

2.View Task for not Registered-error message



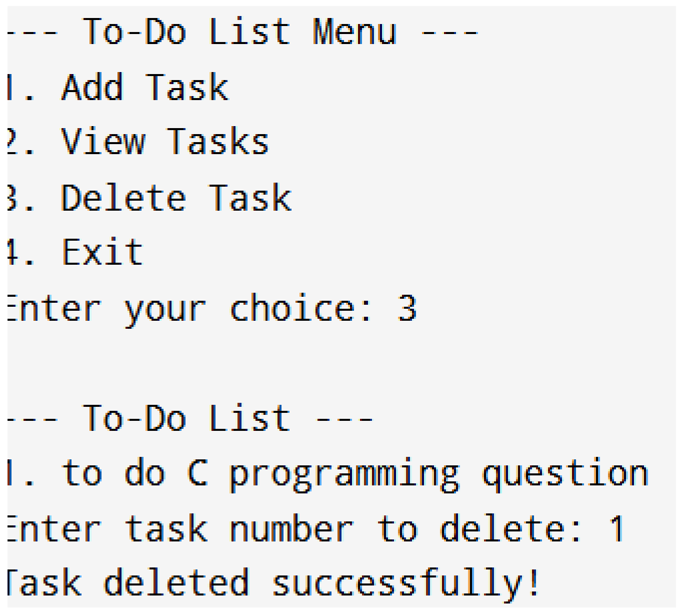
```
--- To-Do List Menu ---
1. Add Task
2. View Tasks
3. Delete Task
4. Exit
Enter your choice: 2
No tasks available.
```

3.View Task for Registered



```
--- To-Do List Menu ---
1. Add Task
2. View Tasks
3. Delete Task
4. Exit
Enter your choice: 2

--- To-Do List ---
1. to do  C progamming questions
```

4.Delete Task



```
--- To-Do List Menu ---
1. Add Task
2. View Tasks
3. Delete Task
4. Exit
Enter your choice: 3

--- To-Do List ---
1. to do C programming question
Enter task number to delete: 1
Task deleted successfully!
```

RESULTS

All test cases executed successfully. The system correctly handled valid and invalid inputs.

CONCLUSION AND FUTURE WORK

CONCLUSION

The To Do List Management System demonstrates how C structures and arrays can be used to manage real-world data. It provides a simple interface for adding tasks, deleting tasks, and viewing tasks.

FUTURE WORK

- Add file handling to store tasks permanently.
- Implement user login system for multiple users.
- Allow setting due dates and priority levels for tasks.
- Add option to edit and update existing tasks.
- Provide search and filter functionality for tasks.

REFERNCES

- Kernighan, B.W., and Ritchie, D.M. The C Programming Language.
- Tutorials Point- C Programming Guide
- GeeksforGeeks- Structures in C