Problem 1: Smart Pricing

Time limit: 3 seconds

Australia is known for its love for sports and healthy lifestyle. The young Australians are very cautious about their diet and choice of food. Normally, in an Australian supermarket one can easily find several choices of the same food item, according to the nutrition and ingredient mix. In addition, there are many well established family businesses with a unique taste and association of customers.

Mr. Jamal is a young and enthusiastic entrepreneur who is running a 40 years old family owned fresh fruits juice and beverage business in Melbourne. They have a unique taste and a large base of customers all over Australia. This business was established by Jamal's grandfather, who migrated from Lebanon back in 1940. Jamal's grandfather uses a special secret ingredient to create a unique taste in beverages. He named that secret ingredient as "Lebon", which passed from generation to generation.

With the advancement of new technologies, Mr. Jamal wanted to set the price of his product (juices) according to the packing size and the amount of secret ingredient "Lebon" used in them. Interestingly, the small size packing bottles cost more than large size bottles due to the amount of labor required to prepare and pack them. In addition, other charges also add up to decide the end product price.

For this purpose, Mr. Jamal labeled the prices by assigning some parameters to each product. Given a list of products, each represented by a tuple containing the product name, bottle size (in liters), amount of Lebon used (in grams), the sticker cost, and the amount of fruit used (in kilogram KG). Interestingly, each product has a different amount of fruit used. As a good friend, Mr. Jamal asked you to find the longest possible sequence of products in the data with increasing bottling size and decreasing Lebon. The product with the lowest bottling-size to Lebon ratio should be considered the "cheapest". If there is a tie, the product with the lowest sticker cost should be considered as "cheapest". If there is still a tie, the product that uses the smallest amount of fruits should be considered "cheapest". Note that it is possible to skip a product when generating the sequence.

Input

The first line of input is an integer ***N (4 <= N <= 100)***, representing the total number of product entries a user wants to enter. The next ***N*** lines represent the input data by the user. The input data for each product is a ***5-tuple*** of the form (name, bottling-size, Lebon, sticker-cost, fruits), where each value is separated by a space. The list represents a single product's data per line. Each product is labeled with a different name. All the input values are integers.

Output

The first line should print an integer ***K***, indicating the length of the longest found sequence. The next ***K*** lines should print the name of the product in order of cheapest price. Such that the cheapest found product is on the top followed by the second cheapest product in the sequence and so on. Let ***j*** represent the output sequence order, such that ***j*** = 1 represent the cheapest product in the sequence, ***j*** = 2 is the second cheapest and so on, the output sequence must follow

$$B(1) < B(2) < B(3) \dots \dots < B(K) \qquad \text{and} \qquad P(1) > P(2) > P(3) \dots \dots > P(K)$$

Where ***B*** is the bottling-size and ***P*** represent the price of the product. For your answer to be correct, ***K*** must be as large as possible. It's guaranteed that there will only be one possible solution with max ***K***.

Sample input & output

The following is an example of a sample input and corresponding correct outputs.

Sample input	Sample Output
Sample 1:	
6 P 7 130 5 3 M 6 210 3 2 L 5 200 7 5 E 1 400 4 7 D 2 300 3 6 J 3 190 8 5	4 E D J P
Sample 2:	
4 C 4 130 7 2 J 2 200 9 4 W 1 180 7 5 F 2 160 5 5	3 W F C