# INTERACTIVE TASK MANAGER

A C Programming Project

# MEMBER OF GRUP

Muktadirul Islam Zion
221-15-5566

Bocktear Abid Borat
221-15-6006

Shohanur Rahman
221-15-6012

# OBJECTIVE

1.Add tasks with priorities and descriptions.

2.Execute tasks with the highest priority.

3.Display the current task list.

4.Delete tasks based on descriptions.

5.Undo the last deleted task.

6.To showcase the use of data structures (priority queue and linked list) and user input handling in C programming.

# MOTIVATION

**01**

Efficiently manage tasks with varying priorities.

**02**

Provide a user-friendly interface for task management.

**03**

Demonstrate the power of data structures in real-world applications.

**04**

Enable users to recover deleted tasks, reducing the risk of data loss.

# WORK METHODOLOGY

## Priority Queue:

1.Tasks are stored in a priority queue based on their priority.
2.Higher priority tasks are executed first.

## User Input Handling:

1.Interactive menu-driven interface.
2.Input validation for robust user interaction.

# WORK METHODOLOGY

## Linked List:

1.Linked lists are used to manage task queues and history.

## Undo/Redo:

1.Deleted tasks are stored in a history list for potential undo.

## Modularity:

1.Functions for adding, executing, displaying, deleting, and undoing tasks.

# BENEFIT/OUTPUT

**Efficient task management:**
1.Users can add, execute, and delete tasks easily.

**Undo functionality:**
1.Users can undo the last deleted task.

**Priority-based execution:**
1.Higher priority tasks are executed first.

**Learning opportunity:**
1.Demonstrates the use of data structures and interactive programming in C.

**Real-world application:**
1.Applicable in various scenarios requiring task management.

# THANK YOU