**Walmart Shopping Budget Simulation - Documentation**

**Objective**

This Python program simulates a shopping experience at Walmart with a fixed budget of $100. It ensures users do not exceed their budget, accounts for tax, and provides a receipt upon checkout.

**Group Member Contributions**

* **Chilawo Munene (G00450721):** Implemented user input handling and validation to ensure proper data entry.
* **Grace Ngoma (G00450740):** Developed budget constraint logic and the calculate\_tax() function to apply tax correctly.
* **Shalom Donga (G00450733):** Created the cart display and final checkout process, including the receipt format.
* **Dingani Freddie Kandiwo (G00450731):** Tested the program for edge cases, fixed bugs, and documented the program logic.

**Features and Functionalities**

1. **User Input Handling**
   * Users can enter item names and prices interactively.
   * Input validation ensures prices are valid and greater than zero.
   * Users can remove items from their cart before completing checkout.
2. **Budget Constraint**
   * Users can add items until the total (including tax) reaches or exceeds $100.
   * If an item exceeds the remaining budget, a warning is displayed.
   * Users can either checkout or continue shopping within their remaining balance.
3. **Tax Calculation**
   * A function calculates tax based on a 10.44% tax rate.
   * Tax is applied to each item before verifying the total budget constraint.
4. **Cart Management**
   * Items and their prices are stored in lists.
   * The program displays the current cart, including:
     + Item names and prices
     + Subtotal before tax
     + Tax amount
     + Total cost after tax
     + Remaining balance
5. **Checkout Process**
   * Users can choose to complete checkout, remove items, or cancel the transaction.
   * A final receipt is displayed upon successful checkout.
6. **Error Handling**
   * Ensures numeric input for prices.
   * Prevents negative values or empty item names.
   * Handles invalid user choices gracefully.

**Functions**

1. **calculate\_tax(price)**
   * Computes tax at a rate of 10.44% for a given item price.
2. **checkout\_process(items, prices, max\_total)**
   * Displays the cart summary.
   * Allows users to proceed to checkout, remove items, or cancel the transaction.
3. **main()**
   * Handles user input for adding items.
   * Manages the budget constraint.
   * Calls checkout\_process() when the user chooses to checkout.

**User Interaction**

* Users enter item names and prices.
* They can review their cart at any time.
* If the budget is exceeded, the program prevents further additions.
* Users can finalize the purchase, remove items, or cancel the transaction.