Nifty Assignment

Water Usage Billing System

- Audience: CS1 course. Introduction to Programming

Overview

In this assignment, you will create a Python program to calculate a water bill based on the number of gallons used in a household. The water bill will be determined using a tiered pricing structure, where different prices per gallon apply to different usage levels. The goal is to develop a program that calculates the total water bill for a household based on their water consumption.

Task 1: Basic Water Billing

Python Module: Python Variables

Create a Python script that takes user input for the number of gallons of water used in a household. Calculate and display the total water bill for the household based on the given input.

Test Task 1

Sample Prompt:

```
How many gallons you used this month?
```

Sample Output: Using 0.08 per gallon

```
You Used ## gallons, your bill is $#.##
```

Sample run

```
> python task1.py
How many gallons you used this month? 2
You used 2 gallons, your bill is $0.16
```

Sample test code

```
> pytest test_task1.py
======================================== test session starts
========================================
test_task1.py ....
[100%]
```

```
================================================= 4 passed in 0.01s
=========================================
```

Task 2: Basic Water Billing with minimum charge

Python Module: Basic Conditionals (branching)

Extend your program to include a basic pricing structure for water usage.

- Define a basic billing structure with a flat rate for the first 2,000 gallons and an additional rate for any gallons used beyond 2,000.
- Calculate and display the total water bill for the household based on the given input.

Test Task 2

Sample Prompt:

```
How many gallons you used this month?
```

Sample Output: Using $30.00 for basic rate and $0.08 per gallon for any additional gallons

```
You Used ## gallons, your bill is $#.##
```

Sample run

```
> python task2.py
How many gallons you used this month? 2010
You used 2010 gallons, your bill is $30.80
```

Test code

```
> pytest test_task2.py
========================================= test session starts
=========================================
test_task2.py ....
[100%]


================================================= 4 passed in 0.01s
=========================================
```

Task 3: Tier Water Billing

Python Module: `Intermediate Conditionals (Branching)`

Extend your program to include a tiered pricing structure for water usage. Define multiple tiers, each with its own price per gallon. For example, consider three tiers: 0-2,000 gallons, 2,001-5,000 gallons, and over 5,000 gallons.

- Calculate the water bill by applying the appropriate rate for each tier based on the user's input.
- Display the total water bill, along with a breakdown of charges for each tier.

Test Task 3

Sample Prompt:

```
How many gallons you used this month?
```

Sample Output: Using the following data

- Tier 1: `$30.00` 0-2000 gallons
- Tier 2: `$0.08` per gallon for 2001 to 5000 gallons
- Tier 3: `$0.20` per gallon for more than 5001 gallons

```
You used ## gallons, your bill is $##.##
Tier1 (0-2000): 2000 gallons,  $##.##
Tier2 (2001-5000): 2210 gallons,  $##.##
Tier3 (more than 5000): 0 gallons,  $##.##
```

Sample run

```
> python task3.py
How many gallons you used this month? 2010
You used 2010 gallons, your bill is $30.80
```

Test code

```
> python task3.py
How many gallons you used this month? 5120
You used 5120 gallons, your bill is $294.00
Tier1 (0-2000): 2000 gallons,  $30.00
Tier2 (2001-5000): 3000 gallons,  $240.00
Tier3 (more than 5000): 120 gallons,  $24.00
```

Test code

```
> pytest test_task3.py
======================================= test session starts
==========================================
test_task3.py ....
[100%]


======================================= 6 passed in 0.01s
==========================================
```

Task 4: Total Year Bill

Python Module: Loops

Extend your program to include a tiered pricing structure for water usage, for a year of usage. The twelve-month usage will come from a list of 12 integers.

- Define multiple tiers, each with its own price per gallon. For example, consider three tiers: 0-2,000 gallons, 2,001-5,000 gallons, and over 5,000 gallons.
- Loop over the list and calculate the water bill by applying the appropriate rate for each tier based on each item of the list.
- Display the total water bill for the year, along with a breakdown of charges for each tier.

Test Task 4

Sample Output:

```
You used ## gallons, your bill is $##.##
Tier1 (0-2000): 2000 gallons,  $##.##
Tier2 (2001-5000): 2210 gallons,  $##.##
Tier3 (more than 5000): 0 gallons,  $##.##
```

Sample run with this sample data:

```
year_usage = [800, 900, 1500, 2000, 2050, 3000,
              3500, 4000, 4500, 5020, 1500, 600]
```

```
> python task4.py
You used 29370 gallons, your bill is $2388.00
Tier1 (0-2000): 19300 gallons,  $360.00
Tier2 (2001-5000): 10050 gallons,  $2024.00
Tier3 (more than 5001): 20 gallons,  $4.00
```

Test code

```
> pytest test_task4.py
===================================== test session starts
==========================================
test_task4.py ....
[100%]

========================================= 1 passed in 0.01s
==========================================
```

Task 5: Intermediate Tiered Billing

Python Module: Functions, Refactoring and Error Handling

Refactor your code from Task 4 by allowing customization of the tiered pricing structure. Make sure you include functions

- Modify your program to allow the user to define the pricing tiers and their respective rates through user input.
- Implement error handling to ensure that tiers are defined correctly (e.g., in ascending order) and that rates are valid positive numbers.
- Calculate the water bill using the user-defined tiered pricing structure.
- Display the total water bill and the breakdown of charges for each user-defined tier.

Test Task 5

Sample Output:

```
You used ## gallons, your bill is $##.##
Tier1 (0-2000): 2000 gallons,  $##.##
Tier2 (2001-5000): 2210 gallons,  $##.##
Tier3 (more than 5000): 0 gallons,  $##.##
```

Sample run with this sample data:

```
year_usage = [800, 900, 1500, 2000, 2050, 3000,
              3500, 4000, 4500, 5020, 1500, 600]
```

```
> python task5.py
You used 29370 gallons, your bill is $2388.00
Tier1 (0-2000): 19300 gallons,  $360.00
Tier2 (2001-5000): 10050 gallons,  $2024.00
Tier3 (more than 5001): 20 gallons,  $4.00
```

Test code

```
> pytest test_task5.py
======================================== test session starts
========================================
test_task5.py ....
[100%]


======================================== 1 passed in 0.01s
========================================
```

Note: You can evaluate the students here based on how many functions they refactored their code.

For example, if you run `help()` inside the `Python` prompt, you should get something similar to this:

```
>>> import task5
>>> help(task5)

Help on module task5:

NAME
    task5 - Calculate your water bill based on tier pricing

FUNCTIONS
    calculate_final_water_bill(gallons, charges)
        Calculate the final water bill

        Args:
            gallons (dict): water usage in gallons per tier
            charges (dict): water charges per tier

    print_water_bill(gallons, charges)
        Print the water bill

        Args:
            gallons (dict): water usage in gallons per tier
            charges (dict): water charges per tier

    tier_water_bill(year_usage)
        Calculate the year's water bill based on tier pricing

        Args:
            year_usage (list): monthly water usage in gallons

DATA
    GALLONS_TIER1 = 2000
    GALLONS_TIER2 = 5000
    PRICE_TIER1 = 30.0
    PRICE_TIER2 = 0.08
```

```
    PRICE_TIER3 = 0.2
    charges = {'tier1': 0.0, 'tier2': 0.0, 'tier3': 0.0}
    gallons = {'tier1': 0, 'tier2': 0, 'tier3': 0}
```

Task 6: Advanced Tiered Billing Challenge (optional)

Python Module: `Files`

- Add an optional challenge task that allows the user to specify different rates for peak and off-peak usage hours.
- Implement a time-of-day water usage input system where the user provides usage for both peak and off-peak hours.
- Calculate the water bill by applying the appropriate rates for each time period.
- Display the total water bill, showing charges for peak and off-peak hours separately.

The user should read all the data from a CSV file.

Task 7: Documentation and Testing (optional)

Python Module: `Testing`

- Write clear and concise documentation for your program, including explanations of how to use it and what each function does.
- Test your program with different input scenarios, including edge cases, and ensure it produces accurate results.

Other ideas

- Smart irrigation system. Connect to a weather API and automatically postpone the watering cycle based on the weather.
- Two minute shower challenge
- Xeriscape Yard: Initial investment, calculate how long it will take to ROI.

Student Survey

After completing, you can survey students to gauge their awareness and consciousness about water conservation related to the assignment

1. Before working on the water usage billing assignment, how conscious were you about your personal water consumption habits?

   - ☐ Not conscious at all
   - ☐ Slightly conscious
   - ☐ Moderately conscious
   - ☐ Very conscious

- ☐ Extremely conscious

2. After completing the water usage billing assignment, do you feel more aware of the importance of water conservation?

   - ☐ Yes, I am more aware now.
   - ☐ No, my awareness remains the same.
   - ☐ I'm not sure.

3. Did the assignment change any of your water usage habits at home? If so, please briefly describe how. [Open-ended text response]

4. What do you think could be done to further promote water conservation awareness among students? [Open-ended text response]