Quiz of Data Structure

File format: Class_NIM_NAMA.C (Ex: LA05_2702200000_Dature.C)

Case Study: Task Management System

As a project manager at a software development company, you are responsible for managing tasks and ensuring that projects are completed on time. The development team works on multiple projects simultaneously, and each project consists of various tasks with different priorities and due dates. To efficiently manage these tasks, you decide to implement a task management system using a priority queue. The priority of each task is determined by its due date, with tasks due earlier having higher priority. The task management system should allow you to add new tasks, view the list of tasks sorted by priority, and remove tasks in order of priority.

Design and implement a task management system **using double linked list that adopt priority queue** data structure. Your task management system should include the following functionalities:

1. Add a new task to the priority queue, specifying **the task description, assigned employee, and due date**.
2. View the list of tasks sorted **by priority (due date)**, displaying the task ID, description, assigned employee, due date, and status (completed or pending).
3. Remove and display the highest priority task (task with the earliest due date) from the priority queue.

Ensure that your implementation is efficient and allows for easy management of tasks in a software development environment.

Example:

Sample 1:

Number of tasks: *3*

Task Description: *Design UI for website*
Assign Employee: *Ariel*
Due Date: *2024-04-20*

Task Description: *Implement backend logic*
Assign Employee: *Shan*
Due Date: *2024-04-15*

Task Description: *Test application functionality*
Assign Employee: *Ariel*
Due Date: *2024-04-30*

ID	Task Description	Assigned Employee	Due Date	Status
2	Implement backend logic	Shan	2024-04-15	Pending
1	Design UI for website	Ariel	2024-04-20	Pending
3	Test application functionality	Raihan	2024-04-30	Pending

Is there any task that has reached its due date? *Yes*

How many? *1*

Task details:
ID: 2
Description: Implement backend logic
Assigned Employee: Shan
Due Date: 2024-04-15
Status: Completed

The rest Tasks:

ID	Task Description	Assigned Employee	Due Date	Status
1	Design UI for website	Ariel	2024-04-20	Pending
3	Test application functionality	Raihan	2024-04-30	Pending

Explanation:

1. First, we have total **3** task to do.
2. Automatically, the system will looping for **3** times to accept the data (task description, assigned employee, and due date), the default value of status is **Pending**.
3. After the data filled, the list of data will showed on the screen sorted by due date, ID belongs to the input sequence
4. Then, reply **yes** if there is any
5. Next input **1** as the number of task that reached its due date.
6. Automatically the system show the task details that reached due date, update the status as completed, and remove it from the list.
7. At the and the rest task will appear following the task details.

Sample 2:

Number of tasks: *5*

Task Description: *Design UI for website*
Assign Employee: *Ariel*

Due Date: *2024-04-20*

Task Description: *Implement backend logic*
Assign Employee: *Shan*
Due Date: *2024-04-15*

Task Description: *Test application functionality*
Assign Employee: *Ariel*
Due Date: *2024-04-30*

Task Description: *UAT*
Assign Employee: *Leviathan*
Due Date: *2024-05-11*

Task Description: *IUT*
Assign Employee: *Sky*
Due Date: *2024-05-02*

ID	Task Description	Assigned Employee	Due Date	Status
2	Implement backend logic	Shan	2024-04-15	Pending
1	Design UI for website	Ariel	2024-04-20	Pending
3	Test application functionality	Raihan	2024-04-30	Pending
5	IUT	Sky	2024-05-02	Pending
4	UAT	Leviathan	2024-05-11	Pending

Is there any task that has reached its due date? *Yes*

How many? *2*

Task details:
ID: 2
Description: Implement backend logic
Assigned Employee: Shan
Due Date: 2024-04-15
Status: Completed

Task details:
ID: 1
Description: Design UI for website
Assigned Employee: Ariel

Due Date: 2024-04-20
Status: Completed

The rest Tasks:

ID	Task Description	Assigned Employee	Due Date	Status
3	Test application functionality	Raihan	2024-04-30	Pending
5	IUT	Sky	2024-05-02	Pending
4	UAT	Leviathan	2024-05-11	Pending

Explanation:

1. First, we have total **5** task to do.
2. Automatically, the system will looping for **5** times to accept the data (task description, assigned employee, and due date), the default value of status is **Pending**.
3. After the data filled, the list of data will showed on the screen sorted by due date, ID belongs to the input sequence
4. Then, reply **yes** if there is any
5. Next input **2** as the number of task that reached its due date.
6. Automatically the system show the task details that reached due date, update the status as completed, and remove it from the list.
7. At the and the rest task will appear following the task details.

Note: The *Italic* text on sample means input from keyboard