**A logo of a university

AI-generated content may be incorrect.**

**SUBMITTED BY: HADIA AJMAL**

**REG NO: FA23-BSE-046**

**COURDE: DS theory**

**CODE:**

#include <iostream>

#include <string>

using namespace std;

// Node structure for a task

struct Task {

int id;

string description;

int priority;

Task\* next;

};

class TaskManager {

private:

Task\* head; // Head of the linked list

public:

TaskManager() { head = nullptr; }

// Function to add a new task based on priority

void addTask(int id, string desc, int priority) {

Task\* newTask = new Task{id, desc, priority, nullptr};

if (!head || head->priority < priority) {

newTask->next = head;

head = newTask;

return;

}

Task\* temp = head;

while (temp->next && temp->next->priority >= priority) {

temp = temp->next;

}

newTask->next = temp->next;

temp->next = newTask;

}

// Function to remove the highest priority task (from the start)

void removeHighestPriorityTask() {

if (!head) {

cout << "No tasks available to remove.\n";

return;

}

Task\* temp = head;

head = head->next;

delete temp;

cout << "Highest priority task removed.\n";

}

// Function to remove a task by its ID

void removeTaskById(int id) {

if (!head) {

cout << "No tasks available to remove.\n";

return;

}

if (head->id == id) {

Task\* temp = head;

head = head->next;

delete temp;

cout << "Task with ID " << id << " removed.\n";

return;

}

Task\* temp = head;

while (temp->next && temp->next->id != id) {

temp = temp->next;

}

if (!temp->next) {

cout << "Task with ID " << id << " not found.\n";

return;

}

Task\* delTask = temp->next;

temp->next = temp->next->next;

delete delTask;

cout << "Task with ID " << id << " removed.\n";

}

// Function to display all tasks

void viewTasks() {

if (!head) {

cout << "No tasks available.\n";

return;

}

Task\* temp = head;

while (temp) {

cout << "ID: " << temp->id << ", Description: " << temp->description << ", Priority: " << temp->priority << "\n";

temp = temp->next;

}

}

~TaskManager() {

while (head) {

Task\* temp = head;

head = head->next;

delete temp;

}

}

};

int main() {

TaskManager manager;

int choice, id, priority;

string desc;

while (true) {

cout << "\nTask Management System\n";

cout << "1. Add Task\n";

cout << "2. View Tasks\n";

cout << "3. Remove Highest Priority Task\n";

cout << "4. Remove Task by ID\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter Task ID: ";

cin >> id;

cin.ignore();

cout << "Enter Task Description: ";

getline(cin, desc);

cout << "Enter Priority (higher number = higher priority): ";

cin >> priority;

manager.addTask(id, desc, priority);

break;

case 2:

manager.viewTasks();

break;

case 3:

manager.removeHighestPriorityTask();

break;

case 4:

cout << "Enter Task ID to remove: ";

cin >> id;

manager.removeTaskById(id);

break;

case 5:

cout << "Exiting program...\n";

return 0;

default:

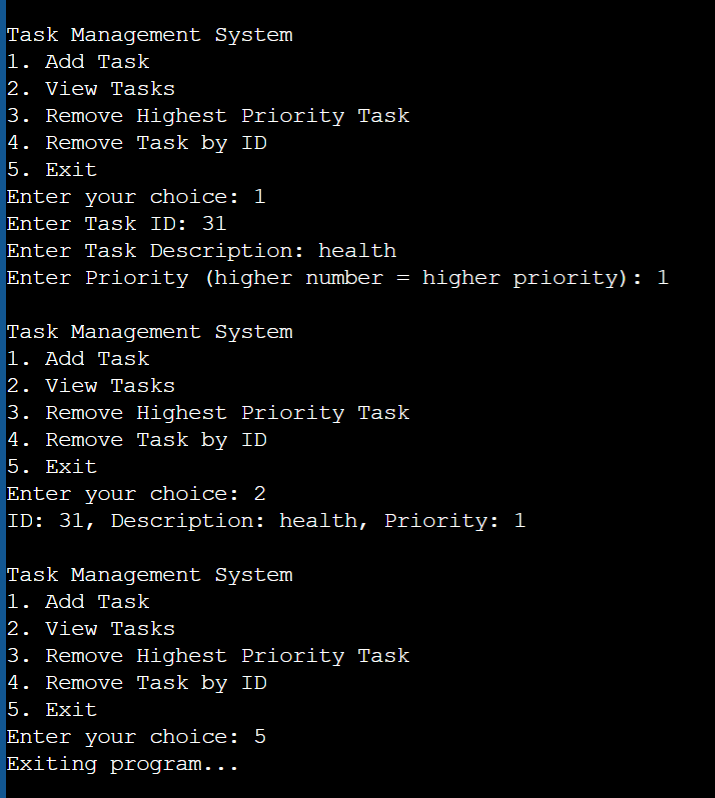
cout << "Invalid choice, try again.\n";

}

}

}

**OUTPUT:**

****