

Digital Wallet

You are supposed to make a digital wallet system that allows people to transfer money among their wallets. The wallet system uses its own currency called FkRupee (F₹). The smallest amount that the users can transfer is F₹ 0.0001. The description of the wallet operations follows

The command **CreateWallet** <accountHolder1> <amount> creates a new wallet with a balance of F₹ <amount> in the name of <accountHolder1>.

The command **TransferMoney** <accountHolder1> <accountHolder2> <amount> would decrease F₹ <amount> from accountHolder1's account and add the same amount in accountHolder2's account.

The command **Statement** <accountHolder1> should display the account statement for accountHolder1's account. The account statement should contain all the transactions made in that account.

The command **Overview** should display the current balance of all the accounts.

Your wallet system also provides some offers to the customers.

Offer 1: When customer A transfers money to customer B and both the account holders have the same balance after the transaction then both the customers get F₹ 10 as a reward.

Offer 2: Whenever the command **Offer2** is fired 3 customers with the highest number of transactions will get F₹ 10, F₹ 5, and F₹ 2 as rewards. If there is a tie (customers having the same number of transactions) then the customer having higher account balance should be given preference. If there is still a tie then the customer whose account was created first should be given preference.

Bonus

Add a command called **FixedDeposit** <accountHolder1> <fd_amount>. Whenever the command is fired an amount equal to <fd_amount> is parked for <accountHolder1>. If for the next 5 transactions the account balance for that <accountHolder1> remains above <fd_amount> then <accountHolder1> gets F₹ 10 as interest. If the account balance goes below <fd_amount> then the FD should be dissolved and the user would need to give the **FixedDeposit** command again to start a new FD.

As an added bonus display the <fd_amount> and remaining transactions in the **Overview** and **Statement** command also.

Note

1. No account can contain balance less than 0.

Sample

(Bold is user input)

```
CreateWallet Harry 100
CreateWallet Ron 95.7
CreateWallet Hermione 104
CreateWallet Albus 200
CreateWallet Draco 500
Overview
Harry 100
Ron 95.7
Hermione 104
Albus 200
Draco 500
TransferMoney Albus Draco 30
TransferMoney Hermione Harry 2
TransferMoney Albus Ron 5
Overview
Harry 112
Ron 100.7
Hermione 112
Albus 165
Draco 530
Statement Harry
Hermione credit 2
Offer1 credit 10
Statement Albus
Draco debit 30
Ron debit 5
Offer2
Overview
Harry 114
Ron 100.7
Hermione 112
Albus 175
Draco 535
```

Notes

1. Code should be demo-able
2. Code should be modular
3. Input can be taken from *command line* or *file* or by *hard coding a **single** string in your code*
4. You are expected to find and handle all corner cases and data validations
5. All data should be stored in memory. Do not use files/databases etc for storage.
6. Attempt the bonus only after everything else is working properly