OPEN-ENDED PROBLEM

CE0417 | DATA STRUCTURES & ALGORITHMS

PROBLEM STATEMENT:

To-Do List Application

"You will create a command-line To-Do List application that lets users track their tasks. The application will use a linked list data structure to store the tasks, and will allow the user to add, view, and remove tasks from the list."

CODE:

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

// Define the Node structure for each task
struct Node {
    string task;
    Node* next;
};

// Define the TodoList class
class TodoList {
private:
    Node* head;

public:
    // Constructor to initialize an empty list
    TodoList() {
```

```
    head = nullptr;
  }

  // Destructor to free the memory used by the list
  ~TodoList() {
    Node* current = head;
    while (current != nullptr) {
      Node* temp = current;
      current = current->next;
      delete temp;
    }
  }

  // Function to add a new task to the list
  void addTask(string description) {
    Node* newTask = new Node;
    newTask->task = description;
    newTask->next = head;
    head = newTask;
    cout << "Task added: " << description << endl;
  }

  // Function to remove a task from the list
  void removeTask(string description) {
    Node* current = head;
    Node* prev = nullptr;
    while (current != nullptr && current->task != description) {
      prev = current;
      current = current->next;
    }
    if (current == nullptr) {
```

```
            cout << "Error: task not found." << endl;
        }
        else {
            if (prev == nullptr) {
                head = current->next;
            }
            else {
                prev->next = current->next;
            }
            cout << "Task removed: " << current->task << endl;
            delete current;
        }
    }

    // Function to print all tasks in the list
    void printTasks() {
        if (head == nullptr) {
            cout << "The list is empty." << endl;
        }
        else {
            Node* current = head;
            cout << "Tasks:" << endl;
            while (current != nullptr) {
                cout << "- " << current->task << endl;
                current = current->next;
            }
        }
    }
};

// Main program to test the TodoList class
```

```
int main() {
    TodoList todoList;

    while (true) {
        // Print menu options
        cout << "Menu:" << endl;
        cout << "1. Add a task" << endl;
        cout << "2. Remove a task" << endl;
        cout << "3. View all tasks" << endl;
        cout << "4. Exit" << endl;

        // Get user input
        int choice;
        cout << "Enter your choice (1-4): ";
        cin >> choice;

        // Process user choice
        if (choice == 1) {
            string description;
            cout << "Enter task description: ";
            cin.ignore();  // Ignore any extra newline characters in the input buffer
            getline(cin, description);
            todoList.addTask(description);
        }
        else if (choice == 2) {
            string description;
            cout << "Enter task description: ";
            cin.ignore();  // Ignore any extra newline characters in the input buffer
            getline(cin, description);
            todoList.removeTask(description);
        }
```

```
        else if (choice == 3) {
            todoList.printTasks();
        }
        else if (choice == 4) {
            cout << "Exiting program." << endl;
            break;
        }
        else {
            cout << "Invalid choice. Please enter a number from 1 to 4." << endl;
        }

        cout << endl;
    }

    return 0;
}
```

OUTPUT Screenshots:

Output Clear

```
/tmp/pDDsc8pTQq.o
Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4): 1
Enter task description: DSA OEP submission
Task added: DSA OEP submission

Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4): 1
Enter task description: OS viva
Task added: OS viva

Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4): 3
Tasks:
- OS viva
- DSA OEP submission

Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4): 2
Enter task description: OS viva
Task removed: OS viva

Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4): 3
Tasks:
- DSA OEP submission

Menu:
1. Add a task
2. Remove a task
3. View all tasks
4. Exit
Enter your choice (1-4):
```