

C. Item World

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Each item in the game has a level. The higher the level is, the higher basic parameters the item has. We shall consider only the following basic parameters: attack (*atk*), defense (*def*) and resistance to different types of impact (*res*).

Each item belongs to one class. In this problem we will only consider three of such classes: *weapon*, *armor*, *orb*.

Besides, there's a whole new world hidden inside each item. We can increase an item's level travelling to its world. We can also capture the so-called residents in the Item World

Residents are the creatures that live inside items. Each resident gives some bonus to the item in which it is currently located. We will only consider residents of types: *gladiator* (who improves the item's *atk*), *sentry* (who improves *def*) and *physician* (who improves *res*).

Each item has the size parameter. The parameter limits the maximum number of residents that can live inside an item. We can move residents between items. Within one moment of time we can take some resident from an item and move it to some other item if it has a free place for a new resident. We cannot remove a resident from the items and leave outside — any of them should be inside of some item at any moment of time.

Laharl has a certain number of items. He wants to move the residents between items so as to equip himself with weapon, armor and a defensive orb. The weapon's *atk* should be largest possible in the end. Among all equipping patterns containing weapon's maximum *atk* parameter we should choose the ones where the armor's *def* parameter is the largest possible. Among all such equipment patterns we should choose the one where the defensive orb would have the largest possible *res* parameter. Values of the parameters *def* and *res* of weapon, *atk* and *res* of armor and *atk* and *def* of orb are indifferent for Laharl.

Find the optimal equipment pattern Laharl can get.

Input

The first line contains number n ($3 \leq n \leq 100$) — representing how many items Laharl has.

Then follow n lines. Each line contains description of an item. The description has the following form: "*name class atk def res size*" — the item's name, class, basic attack, defense and resistance parameters and its size correspondingly.

- *name* and *class* are strings and *atk*, *def*, *res* and *size* are integers.
- *name* consists of lowercase Latin letters and its length can range from 1 to 10, inclusive.
- *class* can be "weapon", "armor" or "orb".
- $0 \leq atk, def, res \leq 1000$.
- $1 \leq size \leq 10$.

It is guaranteed that Laharl has at least one item of each class.

The next line contains an integer k ($1 \leq k \leq 1000$) — the number of residents.

Then k lines follow. Each of them describes a resident. A resident description looks like: "*name type bonus home*" — the resident's name, his type, the number of points the resident adds to the item's corresponding parameter and the name of the item which currently contains the resident.

- *name*, *type* and *home* are strings and *bonus* is an integer.
- *name* consists of lowercase Latin letters and its length can range from 1 to 10, inclusive.
- *type* may be "gladiator", "sentry" or "physician".

- $1 \leq \textit{bonus} \leq 100$.

It is guaranteed that the number of residents in each item does not exceed the item's size.

The names of all items and residents are pairwise different.

All words and numbers in the input are separated by single spaces.

Output

Print on the first line the name of the weapon in the optimal equipping pattern; then print the number of residents the weapon contains; then print the residents' names.

Print on the second and third lines in the same form the names of the armor and defensive orb as well as the residents they contain.

Use single spaces for separation.

If there are several possible solutions, print any of them.

Examples

input
<pre>4 sword weapon 10 2 3 2 pagstarmor armor 0 15 3 1 iceorb orb 3 2 13 2 longbow weapon 9 1 2 1 5 mike gladiator 5 longbow bobby sentry 6 pagstarmor petr gladiator 7 iceorb teddy physician 6 sword blackjack sentry 8 sword</pre>
output
<pre>sword 2 petr mike pagstarmor 1 blackjack iceorb 2 teddy bobby</pre>

input
<pre>4 sword weapon 10 2 3 2 pagstarmor armor 0 15 3 1 iceorb orb 3 2 13 2 longbow weapon 9 1 2 1 6 mike gladiator 5 longbow bobby sentry 6 pagstarmor petr gladiator 7 iceorb teddy physician 6 sword blackjack sentry 8 sword joe physician 6 iceorb</pre>
output
<pre>longbow 1 mike pagstarmor 1 bobby iceorb 2 petr joe</pre>

Note

In the second sample we have no free space inside the items, therefore we cannot move the residents between them.