COP 3223 Recitation: One Dimensional Arrays

Problem: Accounting

Your roommate is a rather savvy businesswoman. She's gotten access to discounts all over town from the various 18 and up clubs to restaurants and more. She's been selling her discounts to students but is having trouble keeping tabs on who owes her money. Luckily, your mother is an accountant and you've learned a thing or two about keeping positive balances. Coupled with your C programming skills, you've given her an offer she can't refuse.

For each student she sells items to, she keeps track of the sale price of each item she's sold to that person as well as the amount of money she's collected from that person. For 20% of her total revenue, you've offered to write a computer program that takes in the value of every transaction, and automatically sends text message reminders to each of the customers that owe your roommate money, once a week.

Your roommate takes you up on the offer, realizing that she actually collects less than 80% of what's due to her.

Write a program that reads in input from a file that stores information about every transaction and prints out the appropriate text messages to the screen.

You may assume that your roommate has no more than 100 customers and each customer is labeled 0 through 99.

Input File Format

The first line of the input file contains a single positive integer, *n (n ≤ 1000),* representing the number of commands to process. The commands follow, one per line.

There are three types of commands:

1 - this is a payment received from a customer

2 - this is an item given to a customer

3 - print out all text messages for negative account balances

Each command line will start with an integer, 1, 2 or 3, corresponding to the types of commands listed above. For command types 1 and 2, this will be followed with a space and a second integer, *ID (0 ≤ ID ≤ 99),* representing the identification number of the customer for the transaction. This will be followed by another space and a positive integer, *v (v ≤ 100),* representing the amount of dollars for the transaction. Commands of type 3 will have no further information following them.

For commands of types 1 and 2, your program should update the appropriate customer account. (A command of type one adds value to the account and a command of type two subtracts value from an account.)

For commands of type 3, you should print out a message with the following format for each customer (in order of customer number) that has a negative account balance:

```
Customer X, you owe $Y. Please pay immediately!
```

where X is the account number and Y is the number of dollars owed (a positive amount).

If all accounts are paid properly, then print out the single line:

```
All accounts are paid up to date!!!
```

Please make sure that your program reads from a file (either account1.txt, account2.txt or account3.txt) and outputs to the screen.

<u>Sample Data and Testing</u>

Please test your program with the posted input files:

```
account1.txt
account2.txt
account3.txt
```

The correct output for each is posted in the files:

```
account1.out
account2.out
account3.out
```