REVERSING THE IRREVERSIBLE

OVERVIEW

This C program reads transaction data from a text file processes the transactions using a singly linked list, and identifies and removes fraudulent transactions based on a given list of fraudulent transaction IDs. The program demonstrates the following functionality:

1. TRANSACTION NODE STRUCTURE (STRUCT NODE)

- The program defines a structure called **Node** to represent a transaction node in the linked list.
- Each node contains the following fields:
    - **transactionId**: An integer representing the unique identifier of the transaction.
    - **transactionAmount**: An integer representing the amount of the transaction.
    - **next**: A pointer to the next transaction node in the linked list.

2. FUNCTION TO REVERSE TRANSACTIONS (REVERSETRANSACTIONS)

- The **reverseTransactions** function is responsible for identifying and reversing fraudulent transactions in the linked list.
- It takes the following parameters:
    - **head**: A pointer to a pointer to the head of the linked list.
    - **fraudulentTransactions**: An array of integers representing the IDs of fraudulent transactions.
    - **numberOfFraudulentTransactions**: An integer indicating the number of fraudulent transactions.
- The function iterates through the linked list and checks each transaction's ID against the list of fraudulent transactions.
- If a transaction is found to be fraudulent, it is removed from the linked list by updating the pointers.
- If the fraudulent transaction is the head, the head pointer is updated to the next node.
- If the fraudulent transaction is not the head, the **previous** node's **next** pointer is updated to skip the fraudulent transaction.
- Memory associated with the fraudulent transaction node is freed.
- The function continues to the next node until the end of the list is reached.

3. FUNCTION TO PRINT TRANSACTIONS (PRINTTRANSACTIONS)

- The **printTransactions** function is responsible for printing the transactions that remain after reversing fraudulent ones.
- It takes a pointer to the head of the linked list as a parameter.
- The function iterates through the linked list and prints each transaction's ID and amount.

4. FUNCTION TO FREE LINKED LIST MEMORY (FREELINKEDLIST)

- The **freeLinkedList** function is responsible for freeing the memory associated with the entire linked list.
- It takes the head of the linked list as a parameter.
- The function iteratively traverses the linked list, freeing each node's memory, and updating the **current** and **next** pointers.
- After freeing all nodes, the linked list is completely deallocated.

5. MAIN FUNCTION

- The **main** function is the entry point of the program.
- It opens the text file in read mode to read transaction data.
- If the file opening fails, it displays an error message and terminates with a return code of 1.
- The function reads the number of total transactions and the number of fraudulent transactions from the file.
- It then initializes the head and tail pointers of the linked list to **NULL**.
- The program reads each transaction from the file and creates a new transaction node for it.
- New nodes are added to the end of the linked list, and the tail pointer is updated accordingly.
- The program also reads the IDs of fraudulent transactions into an array.
- The fraudulent transactions are then identified and reversed using the **reverseTransactions** function.
- After reversing fraudulent transactions, the remaining transactions are printed using the **printTransactions** function.
- Memory allocated for the linked list and the fraudulent transactions array is freed.
- The program returns 0, indicating successful execution.