Allowed Time: 100 Minutes

Instructions: **Total Marks:**

1. Gossips are not allowed.
2. Teacher assistants are for your help, so be nice with them. Respect them as they are teaching you. Raise your hands if you have some problem and need help from TA. Avoid calling them by raising your voice and disturbing the environment of Lab.
3. TA may deduct your marks for any kind of ill-discipline or misconduct from your side.
4. Evaluation will be considered final and you cannot debate for the marks. So, focus on performing the tasks when the time is given to you.

Task 01: **(10 Marks)**

Discount Calculator in C

You are required to write a C program that calculates the final price of a product after applying a discount. The program should take user input for the original price of the product and the discount percentage. It should then calculate and display:

1. The amount of discount applied.
2. The final price after applying the discount.

Requirements:

1. The program should prompt the user to enter the original price of the product. If the user enters a negative value, the program should display an error message and ask the user to enter the price again.
2. The user should then enter the discount percentage. If the percentage is negative or greater than 100, the program should display an error message and prompt the user to enter the value again.
3. The program should calculate the discount amount using the formula:

   Discount Price = ((Original price * Discount %) / 100)

4. The final price should be calculated as: Final price = Original Price – Discount Price
5. The program should display the discount amount and the final price after applying the discount.

Sample:

Enter the original price: -50
Invalid input! The price cannot be negative. Please enter again.
Enter the original price: 500
Enter the discount percentage: 120
Invalid input! Discount percentage must be between 0 and 100. Please enter again.
Enter the discount percentage: 20

Discount Amount: 100.00
Final Price After Discount: 400.00

Code:

```c
#include <stdio.h>

int main() {
    float originalPrice, discountPercentage, discountAmount, finalPrice;

    // Input validation loop for original price
    do {
        printf("Enter the original price: ");
        scanf("%f", &originalPrice);

        if (originalPrice < 0) {
            printf("Invalid input! The price cannot be negative. Please enter again.\n");
        }
    } while (originalPrice < 0);

    // Input validation loop for discount percentage
    do {
        printf("Enter the discount percentage: ");
        scanf("%f", &discountPercentage);

        if (discountPercentage < 0 || discountPercentage > 100) {
            printf("Invalid input! Discount percentage must be between 0 and 100. Please enter
again.\n");
        }
    } while (discountPercentage < 0 || discountPercentage > 100);

    // Calculating discount amount and final price
    discountAmount = (originalPrice * discountPercentage) / 100;
    finalPrice = originalPrice - discountAmount;

    // Displaying results
    printf("\nDiscount Amount: %.2f\n", discountAmount);
    printf("Final Price After Discount: %.2f\n", finalPrice);

    return 0;
}
```

Task 02: (05 Marks)

Counting the Number of Digits in a Number

Write a C program that takes an integer input from the user and counts the number of digits in the given number.

Requirements:

1. The program should prompt the user to enter an integer (positive, negative, or zero).
2. It should count and display the number of digits in the given number.
3. The program should handle negative numbers correctly (e.g., `-123` should be counted as having 3 digits).
4. The program should work for both small and large numbers.
5. If the user enters `0`, the program should correctly count it as **one digit**.

Sample:

Case 1: Positive Number

```
Enter a number: 4567
```

Output:

```
The number of digits in 4567 is: 4
```

Case 2: Negative Number

Input:

```
Enter a number: -98765
```

Output:

```
The number of digits in -98765 is: 5
```

Case 3: Single Digit (Zero)

Input:

```
Enter a number: 0
```

Output:

```
The number of digits in 0 is: 1
```

Code:

```c
#include <stdio.h>

int main() {
    int number, count = 0;

    // Taking input from the user
    printf("Enter a number: ");
    scanf("%d", &number);

    // Handling the special case when number is 0
    if (number == 0) {
        count = 1;
    } else {
        // Convert negative number to positive for counting
        if (number < 0) {
            number = -number;
        }

        // Counting the digits
        while (number > 0) {
            number /= 10; // Remove the last digit
            count++;
        }
    }

    // Displaying the result
```

```
    printf("The number of digits is: %d\n", count);

    return 0;
}
```

Task 03: **(15 Marks)**

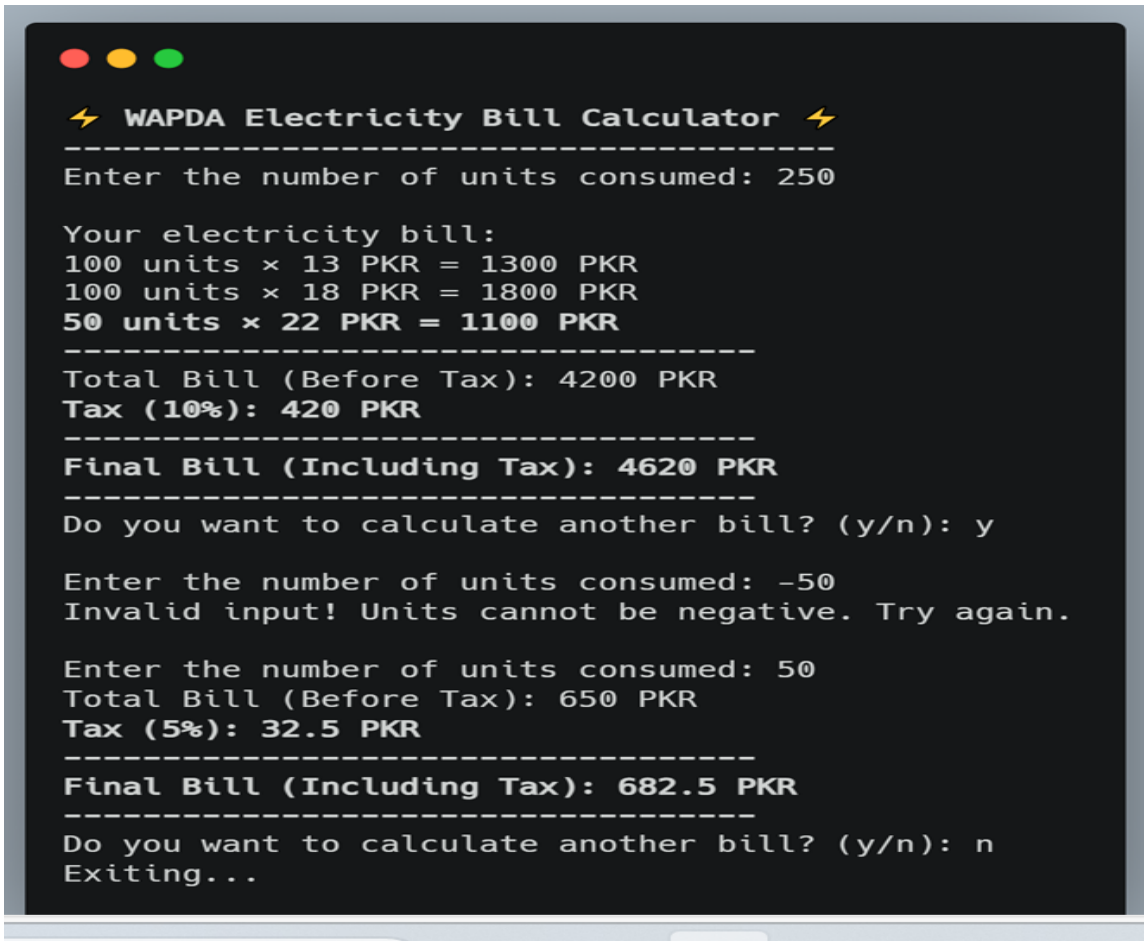
Electricity Bill Calculator for WAPDA

Objective:

You are required to write a C program that calculates the monthly electricity bill for residential consumers in Pakistan based on the number of units consumed. The program should incorporate loops for input validation and multiple calculations, as well as nested if-else conditions to determine billing and tax slabs.

Requirements:

1. **Input Validation:**
   - The program should prompt the user to enter the number of electricity units consumed.
   - If the user enters a negative or zero value, the program should display an error message and prompt them to enter a valid number again.
2. **Billing Calculation:**
   - The electricity bill should be calculated based on the following WAPDA billing slabs:
     - **Up to 100 units:** 13 PKR per unit
     - **101 - 200 units:** 18 PKR per unit
     - **201 - 300 units:** 22 PKR per unit
     - **Above 300 units:** 25 PKR per unit
3. **Tax Calculation:**
   - Once the total bill (before tax) is calculated, apply the tax based on the following slabs:
     - **Bill ≤ 2000 PKR:** 5% tax
     - **2001 - 5000 PKR:** 10% tax
     - **5001 - 10000 PKR:** 15% tax
     - **Above 10000 PKR:** 20% tax
4. **Final Output:**
   - The program should display:
     - **Total bill before tax**
     - **Applied tax amount**
     - **Final bill after tax**
5. **Multiple Calculations:**
   - After displaying the bill, the program should ask the user if they want to calculate another bill.
   - If the user enters ‘Y’ or ‘y’, the program should restart.
   - If the user enters ‘N’ or ‘n’, the program should exit.
   - Any other input should prompt an appropriate message.



Code:

```c
#include <stdio.h>

int main() {
    int units;
    char choice;

    do {
        float bill = 0, tax = 0, finalBill = 0;

        // Input validation loop
        do {
            printf("Enter the number of units consumed: ");
            scanf("%d", &units);

            if (units <= 0) {
                printf("Invalid input! Please enter a positive number of units.\n");
            }
        } while (units <= 0);

        // Billing slab calculation
        if (units <= 100) {
            bill = units * 13;
        } else if (units <= 200) {
            bill = (100 * 13) + ((units - 100) * 18);
```

```
        } else if (units <= 300) {
            bill = (100 * 13) + (100 * 18) + ((units - 200) * 22);
        } else {
            bill = (100 * 13) + (100 * 18) + (100 * 22) + ((units - 300) * 25);
        }

        // Tax slab calculation
        if (bill <= 2000) {
            tax = bill * 0.05;
        } else if (bill <= 5000) {
            tax = bill * 0.10;
        } else if (bill <= 10000) {
            tax = bill * 0.15;
        } else {
            tax = bill * 0.20;
        }

        // Final bill calculation
        finalBill = bill + tax;

        // Displaying results
        printf("\nTotal Bill Before Tax: %.2f PKR\n", bill);
        printf("Applied Tax: %.2f PKR\n", tax);
        printf("Final Bill After Tax: %.2f PKR\n", finalBill);

        // Asking user for another calculation
        printf("\nDo you want to calculate another bill? (Y/N): ");
        scanf(" %c", &choice);

    } while (choice == 'Y' || choice == 'y');

    printf("\nThank you for using the WAPDA Electricity Bill Calculator!\n");

    return 0;
}
```