```
//Program 1
//Tax Calculation Program
//Help the user calculate the minimum amount of individual taxes owed for 2024
//Date:01/30/2025

using System;
using System.Collections.Generic;

class TaxCalculator
{
    static void Main()
    {
        List<double> w2Incomes = GetW2Incomes();
        double grossIncome = CalculateGrossIncome(w2Incomes);

        List<double> deductions = GetDeductions();
        double totalDeductions = CalculateTotalDeductions(deductions);

        double adjustedGrossIncome = Math.Max(grossIncome - totalDeductions, 0);

        double taxesOwed = CalculateTaxes(adjustedGrossIncome);

        DisplayResults(grossIncome, totalDeductions, adjustedGrossIncome,
taxesOwed);
    }

    static List<double> GetW2Incomes()
    {
        List<double> incomes = new List<double>();
        Console.WriteLine("Enter the number of W2 incomes:");
        int numIncomes = GetPositiveInteger();

        for (int i = 0; i < numIncomes; i++)
        {
            Console.WriteLine($"Enter W2 income #{i + 1}:");
            double income = GetPositiveDouble();
            incomes.Add(income);
        }

        return incomes;
    }

    static double CalculateGrossIncome(List<double> incomes)
    {
        double grossIncome = 0;
        foreach (double income in incomes)
        {
            grossIncome += income;
        }
        return grossIncome;
    }

    static List<double> GetDeductions()
    {
        List<double> deductions = new List<double>();
        Console.WriteLine("Enter deductions (enter 0 to finish):");

        while (true)
        {
            double deduction = GetNonNegativeDouble();
            if (deduction == 0)
            {
                break;
            }
            deductions.Add(deduction);
        }

        return deductions;
    }

    static double CalculateTotalDeductions(List<double> deductions)
    {
        double totalDeductions = 0;
        foreach (double deduction in deductions)
        {
            totalDeductions += deduction;
        }

        double standardDeduction = 14600;
        return Math.Max(totalDeductions, standardDeduction);
    }

    static double CalculateTaxes(double adjustedGrossIncome)
    {
        if (adjustedGrossIncome <= 0)
        {
            return 0;
        }

        double taxesOwed = 0;

        if (adjustedGrossIncome > 609350)
        {
            taxesOwed += (adjustedGrossIncome - 609350) * 0.37;
            adjustedGrossIncome = 609350;
        }
        if (adjustedGrossIncome > 243725)
        {
            taxesOwed += (adjustedGrossIncome - 243725) * 0.35;
            adjustedGrossIncome = 243725;
        }
        if (adjustedGrossIncome > 191950)
        {
            taxesOwed += (adjustedGrossIncome - 191950) * 0.32;
            adjustedGrossIncome = 191950;
        }
        if (adjustedGrossIncome > 100525)
        {
            taxesOwed += (adjustedGrossIncome - 100525) * 0.24;
            adjustedGrossIncome = 100525;
        }
        if (adjustedGrossIncome > 47150)
        {
            taxesOwed += (adjustedGrossIncome - 47150) * 0.22;
            adjustedGrossIncome = 47150;
        }
        if (adjustedGrossIncome > 11600)
        {
            taxesOwed += (adjustedGrossIncome - 11600) * 0.12;
            adjustedGrossIncome = 11600;
        }
        taxesOwed += Math.Round(adjustedGrossIncome * 0.10, 2);

        return taxesOwed;
    }

    static void DisplayResults(double grossIncome, double totalDeductions, double
adjustedGrossIncome, double taxesOwed)
    {
        Console.WriteLine("\n--- Tax Calculation Results ---");
        Console.WriteLine($"Gross Income: ${grossIncome:F2}");
        Console.WriteLine($"Total Deductions: ${totalDeductions:F2}");
        Console.WriteLine($"Adjusted Gross Income: ${adjustedGrossIncome:F2}");
        Console.WriteLine($"Total Taxes Owed: ${taxesOwed:F2}");
        Console.WriteLine($"Taxes as % of Adjusted Gross Income: {(taxesOwed /
adjustedGrossIncome) * 100:F2}%");
        Console.WriteLine($"Taxes as % of Gross Income: {(taxesOwed / grossIncome)
* 100:F2}%");
    }

    static int GetPositiveInteger()
    {
        int value;
        while (true)
        {
            if (int.TryParse(Console.ReadLine(), out value) && value > 0)
            {
                return value;
            }
            Console.WriteLine("Invalid input. Please enter a positive integer.");
        }
    }

    static double GetPositiveDouble()
    {
        double value;
        while (true)
        {
            if (double.TryParse(Console.ReadLine(), out value) && value > 0)
            {
                return value;
            }
            Console.WriteLine("Invalid input. Please enter a positive number.");
        }
    }

    static double GetNonNegativeDouble()
    {
        double value;
        while (true)
        {
            if (double.TryParse(Console.ReadLine(), out value) && value >= 0)
            {
                return value;
            }
            Console.WriteLine("Invalid input. Please enter a non-negative
number.");
```

```
                }
            }
        }
```