Assignment 3 - Looping, Modularity, Expressions, File Input, File Output

Assignment Specifications

In this assignment, you develop a program to produce a transaction report. Details are given in the specs below.

This program is to be written so that input is from a file (batch input), and the output is text that is printed to a disk file. The program must be written using void functions and the appropriate type of loop structure. You will not receive full credit if you do not use void functions. Do **not** write the program using selection structures (e.g., if/else), as we have not covered these yet, and they are not needed for this program.

You will need to create a data file (for input) on disk (as you did in the previous assignment). Use the sample data found in the specs for this file for your submission run.

Specifications

You are writing this program for the owner of a small business that deals with transporting goods. He wants to utilize the computer for the keeping of records, plus doing a bit of analysis. He is maintaining an operating account and would like to figure out the net balance of money after purchasing goods and selling goods, plus keeping track of various incidental charges. Someone will be entering each of the transactions into a textfile. You will write a C program to read this file and produce a report. This program should read a transaction type that it one of the following:

TAKE IN	cost of a product brought into inventory (negative number)
SHIP OUT	price of a product sold and removed from inventory (positive number)
FEE	Miscellaneous fee paid (negative number)
TARIFF	Tariff charged by receiving country (negative number)
HANDLING	charge to the customer for handling (positive number)

This string is followed by some amount of money (dollars and cents) that belongs to that transaction. The number may be positive or negative; that is determined by the type of transaction to which it is attached, so there is **no** need to determine the sign of the number using selection. Your program should keep track of the running total (net) of these figures. The operating account always starts with a balance of $1000.00 (might change every great once in a while).

Here is some sample input (the string begins in column one of the textfile). It is unknown how many lines of input there are (this easily varies from run to run). All lines are structured similar to these:

```
TAKE IN   -2325.19
SHIP OUT  4735.41
TARIFF    -117.88
TAKE IN   -5100.25
FEE       -510.00
SHIP OUT  7432.44
HANDLING  250
TAKE IN   -500.34
TAKE IN   -3333.44
FEE       -150.00
```

(Note: The data file starts with the first line of data, not with a blank line. The amount always starts on or after column 11.)

Here is some sample output, based on the sample input above. The first line of output is the line with “Turtle Bay Traders”; do not output a blank line before it.

```
Turtle Bay Traders
4800 Palawan Way
Mandalay Beach, CA 99499

Operating Account Starting Balance:  1000.00

Transaction       Amount                Net
-----------       ------                ---
TAKE IN          2325.19           -1325.19
SHIP OUT         4735.41            3410.22
TARIFF            117.88            3292.34
TAKE IN          5100.25           -1807.91
FEE               510.00           -2317.91
SHIP OUT         7432.44            5114.53
HANDLING          250.00            5364.53
TAKE IN           500.34            4864.19
TAKE IN          3333.44            1530.75
FEE               150.00            1380.75

Operating Account Ending Balance:    1380.75
There were 10 transactions processed.
```

(Note: In the actual output, the first line into the output file is the "Turtle Bay Trader" line; there is no blank line in the output file before it..

You may assume that any of the transaction types are no longer than 10 characters in length. (Remember what you have to do for the declaration.) But make sure your program can also handle transaction types that are less than that length Also, note that the number appearing on the report in the Amount column MUST be represented as a positive number, no matter what the actual transaction amount is.

In both the header and the summary, the balance should appear with the decimal point aligned with the decimal points in the Net column in the middle section of the report. The ending balance could be a negative number.

The number of transactions processed should appear in the sentence without any extra spaces before it (in other words, only one space after the word "were", no matter what)..

Make sure that your columnar alignment in the report is EXACTLY what you see above. Don't try to "correct" anything you see above, such as the alignment of the headers.

What You Are Expected To Submit

For this assignment, I want you to develop (but not turn in) a full top-down design including a module structure chart, and a Nassi-Shneiderman chart for each module box.

Then, of course, develop and test the actual program.

Submit the following: source code file (**.c**) and the output file from the test run of your program, using the data shown in the specs. Check your answers carefully!

Your source code filename should be YOUR first initial and last name and the addition **_tr** plus the **.c** extension (such as **kjefferies_tr.c** for me).
Your external filename for the input file **must** be **c:\class\transactions.txt** If you have not yet created a **c:\class** folder for your data and output files, then do so prior to working on this program.
Your external filename for the output file should be **c:\class**\ plus YOUR first initial and last name and the addition **_tr** plus the **.txt** extension (such as **c:\class\kjefferies_tr.txt** for me).

You must use these filenames and filename conventions and the **c:\class** directory to receive full credit.

Do <u>not</u> submit your input file or .exe file.

Be sure to include appropriate comments and indentation in your source code. You will continue to be graded on style.