

Programming Assignment #5

Part 1 (data input):

Write a C program that prompts the user to enter some data regarding some clients to a business. The user should prompt for customer account number (a positive integer between 1 and 1000), a last name (a string), and their account balance (a positive floating point number) as shown below. Notice how the user input terminates once the user enters a **-999** as the account number:

Enter account number, last name, and balance.

Enter -999 to end input:

```
? 100 Smith 24.98
? 8000 Jones 334.33
*** Invalid account number. Please enter 1 - 1000 or -999 to exit ***
? 800 Jones 334.33
? 400 Johnson 56.55
? 300 Roberts -330.90
*** Invalid balance amount. Please enter a positive value. ***
? 300 Roberts 330.90
? 500 White 0.00
? -999
```

(What the user types in is shown in **blue** for clarity only.)

You could store this information in either 3 separate arrays or in an array of type struct (something).

Make the array sizes large enough to hold data for up to 5 clients.

Part 2 (data output):

Once the user has completed entering the data, the program should then SORT THE DATA IN ASCENDING ORDER BY ACCOUNT NUMBER, and then output the client data in table form as follows:

ACCOUNT	LAST NAME	BALANCE
100	Smith	24.98
300	Roberts	330.90
400	Johnson	56.55
500	White	0.00
800	Jones	334.33

If the user enters -999 as the first account number, the program should simply exit, without displaying any information such as:

Enter account number, last name, and balance.
Enter -999 to end input:

? -999

Hints:

- If you choose to store the information in arrays, the array for the last name could look like: **char last_name[5][30];** and you would prompt as follows: **scanf ("%s", last_name[x]);** where x is for each client and goes from 0 to 4. When outputting this information, you simply use the **%s** format specifier again as: **printf ("%s", last_name[x]);**
- When prompting for the information, use 3 separate **scanf** statements for each client. That is, although the user will enter the information all on 1 line, your **scanf** statements can look like the following (assume you have chosen to store your data in arrays):

```
scanf ("%i", &client_num[x]);
scanf ("%s", last_name[x]);
scanf ("%f", &balance[x]);
```

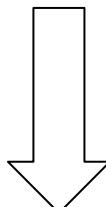
That way, after you read the **client_num**, you can test for the value of -999 before going on to read the **last_name** and **balance**.

Important: Do not wait until the last minute to work on this one. :-)

5 POINTS OF EXTRA CREDIT FOR ASSIGNMENT #5

To earn 5 extra credit point, before displaying results, prompt the user to see if information should be sorted by Account Number, Last Name, or Balance. Based on what the user inputs, sort data accordingly.

Sample program execution (with bonus) on next page



Enter account number, last name, and balance.
Enter -999 to end input:

? 100 Smith 24.98

? 8000 Jones 334.33

*** Invalid account number. Please enter 1 - 1000 or -999 to exit ***

? 800 Jones 334.33

? 400 Johnson 56.55

? 300 Roberts -330.90

*** Invalid balance amount. Please enter a positive value. ***

? 300 Roberts 330.90

? 500 White 0.00

? -999

How would you like information sorted?

1 = Account Number

2 = Last Name

3 = Balance

? 5

*** Invalid number entered. ***

How would you like information sorted?

1 = Account Number

2 = Last Name

3 = Balance

? 3

ACCOUNT	LAST NAME	BALANCE
500	White	0.00
100	Smith	24.98
400	Johnson	56.55
300	Roberts	330.90
800	Jones	334.33