Assignment 1

NAME

Muhammad Ibrahim Afzal

REG NO

Sp23-bse-060

Submitted To

Mr Muhammad Kamran

Date

24-09-2024

```cpp
#include <iostream>
#include <string>
using namespace std;

// Task node structure
struct Task {
    int taskID;
    string description;
    int priority;
    Task* next = NULL;
};

// Head of the task list
Task* head = NULL;

// Function to add a new task to the list based on priority
void add_task() {
    Task* newTask = new Task;

    cout << "Enter Task ID: ";
    cin >> newTask->taskID;
    cin.ignore();  // to consume the newline character after taskID input
    cout << "Enter Task Description: ";
    getline(cin, newTask->description);
    cout << "Enter Task Priority (higher number means higher priority): ";
    cin >> newTask->priority;

    // Insert the new task in the correct position based on priority
    if (head == NULL || head->priority < newTask->priority) {
        newTask->next = head;
        head = newTask;
```

```
        head = newTask;
    } else {
        Task* current = head;
        while (current->next != NULL && current->next->priority >= newTask->priority) {
            current = current->next;
        }
        newTask->next = current->next;
        current->next = newTask;
    }

    cout << "Task added successfully.\n";
}

// Function to view all tasks
void view_tasks() {
    if (head == NULL) {
        cout << "No tasks in the list.\n";
        return;
    }

    Task* current = head;
    cout << "Tasks in the list (sorted by priority):\n";
    while (current != NULL) {
        cout << "Task ID: " << current->taskID
             << ", Description: " << current->description
             << ", Priority: " << current->priority << "\n";
        current = current->next;
    }
}

// Function to remove the task with the highest priority (first task)
```

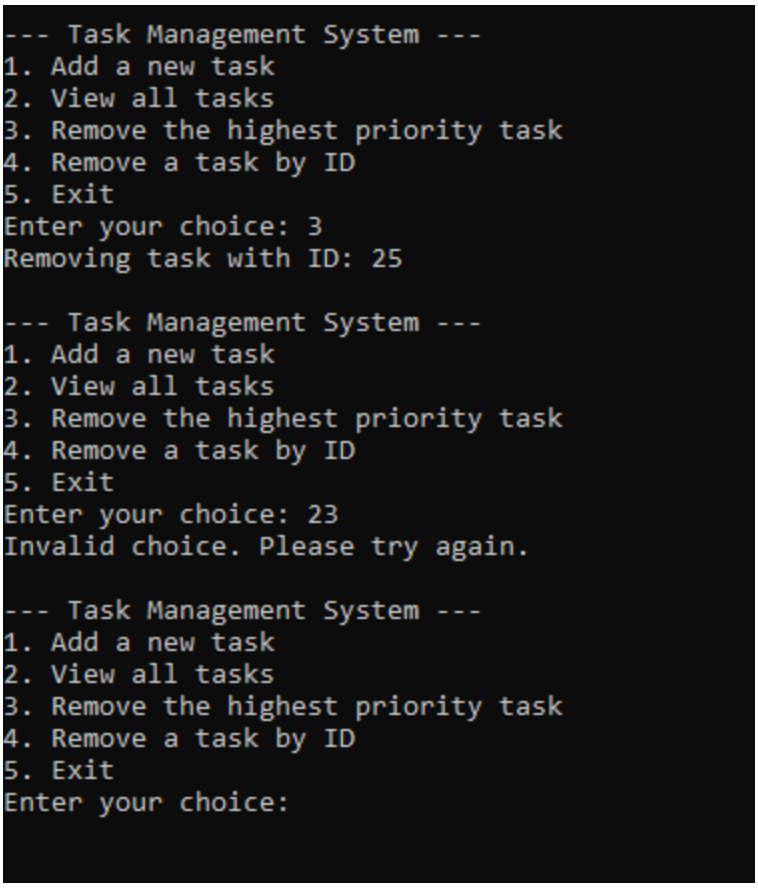
```cpp
void remove_highest_priority_task() {
    if (head == NULL) {
        cout << "No tasks to remove.\n";
        return;
    }

    Task* temp = head;
    head = head->next;
    cout << "Removing task with ID: " << temp->taskID << "\n";
    delete temp;
}

// Function to remove a specific task by its task ID
void remove_task_by_id() {
    if (head == NULL) {
        cout << "No tasks in the list.\n";
        return;
    }

    int taskID;
    cout << "Enter the Task ID to remove: ";
    cin >> taskID;

    if (head->taskID == taskID) {
        Task* temp = head;
        head = head->next;
        delete temp;
        cout << "Task with ID " << taskID << " removed.\n";
        return;
    }
```

```
    Task* current = head;
    Task* prev = NULL;
    while (current != NULL && current->taskID != taskID) {
        prev = current;
        current = current->next;
    }

    if (current == NULL) {
        cout << "Task with ID " << taskID << " not found.\n";
    } else {
        prev->next = current->next;
        delete current;
        cout << "Task with ID " << taskID << " removed.\n";
    }
}

// Main function with the menu
int main() {
    int choice;
    do {
        cout << "\n--- Task Management System ---\n";
        cout << "1. Add a new task\n";
        cout << "2. View all tasks\n";
        cout << "3. Remove the highest priority task\n";
        cout << "4. Remove a task by ID\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
```

```
        switch (choice) {
            case 1:
                add_task();
                break;
            case 2:
                view_tasks();
                break;
            case 3:
                remove_highest_priority_task();
                break;
            case 4:
                remove_task_by_id();
                break;
            case 5:
                cout << "Exiting...\n";
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 5);

    return 0;
}
```

```
--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 1
Enter Task ID: 23
Enter Task Description: Complete Project
Enter Task Priority (higher number means higher priority): 5
Task added successfully.

--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 1
Enter Task ID: 24
Enter Task Description: Write Report
Enter Task Priority (higher number means higher priority): 8
Task added successfully.

--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
```

```
Enter your choice: 1
Enter Task ID: 25
Enter Task Description: Team Meeting
Enter Task Priority (higher number means higher priority): 12
Task added successfully.

--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 2
Tasks in the list (sorted by priority):
Task ID: 25, Description: Team Meeting, Priority: 12
Task ID: 24, Description: Write Report, Priority: 8
Task ID: 23, Description: Complete Project, Priority: 5

--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 23
Invalid choice. Please try again.

--- Task Management System ---
1. Add a new task
2. View all tasks
```


--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 3
Removing task with ID: 25
--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: 23
Invalid choice. Please try again.
--- Task Management System ---
1. Add a new task
2. View all tasks
3. Remove the highest priority task
4. Remove a task by ID
5. Exit
Enter your choice: