Lab 9 B

Problem 1: Bank of IIIT-H

Yash wants to open a bank in IIIT-H as SBI is always closed. He wants you to implement the bank management software for the new bank. A bank management software takes in input as commands from the users and processes them. You will read from a file named "file.txt" which contains commands that you have to process. The commands are as follows:

- **Create "name" :** Create an account with name = "name". Initialise the initial account balance to zero
- **Deposit "name" "amt" :** Add amount = "amt" to the account with name = "name"
- **Withdraw "name" "amt" :** Subtract amount = "amt" from the account with name = "name"
- **Transfer "name1" "name2" "amt" :** Transfer amount = "amt" from account with name = "name1" to account with name = "name2"

You also have to implement warnings for the following cases:

- If you are asked to create a new account then you have to print the following : `printf("Account with name %s already exists.\n", name)`
- If any deposit/withdraw is asked to be performed on an account which does not exist print the following : `printf("Account with name %s does not exist.\n", name)`
- If you are asked to do a transer from one account to another and if any of them does not exist then print the following warning : `printf("Account with name %s or %s does not exist.\n", name1, name2)`
- If money to be deducted is more than the account balance then print the following : `printf("Insufficient funds for %s.\n", name)`

All the outputs from your program have to be printed on stdout. If a warnings occur it has to be printed in order of their arrival. After all the commands have been processed, you have to print the account balances of all the accounts int the following format : `name balance`. The order of the accounts should be the same as the order in which they were created. There should be newline character after each account.

Constraints:

- 1<=number of accounts<=100
- 1<=length of name<=100
- 1<=amt<=1e9
- 1<=number of commands<=1000

Example File:

```
Create Dave
Create Charlie
Deposit Dave 36
Transfer Dave Charlie 264
Withdraw Charlie 259
Withdraw Charlie 456
Transfer Charlie Dave 252
Transfer Charlie Dave 191
Withdraw Charlie 294
Transfer Charlie Dave 122
Transfer Dave Charlie 160
Create Alice
Withdraw Dave 380
Deposit Alice 727
```

Example Output:

```
Insufficient funds for Dave.
Insufficient funds for Charlie.
Insufficient funds for Charlie.
Insufficient funds for Charlie.
Insufficient funds for Charlie.
Insufficient funds for Charlie.
Insufficient funds for Charlie.
Insufficient funds for Dave.
Insufficient funds for Dave.
Dave: 36
Charlie: 0
Alice: 727
```

Problem 2: IIIT-H Library Management System

In a library management system, you need to manage students and books available for borrowing. The system should allow the following operations:

- **Add Student:** Adds a student to the library system with a unique ID and name.
- **Add Book:** Adds a book to the library with a unique ID, title, author, and price.
- **Borrow Book:** Allows a student to borrow a book, provided it is available in the library.
- **Return Book:** Allows a student to return a borrowed book to the library.
- **Display Books:** Displays all the books currently available in the library with their details (ID, Title, Author, Available Copies).
- **Display Students:** Displays all students and their borrowed books.

The input, contains a list of operations to be applied to the library system. Each operation will either add a student, add a book, allow a student to borrow or return a book, or display information about the library or students.

Each line in the `students.txt` follows one of the formats below:

- **Add Student:** `1 StudentID StudentName` – Adds a student with the specified ID and name to the library.
- **Add Book:** `2 BookID BookTitle Author Price Quantity` – Adds a book to the library with the specified ID, title, author, and price.
- **Borrow Book:** `5 StudentID Borrow BookID` – Allows a student to borrow a book if available.
- **Return Book:** `6 StudentID Return BookID` – Allows a student to return a borrowed book to the library.
- **Display Books:** `3` – Shows all the books in the library with their details (showing how many available in lib at the moment)
- **Display Students:** `4` – Shows all the students and their borrowed books and number of books they have borrowed

Task:

1. Use structs to represent students and books in the library system.
2. Maintain a record of each student's ID, name, borrowed books, and the library's books with details such as ID, title, author, price, and quantity.

Constraints:

- Each line represents a valid operation.
- Number of Operations ≤ 100.
- Each book's title, author should be treated as strings, while student IDs and book IDs are

integers, while book price is a float.

Input Format:

The `students.txt` contains lines formatted as follows:

```
1 StudentID StudentName
2 BookID BookTitle Author Price
5 StudentID Borrow BookID
6 StudentID Return BookID
3
4
```

Output Format:

After processing all operations, output the details of all books in the library and the books borrowed by students in the following format:

```
BookID BookTitle Author AvailableCopies Price
```

For students, output in the following format:

```
StudentID StudentName BorrowedBooks (List of BookIDs)
```

Structs:

You are required to use structs to model the system. Below are the suggested structures for your implementation:

- **Book Struct:** The `Book` struct should contain the following information:
  - Book ID (int)
  - Book Title (string)
  - Book Author (string)
  - Price (float) 2 decimal places
  - Quantity Available (int)
- **Student Struct:** The `Student` struct should contain the following information:
  - Student ID (int)
  - Student Name (string)
  - List of Borrowed Books (array or dynamic list of Book IDs)

You may also create additional structs as needed to manage the library operations efficiently.

```
1 101 Bob
2 201 HarryPotter JkRowling 10.00 2
2 202 LordOfTheRings JRRTolkien 15.00 2
1 102 Alice
5 101 Borrow 201
5 102 Borrow 202
6 101 Return 201
3
4
```

Example Output:

```
201 HarryPotter JkRowling 2 10.00
202 LordOfTheRings JRRTolkien 1 15.00
101 Bob 0
102 Alice 1
```

Notes:

- For each operation, maintain a record of the books in the library and the students' borrowing accounts.