F: Vending Machines

Time Limit: 1 second(s)

Chouchou loves sparkling soft drinks, but hates coins! These feelings make vending machines one of Chouchou's favourite things in the world since he can both get a drink and get rid of coins that he has. One day, he walked up to a vending machine and saw that a bottle of his favourite soft drink cost \$2.35. Looking in his pocket, his eyes lit up when he saw that he had exactly \$4.35 in small coins. If he put all of the \$4.35 into the machine, then the machine would give him a \$2 coin back and he would reduce the number of coins in his pocket dramatically!

But there was a problem. Once he had deposited at least \$2.35, the machine would not accept any more of his money since the machine was smart enough to know that there was already enough money for the product. Now he wonders, "If I had put the coins into the machine in a different order, could I have less coins?" This is where you come in. You are to write a program that tells Chouchou what coins to put into the machine (and in what order) so that he walks away with a soft drink and as few coins as possible.

To be precise, you need to select p coins in Chouchou's pocket (and an order of them) so that the value of the first $p-1$ coins is strictly less than the price of the item, while the value of all p coins is at least the price of the item. You may choose which coins the machine returns back to you, as long as it is correct change. The value you wish to minimise is the number of coins in Chouchou's pocket after the transaction (number of coins originally in Chouchou's pocket subtract p and add the number of coins the vending machine gave back in change).

Input

The input will start with two integers: n ($1 \leq n \leq 100$) and m ($1 \leq m \leq 100$) denoting the number of different denominations of coins and the number of coins in Chouchou's pocket, respectively.

The next line will contain n distinct real numbers. These n numbers represent the value of all of the different types of coins that this vending machine accepts and can return as change. Each coin's value is at most \$5.00. You may assume that a one cent coin (\$0.01) is always present on this line.

The third line will contain m real numbers. These m numbers represent the coins in Chouchou's pocket. There may be repeats of a certain type of coin. You may assume that each of these coins appear somewhere on the second line.

The final line will contain one real number denoting the price of the soft drink that Chouchou wishes to purchase. You may assume that Chouchou has enough money in his pocket to purchase the drink.

All real numbers given will have exactly 2 digits present after the decimal point and there will always be at least one digit to the left of the decimal point (e.g., `.01` is not valid).

Output

Your output should show the coins going into the machine and back from the machine as change that will minimise the number of coins in Chouchou's pocket.

The output should consist of exactly two lines. The first line should start with a number k, the number of coins that Chouchou will put into the machine, followed by k real numbers

representing the coins that Chouchou will put into the machine (in the correct order).

The second line should start with an integer ℓ, the number of coins that the machine will return back to Chouchou, followed by ℓ real numbers representing the coins that the machine will return from the machine. The order of the coins on the second line does not matter. You may assume that the vending machine contains an infinite supply of each type of coin.

All coins output should be given in the same format as the input. If there are multiple optimal solutions, you may output any of them.

Sample Input and Output

Sample Input	Output for Sample Input
7 7 0.01 0.05 0.10 0.20 0.50 1.00 2.00 0.10 0.10 0.10 0.20 0.50 1.00 2.00 2.35	5 0.10 0.10 0.10 0.20 2.00 2 0.05 0.10

Sample Input	Output for Sample Input
2 3 0.01 0.50 0.50 0.50 0.50 1.00	2 0.50 0.50 0