ED exercise for 2023.03.19

FooBank

How to check your answer

Send answers to ed.fdi.ucm.es/domjudge/team with your group name and password. Send a `.cpp` + any **official** `.h` header files it requires as as an answer.

Statement

You have just been hired by FooBank, a not-very-prestigious financial institution that is so sketchy that they have asked *you* to write a C++ application to handle their accounts.

FooBank's transactions are indicated by the names of their accountholders, and the amount of money to transfer from one account to the other. When money is deposited into the bank, the source will be noted as * (which has infinite money). When money is withdrawn, the target will be likewise be noted as *. If, for any transaction, the source account does not have enough funds, you must output a complaint and not modify either source or target account balances.

Input

Input contains blocks of transactions, which must be processed independently of each other. The first line in a block will be a positive integer, indicating the number of transactions in that block; or 0 to indicate the end of the input. Within each block, each transaction will be in a line of the form `<source> <target> <quantity>`, where source and target will contain only alphabetical, non-whitespace characters; and quantity will be a positive integer.

Output

For each block of input (except the final 0-length block), you must first process its transactions, printing `<transaction-index> <source> is <shortfall> short of <quantity>` whenever a transaction fails because the source has insufficient funds (transaction-index starts at 1 and is incremented for each transaction). After each block, you must output, by alphabetical order of accountholder names, the funds in each account – except for accounts with a balance of 0, which you should skip. After each block, print a line of whitespace.

Sample input and output

2 * Smith 10 Smith * 7 4 * Alice 100 Bob Alice 1 Alice Eve 105 Alice Aardman 2 0	Smith 3 2 Bob is 1 short of 1 3 Alice is 5 short of 105 Aardman 2 Alice 98

IMPORTANT

Once your program works, ask the teacher to verify that it is indeed OK – and to sign attendance. If you do *not* visit the teacher, the teacher will think that you skipped class.

Your code MUST make use of (and therefore #include) the official TreeMap.h and its dependencies. Using the template is strongly recommended.