## Vending machine  2points

You are to design a software driver for a vending machine. The vending machine operator will provide a list of inventory. A list of user inputs (change deposited, buttons pressed) will also be provided.

Items are referenced by a number and a letter (e.g. A4, E8). This will always be a single letter and a single number.

Actions

Buttons and change may be pressed and deposited in any order. Once a valid combination of total change deposited and buttons pressed has been reached, the machine will dispense the corresponding item. The refund button (represented by a # character) will clear any previously pressed buttons. Invalid selections will reset the machine’s recorded button presses; the deposited change will remain.

Example Actions

Assume A3 costs 0.95.

* Actions,A,3,$1 (Dispense A3)
* Actions,A,3,$0.5,$0.5 (Dispense A3)
* Actions,$0.5,A,$0.5,3 (Dispense A3)
* Actions,A,3,$0.5,#,$0.5,$0.5 (Nothing dispensed yet, but there is $1 in the machine)

Invalid Selections

* 2 letters in a row.
* 2 numbers in a row.
* 1 letter and 1 number for a slot that wasn’t in the original inventory.
* 1 letter and 1 number for a slot that is now empty.
* 1 letter or 1 number after a valid selection was obtained.

Change

Change is given from the initial change provided as a part of the inventory list. Change is represented by a $ character followed by a slot number (single digit, 0-9). Change is dispensed using a greedy algorithm. If we are unable to give exact change (the machine ran out of a particular small denomination coin), the next smallest denomination will be returned in its place; in other words, the customer will never be overcharged, but they may end up paying less for the item. It is noted that the customer may also end up paying less if the greedy algorithm fails to yield exact change.

Output

A final inventory must be the output of this program; this will include the final counts of all product.   
This inventory output will be sorted by their item id (“A4”, etc.) in alphabetical order with the change slots listed after the inventory items.   
The total present in the deposit box will also be provided on the last line of the file prefaced by a pair of $ symbols. Total deposited is expressed with two digits after the decimal place. Any cash that was deposited but not refunded or used to purchase anything at the end of all actions will be added to the deposit box total.

Input description/format

The input file consists of a list of inventory to be initially stocked and a series of Actions taken.

The item inventory rows take the form id,name,cost,quantity, where:

* id: letter and a number (upper-case letters)
* name: string representing the item’s name (case sensitive)
* cost: the dollar amount of the item
* quantity: number of the particular item loaded into the machine

A1,MarsBar,0.05,1   
B1,KitKat,0.65,12

The change inventory rows take the form changeslot,value,quantity, where:

* changeslot: $ followed by a 0-indexed single digit (so, there are a max of 10 slots)
* value: numerical value of this change, to 2 digits after the decimal place in all cases
* quantity: number of units of change available

$1,0.05,100   
$2,0.25,200

After the change inventory, the last row will be the actions taken on the machine. This takes the form Actions,...actions.... Action Values can be the following:

* A single letter (case insensitive) or number is considered to be a button press
* Anything following a $ character is deposited change
* The # character represents a refund request.

Actions,$1,A,2,#,B,3,$0.05   
or   
Actions,$0.05,$0.05,$0.05

All input will be formatted correctly, no invalid inputs will be provided.   
Odd series of actions (such as repeated refund requests or dispensing empty slots should, however, be handled gracefully).

Output description/format

The output will be a list of the remaining inventory (items and change) followed by a total of the money present in the deposit box.

The items are listed alphabetically by their item id, followed by the change slots, followed by the deposit box total represented by a $$ symbol.

Item output takes the form id,name,quantity:

* id: Same format as input, uppercase letter and a number
* name: Identical to the input provided name, case sensitive
* quantity: The remaining count of this item

A1,Candy1,0   
B1,Candy7,5

Changeslot output takes the form changeslot,quantity:

* changeslot: $ character and a number, same as the input
* quantity: remaining inventory of change

$1,82   
$2,190

Deposit box total takes the form $$,total:

* total: value of the deposited cash (2 decimal places)

$$,8.00

Example input

A1,Candy1,0.05,1

B1,Candy7,0.65,12

$1,0.05,100

$2,0.25,200

Actions,$1,A,1,$1,B,1,$1,B,1,$1,B,1,$1,B,1,$1,B,1,$1,B,1,$1,B,1,$1,B,1

Example output

A1,Candy1,0

B1,Candy7,4

$1,80

$2,189

$$,9.00

Practice

C4,Candy14,0.7,14

C5,Candy15,0.75,15

D1,Candy16,0.8,16

D2,Candy17,0.85,17

D3,Candy18,0.9,18

D4,Candy19,0.95,19

D5,Candy20,1,20

E1,Candy21,1.05,21

E2,Candy22,1.1,22

E3,Candy23,1.15,23

E4,Candy24,1.2,24

E5,Candy25,1.25,25

F1,Candy26,1.3,26

$0,0.03,10000000

Actions,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,4,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,C,5,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,1,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,2,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,3,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,4,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,D,5,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,1,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,2,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,3,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,4,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,E,5,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10,F,1,$10