WALLET

FUNCTION 1: INITIALIZEWALLET()

Write a function named initializedwallet() that doesn’t take any arguments and initializes a wallet with balance of 10000.it returns a dictionary representing the wallet , which should include a balance and an empty transaction credit history list, transaction debit history list and mini statement list.

FUNCTION 2: DISPLAYBALANCE(WALLET)

Write a function named displaybalance() that takes a wallet dictionary as its argument and prints the current balance.

FUNCTION:3 ADDFUNDS(WALLET,AMOUNT)

Write a function named addFunds() that takes a wallet dictionary and an integer amount as its arguments. The function should add the amount to the wallet's balance and record the transaction in the transaction credit history. The mini statement has to print only latest 5 transaction history statements. If the amount is zero or negative the function should return immediately without doing any transaction and prints “Amount can’t be negative” onto the screen.

FUNCTION:4 MAKEPAYMENT(WALLET,AMOUNT)

Write a function named makePayment() that takes a wallet dictionary and an integer amount as its arguments. Ensure there are sufficient funds in the wallet. If so, deduct the amount from the balance and record the transaction. If not, print a warning. The mini statement has to print only latest 5 transaction history statements.

FUNCTION:5 TRANSACTIONHISTORY(WALLET)

Write a function named transactionHistory() that takes a wallet dictionary as its argument. The function should display a list of all transactions made, both credits (additions) and debits (payments).

FUNTIONS:6 USEWALLET ()

Write a function named useWallet() that ties everything together. The following steps should be followed Steps: 1. Setup: Welcome the user and initialize a new digital wallet using initializeWallet(). 2. Menu Loop: Create a menu where the user can choose to: a. View balance b. Add funds c. Make a payment d. View transaction history e. Exit 3. User Input: Depending on the user's choice: a. Display the balance using displayBalance(). b. Add funds using addFunds(). c. Make a payment using makePayment(). d. Display transaction history using transactionHistory(). 4. Continue looping until the user chooses to exit.