

**CSE243**

**Data Structures and Problem Solving**

**Fall 2024**

**Submitted by:**

**Omar Tarek Mosaad Kotb - 23P0278**

Table of Contents

[Overview 3](#_Toc185959276)

[Code Analysis 3](#_Toc185959277)

[String Class: 3](#_Toc185959278)

[Vector Template Class: 3](#_Toc185959279)

[UnorderedMap Class: 3](#_Toc185959280)

[Participant Class: 3](#_Toc185959281)

[DebtGraph Class: 4](#_Toc185959282)

[Main Function: 4](#_Toc185959283)

[Example 4](#_Toc185959284)

[Test Cases 5](#_Toc185959285)

[Code 6](#_Toc185959286)

# Overview

This program manages debts between participants and attempts to minimize the number of transactions needed to settle those debts. It consists of several classes to handle strings, vectors, unordered maps, participants, and debt graphs.

# Code Analysis

## String Class:

* Custom string class to manage C-style strings.
* Includes constructors, a destructor, and methods for copying, assignment, comparison, and appending strings.

## Vector Template Class:

* Vector class template to store elements of any type.
* Includes methods to add elements (push\_back), access elements (operator[]), and get the current size.

## UnorderedMap Class:

* Custom unordered map (hash table) to store key-value pairs.
* Includes methods to insert elements, find elements, access elements using the subscript operator (operator[]), and iterate through the elements.

## Participant Class:

* Represents a participant with a name and balance.
* Constructor initializes the name and sets the initial balance to zero.

## DebtGraph Class:

* Manages the participants and their debts.
* Includes methods to add participants (addParticipant), record debts (addDebt), minimize transactions (minimizeTransactions), and display participants (displayParticipants).

## Main Function:

* Asks the user to enter the number of participants and their names.
* Displays the list of participants.
* Asks the user to enter the number of debt records (must be less than or equal to the number of participants).
* Records the debts between participants.
* Calls the minimizeTransactions method to minimize the number of transactions required to settle the debts.

## Example

**Input**:

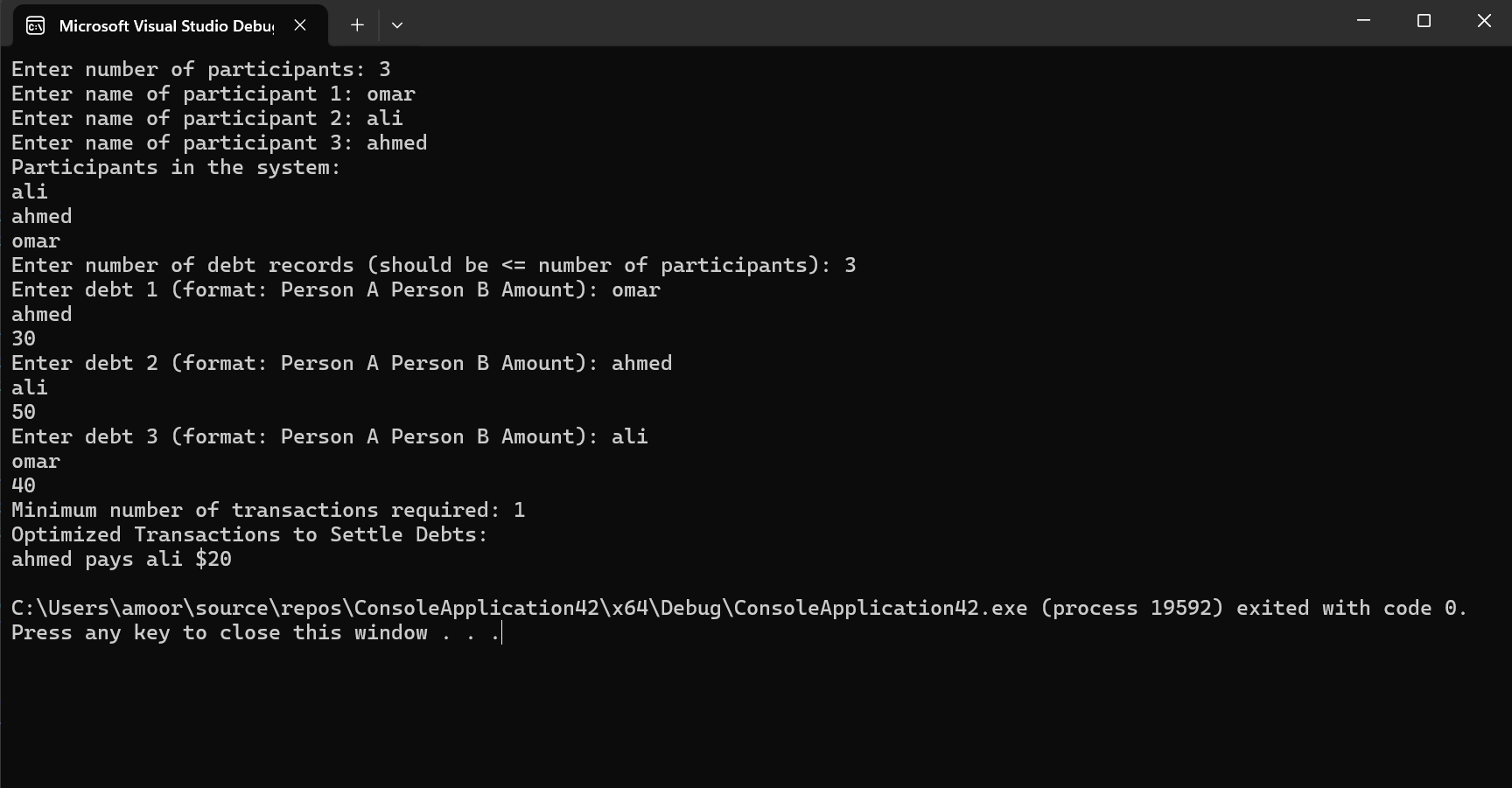
* Number of participants: 3
* Names of participants: Abdulrahman, Omar, Hesham
* Number of debt records: 2
* Debt records: Abdulrahman owes Omar $50, Omar owes Hesham $50

**Output**:

* Minimum number of transactions required: 1
* Optimized Transactions to Settle Debts:
  + Abdulrahman pays Hesham $50

## Test Cases

Code working as intended



Unrecognized Participant

A screenshot of a computer

Description automatically generated

## Code

https://drive.google.com/drive/folders/1KqPZhAerW2kFJFZdrvoFnjrObXIOyeLj