On-Line Banking

`banking.c`, `banking.C`, `banking.java`, `banking.p`

A central bank is an institution that is responsible for the monetary policy of a country. Beside others, it monitors and supervises other banks. Your task is to write a computer program that will serve as a simple supervision instrument for monitoring bank accounts.

Input Specification

The input contains several scenarios. Each scenario begins with a line containing a single positive integer A, the number of bank accounts at the beginning of the supervision, $0 < A \leq 100$. Then there are A lines each describing one account. Such a line contains the account number, one space, and a non-negative decimal number specifying the account balance.

Each of the following lines specifies a request for one bank operation. The line begins with a command and then, separated by a space, there are optional parameters. The list of commands follows, the third column in the table shows the number of parameters.

Command	Meaning	Pars	Parameters Description
`create`	Create a new account	1	New account number
`deposit`	Deposit cash to some account	2	Account number, Amount to deposit
`withdraw`	Withdraw cash from an account	2	Account number, Amount to withdraw
`transfer`	Wire transfer of money	3	Source account, Target account, Amount

The number of requests in every scenario will always be between 0 and 1000 (inclusive). The last request of a scenario is followed by the word "`end`" and one empty line. Then the next scenario begins. The last scenario is followed by a line containing zero in place of A.

All account numbers consist of exactly four decimal digits followed by a slash character ("`/`") and one digit specifying a code of the bank that operates the account. Each bank has its own unique code.

Amounts are always given as non-negative decimal numbers with exactly two digits after the decimal point. No amount in the input will be higher than 10 000. No unnecessary leading zeros are permitted, i.e., only amounts strictly less than 1.00 may start with a zero.

Output Specification

For each request, output exactly one line. The line must begin with the command followed by the amount parameter (if present) separated from the command by a space. Then there is a colon ("`:`"), one space and the result of the operation.

create: If there is already an account with the same number in the same bank, the result will be "`already exists`". Otherwise, create the account (with the initial balance of zero) and output "`ok`".

For all commands other than *create*, if there is no account with the number specified as a parameter, the result will always be "`no such account`". If the account exists (or both accounts do), the following rules apply.

deposit: The result is always "ok" for existing accounts. Simply add the amount to the account balance.

withdraw: If the balance of the specified account is strictly lower than the requested amount, the result will be "insufficient funds". Otherwise, subtract the amount and output "ok".

transfer: First, if both account numbers are equal (including the bank code), the result should be "same account". Similar to the previous case, if the balance of the source account is lower than the amount to be transferred, the result will be "insufficient funds". Otherwise, transfer the money from the source account to the target one and output either "ok" if the transfer happens inside one bank or "interbank" if the money have to be transferred from one bank to another.

After each scenario, output the word "end" and one empty line. After the "end" of the last scenario, output one additional line containing the word "goodbye".

Sample Input

```
3
1234/5 100.00
4321/6 150.20
5432/5 1600.00
withdraw 1234/5 20.00
deposit 1234/5 35.00
withdraw 1234/5 200.00
transfer 5432/5 1234/5 100.50
transfer 5432/5 4321/6 50.00
create 1234/5
create 1236/5
transfer 1236/5 1236/5 100.00
transfer 1236/5 1234/5 100.00
withdraw 0000/0 100.00
deposit 0000/0 0.11
transfer 1234/5 0000/0 10000.00
end

1
9999/9 9.40
deposit 9999/9 6.92
withdraw 9999/9 9.68
withdraw 9999/9 6.64
end

0
```

Output for Sample Input

```
withdraw 20.00: ok
deposit 35.00: ok
withdraw 200.00: insufficient funds
transfer 100.50: ok
transfer 50.00: interbank
create: already exists
create: ok
transfer 100.00: same account
transfer 100.00: insufficient funds
withdraw 100.00: no such account
deposit 0.11: no such account
transfer 10000.00: no such account
end

deposit 6.92: ok
withdraw 9.68: ok
withdraw 6.64: ok
end

goodbye
```