



Program to calculate Electricity Bill

A adware

[Read](#)[Discuss](#)

Given an integer **U** denoting the amount of KWh units of electricity consumed, the task is to calculate the electricity bill with the help of the below charges:

- 1 to 100 units – *Rs.10/unit*
- 100 to 200 units – *Rs.15/unit*
- 200 to 300 units – *Rs.20/unit*
- above 300 units – *Rs.25/unit*

Examples:

Input: $U = 250$

Output: 3500

Explanation:

Charge for the first 100 units – $10 * 100 = 1000$

Charge for the 100 to 200 units – $15 * 100 = 1500$

Charge for the 200 to 250 units – $20 * 50 = 1000$

Total Electricity Bill = $1000 + 1500 + 1000 = 3500$

Input: $U = 95$

Output: 950

Explanation:

Charge for the first 100 units – $10 * 95 = 950$

Total Electricity Bill = 950

Recommended: Please try your approach on [*{IDE}*](#) first, before moving on to the solution.

Approach 1: The idea is to identify the charge bar in which it falls and then calculate the bill according to the charges mentioned above. Below is the illustration of the steps:

- Check units consumed is less than equal to the 100, If yes then the total electricity bill will be:

$$\text{Total Electricity Bill} = (\text{units} * 10)$$

-
- Else if, check that units consumed is less than equal to the 200, if yes then total electricity bill will be:

$$\text{Total Electricity Bill} = (100 * 10) + (\text{units} - 100) * 15$$

-
- Else if, check that units consumed is less than equal to the 300, if yes then total electricity bill will be:

$$\text{Total Electricity Bill} = (100 * 10) + (100 * 15) + (\text{units} - 200) * 20$$

-
- Else if, check that units consumed greater than 300, if yes then total electricity bill will be:

$$\text{Total Electricity Bill} = (100 * 10) + (100 * 15) + (100 * 20) + (\text{units} - 300) * 25$$

-

Below is the implementation of the above approach:

C++

```
// C++ implementation to calculate the
// electricity bill
#include<bits/stdc++.h>
using namespace std;

// Function to calculate the
// electricity bill
int calculateBill(int units)
{
    // Condition to find the charges
    // bar in which the units consumed
    // is fall
    if (units <= 100)
    {
        return units * 10;
    }
    else if (units <= 200)
    {
        return (100 * 10) +
            (units - 100) * 15;
    }
    else if (units <= 300)
    {
        return (100 * 10) +
            (100 * 15) +
            (units - 200) * 20;
    }
    else if (units > 300)
    {

```

```

        return (100 * 10) +
               (100 * 15) +
               (100 * 20) +
               (units - 300) * 25;
    }
    return 0;
}

// Driver Code
int main()
{
    int units = 250;
    cout << calculateBill(units);
}

// This code is contributed by spp____

```

Java

```

// Java implementation to calculate the
// electricity bill

import java.util.*;

class ComputeElectricityBill {

    // Function to calculate the
    // electricity bill
    public static int calculateBill(int units)
    {

        // Condition to find the charges
        // bar in which the units consumed
        // is fall
        if (units <= 100) {
            return units * 10;
        }
        else if (units <= 200) {
            return (100 * 10)
                + (units - 100)
                * 15;
        }
        else if (units <= 300) {
            return (100 * 10)
                + (100 * 15)
                + (units - 200)
                * 20;
        }
        else if (units > 300) {
            return (100 * 10)
                + (100 * 15)
                + (100 * 20)
                + (units - 300)
                * 25;
        }
    }
}

```

```

    }
    return 0;
}

// Driver Code
public static void main(String args[])
{
    int units = 250;

    System.out.println(
        calculateBill(units));
}
}

```

Python3

```

# Python3 implementation to calculate the
# electricity bill

```

```

# Function to calculate the
# electricity bill
def calculateBill(units):

```

```

    # Condition to find the charges
    # bar in which the units consumed
    # is fall
    if (units <= 100):

```

```

        return units * 10;

```

```

    elif (units <= 200):

```

```

        return ((100 * 10) +
                (units - 100) * 15);

```

```

    elif (units <= 300):

```

```

        return ((100 * 10) +
                (100 * 15) +
                (units - 200) * 20);

```

```

    elif (units > 300):

```

```

        return ((100 * 10) +
                (100 * 15) +
                (100 * 20) +
                (units - 300) * 25);

```

```

    return 0;

```

```

# Driver Code
units = 250;
print(calculateBill(units));

```

This code is contributed by Code_Mech

C#

```
// C# implementation to calculate the
// electricity bill
using System;

class ComputeElectricityBill{

// Function to calculate the
// electricity bill
public static int calculateBill(int units)
{

    // Condition to find the charges
    // bar in which the units consumed
    // is fall
    if (units <= 100)
    {
        return units * 10;
    }
    else if (units <= 200)
    {
        return (100 * 10) +
            (units - 100) * 15;
    }
    else if (units <= 300)
    {
        return (100 * 10) +
            (100 * 15) +
            (units - 200) * 20;
    }
    else if (units > 300)
    {
        return (100 * 10) +
            (100 * 15) +
            (100 * 20) +
            (units - 300) * 25;
    }
    return 0;
}

// Driver Code
public static void Main(String []args)
{
    int units = 250;

    Console.WriteLine(calculateBill(units));
}
}

// This code is contributed by spp_____
```

Javascript

```
<script>

// Javascript implementation to calculate the
// electricity bill

// Function to calculate the
// electricity bill
function calculateBill(units)
{
    // Condition to find the charges
    // bar in which the units consumed
    // is fall
    if (units <= 100)
    {
        return units * 10;
    }
    else if (units <= 200)
    {
        return (100 * 10)
            + (units - 100)
                * 15;
    }
    else if (units <= 300)
    {
        return (100 * 10)
            + (100 * 15)
            + (units - 200)
                * 20;
    }
    else if (units > 300)
    {
        return (100 * 10)
            + (100 * 15)
            + (100 * 20)
            + (units - 300)
                * 25;
    }
    return 0;
}

// Driver Code
var units = 250;

document.write(calculateBill(units));

// This code is contributed by Khushboogoyal499

</script>
```

Output

3500

Time Complexity: $O(1)$

Auxiliary Space: $O(1)$

Approach 2 : In this approach, we can use an array to store the different rate of charges and their respective range of units. This approach can make the code more readable and easier to maintain. Here's how the code would look like:

C++

```
#include <bits/stdc++.h>
using namespace std;

const int n = 4;

// Function to calculate the electricity bill
int calculateBill(int units)
{
    int charges[n] = { 10, 15, 20, 25 };
    int range[n] = { 100, 100, 100, INT_MAX };
    int bill = 0;

    for (int i = 0; i < n; i++) {
        if (units <= range[i]) {
            bill += charges[i] * units;
            break;
        }
        else {
            bill += charges[i] * range[i];
            units -= range[i];
        }
    }
    return bill;
}

// Driver code
int main()
{
    int units = 250;
    cout << calculateBill(units);
    return 0;
}
```

Java

```
import java.util.*;

public class Main {
```



```

public static final int n = 4;

// Function to calculate the electricity bill
public static int calculateBill(int units) {
    int[] charges = {10, 15, 20, 25};
    int[] range = {100, 100, 100, Integer.MAX_VALUE};
    int bill = 0;

    for (int i = 0; i < n; i++) {
        if (units <= range[i]) {
            bill += charges[i] * units;
            break;
        }
        else {
            bill += charges[i] * range[i];
            units -= range[i];
        }
    }
    return bill;
}

// Driver code
public static void main(String[] args) {
    int units = 250;
    System.out.println(calculateBill(units));
}

// This code is contributed by divyansh2212

```

Python3

```

import sys

MAX_INT = sys.maxsize
n = 4

# Function to calculate the electricity bill

def calculateBill(units):
    charges = [10, 15, 20, 25]
    # changed variable name from 'range' to 'range_'
    range_ = [100, 100, 100, MAX_INT]
    bill = 0

    for i in range(n):
        if units <= range_[i]:
            bill += charges[i] * units
            break
        else:
            bill += charges[i] * range_[i]
            units -= range_[i]

```

```
return bill
```

```
# Driver code
units = 250
print(calculateBill(units))
```

```
# This code is contributed by shivhack999
```

C#

```
using System;

public class Program {
    const int n = 4;

    // Function to calculate the electricity bill
    static int CalculateBill(int units)
    {
        int[] charges = { 10, 15, 20, 25 };
        int[] range = { 100, 100, 100, int.MaxValue };
        int bill = 0;

        for (int i = 0; i < n; i++) {
            if (units <= range[i]) {
                bill += charges[i] * units;
                break;
            }
            else {
                bill += charges[i] * range[i];
                units -= range[i];
            }
        }
        return bill;
    }

    // Driver code
    public static void Main()
    {
        int units = 250;
        Console.WriteLine(CalculateBill(units));
    }
}

// This code is contributed by sarojmc2e
```

Javascript

```
const n = 4;

// Function to calculate the electricity bill
function calculateBill(units) {
    const charges = [10, 15, 20, 25];
```

```
const range = [100, 100, 100, Number.MAX_VALUE];
let bill = 0;

for (let i = 0; i < n; i++) {
  if (units <= range[i]) {
    bill += charges[i] * units;
    break;
  } else {
    bill += charges[i] * range[i];
    units -= range[i];
  }
}
return bill;
}

// Driver code
const units = 250;
console.log(calculateBill(units));
// This code is contributed by shiv1o43g
```

Output

3500

Time Complexity : The time complexity of the `calculateBill` function is $O(n)$, where n is the number of ranges of units and their respective charges. This is because the function uses a **for** loop to iterate through the **range** and **charges** arrays, and for each iteration, it performs a constant amount of work (calculating the bill based on the units consumed).

Since n is a constant value, the time complexity can be considered as $O(1)$ in the best-case scenario. The function takes a constant amount of time to run, regardless of the number of units consumed.

Auxiliary Space : The space complexity of this code is $O(n)$, where n is the number of rate of charges. This is because the program uses two arrays, **charges** and **range**, both of which have a size of n elements. The arrays take up $2 * n * \text{sizeof(int)}$ bytes of memory. In this case, $n = 4$, so the total memory occupied by the arrays is $2 * 4 * \text{sizeof(int)}$.

Last Updated : 04 Apr, 2023

Similar Reads

1. Spring MVC with MySQL - Sample Project For Calculating Electricity Bill
2. Simple Bill Splitter Application using Java Servlets

3. How to Build a Simple Bill Splitter App in Android?

4. Python Program for Program to calculate area of a Tetrahedron

5. Write a program to Calculate Size of a tree | Recursion

6. Efficient program to calculate e^x

7. C program to calculate the value of nPr

8. Program to calculate the value of $\sin(x)$ and $\cos(x)$ using Expansion

9. Program to calculate area of a Tetrahedron

10. Program to calculate area and perimeter of equilateral triangle

Related Tutorials

1. Coding For Kids - Online Free Tutorial to Learn Coding

2. Mathematical and Geometric Algorithms - Data Structure and Algorithm Tutorials

3. Computer Science and Programming For Kids

4. Pandas AI: The Generative AI Python Library

5. OpenAI Python API - Complete Guide

[Previous](#)[Next](#)

Article Contributed By :

A**adware**
adware

Vote for difficulty

Current difficulty : [Basic](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [Code_Mech](#), [spp_____](#), [khushboogoyal499](#), [sushmitamittal1329](#), [divyansh2212](#), [prajwalkandekar123](#), [shivhack999](#), [sarojmcy2e](#), [shiv1o43g](#)

Article Tags : [C Language](#), [C#](#), [C++](#), [DSA](#), [Java](#), [Mathematical](#), [Programming Language](#), [Python](#), [School Programming](#)

Practice Tags : [CPP](#), [Java](#), [Mathematical](#), [python](#)

[Report Issue](#)

A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org



Company

[About Us](#)[Legal](#)[Careers](#)[In Media](#)

Explore

[Job-A-Thon For Freshers](#)[Job-A-Thon For Experienced](#)[GfG Weekly Contest](#)[Offline Classes \(Delhi/NCR\)](#)

[Contact Us](#)[DSA in JAVA/C++](#)[Advertise with us](#)[Master System Design](#)[Master CP](#)

Languages

[Python](#)[Java](#)[C++](#)[PHP](#)[GoLang](#)[SQL](#)[R Language](#)[Android Tutorial](#)

Data Structures

[Array](#)[String](#)[Linked List](#)[Stack](#)[Queue](#)[Tree](#)[Graph](#)

Algorithms

[Sorting](#)[Searching](#)[Greedy](#)[Dynamic Programming](#)[Pattern Searching](#)[Recursion](#)[Backtracking](#)

Web Development

[HTML](#)[CSS](#)[JavaScript](#)[Bootstrap](#)[ReactJS](#)[AngularJS](#)[NodeJS](#)

Computer Science

[GATE CS Notes](#)[Operating Systems](#)[Computer Network](#)[Database Management System](#)[Software Engineering](#)[Digital Logic Design](#)[Engineering Maths](#)

Python

[Python Programming Examples](#)[Django Tutorial](#)[Python Projects](#)[Python Tkinter](#)[OpenCV Python Tutorial](#)[Python Interview Question](#)

Data Science & ML

[Data Science With Python](#)[Data Science For Beginner](#)[Machine Learning Tutorial](#)[Maths For Machine Learning](#)

DevOps

[Git](#)[AWS](#)[Docker](#)[Kubernetes](#)

[Pandas Tutorial](#)[Azure](#)[NumPy Tutorial](#)[GCP](#)[NLP Tutorial](#)[Deep Learning Tutorial](#)

Competitive Programming

[Top DSA for CP](#)[Top 50 Tree Problems](#)[Top 50 Graph Problems](#)[Top 50 Array Problems](#)[Top 50 String Problems](#)[Top 50 DP Problems](#)[Top 15 Websites for CP](#)

Interview Corner

[Company Wise Preparation](#)[Preparation for SDE](#)[Experienced Interviews](#)[Internship Interviews](#)[Competitive Programming](#)[Aptitude Preparation](#)

Commerce

[Accountancy](#)[Business Studies](#)[Economics](#)[Management](#)[Income Tax](#)[Finance](#)

SSC/ BANKING

[SSC CGL Syllabus](#)[SBI PO Syllabus](#)[SBI Clerk Syllabus](#)[IBPS PO Syllabus](#)[IBPS Clerk Syllabus](#)

System Design

[What is System Design](#)[Monolithic and Distributed SD](#)[Scalability in SD](#)[Databases in SD](#)[High Level Design or HLD](#)[Low Level Design or LLD](#)[Top SD Interview Questions](#)

GfG School

[CBSE Notes for Class 8](#)[CBSE Notes for Class 9](#)[CBSE Notes for Class 10](#)[CBSE Notes for Class 11](#)[CBSE Notes for Class 12](#)[English Grammar](#)

UPSC

[Polity Notes](#)[Geography Notes](#)[History Notes](#)[Science and Technology Notes](#)[Economics Notes](#)[Important Topics in Ethics](#)[UPSC Previous Year Papers](#)

Write & Earn

[Write an Article](#)[Improve an Article](#)[Pick Topics to Write](#)[Write Interview Experience](#)[Internships](#)

@geeksforgeeks , Some rights reserved