2. Change Maker

There is a money bag that contains several denominations of currency in various quantities. Write a program for a change maker that will dispense change using the smallest possible count of available denominations.

Format	**Meaning**
X:Y where X and Y are numbers	Y numbers of X denomination
5:10	10 numbers of 5 rupee notes
100:5,50:5,20:4,10:10,5:10	A comma separated input line that specifies the denominations and the count of those denominations present in the money bag

Version 1

Assumptions

1. There is unlimited counts of all denominations
2. Exact change is available for the given amount for which change is to be dispensed
3. The amount being entered is always a positive integer
4. The count of currency being dispensed need not be OPTIMAL. You can use greedy algo.

Data Format

Input Format
First line contains a comma separated list of numbers denoting the denominations available. Subsequent lines contain the amount for which change needs to be dispensed.

Output Format
First line contains a comma separated list of numbers denoting the denominations available Subsequent lines contain comma separated list of values denoting the denominations being dispensed along with their counts

Example

Input lines
500,100,50,20,10,5,1 23 246 5383

Output lines
500,100,50,20,10,5,1 20:1,1:3 100:2,20:2,5:1,1:1 500:10,100:3,50:1,20:1,10:1,3:1

File naming conventions

change_v1_**Rollno**.c (e.g., change_v1_MT2014001.c) - Roll number should be in upper case

Version 2

Assumptions

1. There are limited counts of denominations available
2. Exact change is may not be available for the given amount. In that case dispense the maximum possible change for the given amount
3. The amount being entered is always a positive integer
4. The count of currency being dispensed need not be OPTIMAL. You can use greedy algo.

Data Format

Input Format
First line contains a comma separated list of numbers denoting the denominations available and their counts. Subsequent lines contain the amount for which change needs to be dispensed.

Output Format
First line contains a comma separated list of numbers denoting the denominations remaining after dispensing. Subsequent lines contain comma separated list of values denoting the denominations being dispensed along with their counts

Example

Input lines
500:5,100:3,50:6,20:8,10:10,5:10,1:100 23 246 5383

Output lines
500:0,100:0,50:0,20:0,10:0,5:0,1:0 20:1,1:3 100:2,20:2,5:1,1:1 500:5,100:1,50:6,20:5,10:10,5:9,1:96

File naming conventions

change_v2_**Rollno**.c (e.g., change_v2_MT2014001.c) - Roll number should be in upper case

Version 3 (Bonus)

In this version, you are to output the OPTIMAL result only (i.e., the one that dispenses the least number of coins/notes).

Assumptions

1. There is unlimited counts of all denominations
2. Exact change is available for the given amount for which change is to be dispensed
3. The amount being entered is always a positive integer
4. Greedy algorithm may not be optimal. Explore dynamic programming.
5. You may try brute force approach that enumerates all possible ways of creating the change and output the one that is optimal.

Data Format

Input Format
First line contains a comma separated list of numbers denoting the denominations available Subsequent lines contain the amount for which change needs to be dispensed.

Output Format
First line contains a comma separated list of numbers denoting the denominations available Subsequent lines contain comma separated list of values denoting the denominations being dispensed along with their counts

Example

Input lines		Output lines
20,15,7,1 22		20,15,7,1 15:1,7:1

File naming conventions

change_v3_**Rollno**.c (e.g., change_v2_MT2014001.c) - Roll number should be in upper case