#include <iostream>

using namespace std;

struct Node {

    string partyName;

    Node\* next;

};

class LinkedListQueue {

private:

    Node\* front;

    Node\* rear;

public:

    LinkedListQueue() : front(nullptr), rear(nullptr) {}

    void enqueue(const string& partyName) {

        Node\* newNode = new Node{partyName, nullptr};

        if (rear) {

            rear->next = newNode;

        }

        rear = newNode;

        if (!front) {

            front = rear;

        }

        cout << "Party '" << partyName << "' added to the waitlist." << endl;

    }

    void dequeue() {

        if (front == nullptr) {

            cout << "No parties in the waitlist." << endl;

            return;

        }

        Node\* temp = front;

        cout << "Party '" << front->partyName << "' has been seated." << endl;

        front = front->next;

        if (front == nullptr) {

            rear = nullptr; // If the queue becomes empty

        }

        delete temp; // Free the memory

    }

    void display() {

        if (front == nullptr) {

            cout << "The waitlist is currently empty." << endl;

            return;

        }

        cout << "Current Waitlist:" << endl;

        Node\* temp = front;

        while (temp) {

            cout << "- " << temp->partyName << endl;

            temp = temp->next;

        }

    }

    ~LinkedListQueue() {

        while (front) {

            dequeue(); // Free all nodes

        }

    }

};

int main() {

    LinkedListQueue restaurantWaitlist;

    int choice;

    string partyName;

    do {

        cout << "\nRestaurant Waitlist System\n";

        cout << "1. Add Party to Waitlist\n";

        cout << "2. Seat Party\n";

        cout << "3. Display Waitlist\n";

        cout << "4. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice) {

            case 1:

                cout << "Enter party name: ";

                cin.ignore();

                getline(cin, partyName);

                restaurantWaitlist.enqueue(partyName);

                break;

            case 2:

                restaurantWaitlist.dequeue();

                break;

            case 3:

                restaurantWaitlist.display();

                break;

            case 4:

                cout << "Exiting the system." << endl;

                break;

            default:

                cout << "Invalid choice. Please try again." << endl;

        }

    } while (choice != 4);

    return 0;

}