Within 2k limit, no waiver

	2000
Input	No Waiver
Expected	12.2
Output	Base + delivery + tax
Actual	12.2
Output	Base + delivery + tax
Pass/Fail	Pass

Over 2k limit, no waiver

	2002
Input	No Waiver
Expected	20.21
Output	Base + delivery + tax
Actual	20.21
Output	Base + delivery + tax
Pass/Fail	Pass

Within 5k limit, no waiver

	5000
Input	No Waiver
Expected	32.5
Output	Base + delivery + tax
Actual	32.5
Output	Base + delivery + tax
Pass/Fail	Pass

Over 5k limit, no waiver

	5002
Input	No Waiver
Expected	32.51
Output	Base + delivery + tax
Actual	32.51
Output	Base + delivery + tax
Pass/Fail	Pass

Over 8k limit, no waiver

	8002
Input	No Waiver
Expected	62.04
Output	Base + delivery + tax
	62.04

Actual Output	Base + delivery + tax
Pass/Fail	Pass

Over 5k limit, W/ waiver

Input	5002 Waiver
Expected Output	28.01 Base + delivery + tax
Actual Output	28.01 Base + delivery + tax
Pass/Fail	Pass

Over 8k limit, W/ waiver

Input	8002 Waiver
Expected Output	61.01 Base + delivery + tax
Actual Output	61.01 Base + delivery + tax
Pass/Fail	Pass

```
import java.util.Locale;

import java.util.Scanner;


public class proj04 {

   public static void main(String[] args) {

      Locale.setDefault(Locale.US);

      try (Scanner scanner = new Scanner(System.in)) {

         System.out.print("Enter monthly usage in gallons: ");

         if (!scanner.hasNextInt()) {

            System.out.println("Invalid usage. Please provide a non-negative whole number of gallons.");

            return;
```

```
}

int usage = scanner.nextInt();
if (usage < 0) {
   System.out.println("Invalid usage. Usage cannot be negative.");
   return;
}

System.out.print("Is this a low-income household? (y/n): ");
boolean lowIncome = false;
if (scanner.hasNext()) {
   String response = scanner.next().trim().toLowerCase(Locale.US);
   lowIncome = response.startsWith("y");
}

double waterCharge = calculateWaterCharge(usage);
double surcharge = calculateSurcharge(usage);
double tax = lowIncome ? 0.0 : waterCharge * 0.025;
double credit = (lowIncome && usage <= 8000) ? 4.0 : 0.0;
double totalDue = Math.max(0.0, waterCharge + surcharge + tax - credit);

System.out.println();
System.out.printf("Usage:%18d gallons%n", usage);
System.out.printf("Water charge:%11.2f%n", waterCharge);
System.out.printf("Surcharge:%12.2f%n", surcharge);
System.out.printf("Tax:%18.2f%n", tax);
```

```
        if (credit > 0) {
            System.out.printf("Low-income credit:%4.2f%n", credit);
        }
        System.out.printf("Total due:%13.2f%n", totalDue);
    }
}

private static double calculateWaterCharge(int usage) {
    if (usage <= 2000) {
        return 8.0;
    }

    double charge = 8.0;
    int remaining = usage - 2000;

    int midTierGallons = Math.min(remaining, 3000);
    charge += midTierGallons * 0.004;
    remaining -= midTierGallons;

    if (remaining > 0) {
        charge += remaining * 0.007;
    }

    return charge;
}
```

```
    private static double calculateSurcharge(int usage) {
        if (usage <= 2000) {
            return 4.0;
        }
        if (usage <= 8000) {
            return 12.0;
        }
        return 20.0;
    }
}
```