Create a program that keeps track of enrolled heroes and their collection of spells (spellbook). You will be receiving the following commands until you receive the command "End":

* "Enroll (HeroName}":

Adds the hero to your collection of heroes.

If the hero is already present in your collection, print: "{HeroName} is already enrolled."

* "Learn (HeroName} {SpellName}":

Adds the spell to the hero's spellbook.

If the hero does not exist in the collection, print: "(HeroName} doesn't exist."

If the hero already has the spell in his spellbook, print: "(HeroName} has already learnt

(SpellName)."

* "Unlearn (HeroName} {SpellName}":

Removes the spell from the hero's spellbook.

If the hero doesn't exist in the collection, print: "{HeroName} doesn't exist."

If the spell doesn't exist in the hero's spellbook, print: "{HeroName} doesn't know

(SpellName)."

After receiving the "End" command, print all the heroes:

"Heroes:

== {name1): (spell1), (spell2), (spelln}

== {name2): {spell1), (spell2)}, {spelln)

…

== {nameN}: {spell1), (spell2}, {spelln}"

Input/Constraints

You will be receiving lines until you receive the "End" command

Output

Print the heroes in the format described above.

Input

· The possible commands are:

* "End"
* "Enroll {HeroName}"
* "Learn {HeroName} {SpellName}"
* "Unlearn {HeroName} {SpellName}"

Output

· The possible outputs are:

* "{HeroName} is already enrolled."
* "{HeroName} doesn't exist."
* "{HeroName} has already learnt {SpellName}."
* "{HeroName} doesn't know {SpellName}."
* "Heroes:

== {name1}: {spell1}, {spell2}, {spelln}

== {name2}: {spell1}, {spell2}, {spelln}

…

== {nameN}: {spell1}, {spell2}, {spelln}"

Вход

Enroll Stefan

Enroll Peter

Enroll Stefan

Learn Stefan ItShouldWork

Learn John ItShouldNotWork

Unlearn George Dispel

Unlearn Stefan ItShouldWork

End

Изход

Stefan is already enrolled.

John doesn't exist.

George doesn't exist.

Heroes:

== Stefan:

== Peter:

Вход

Enroll Stefan

Learn Stefan ItShouldWork

Learn Stefan ItShouldWork

Unlearn Stefan NotFound

End

Изход

Stefan has already learnt ItShouldWork.

Stefan doesn't know NotFound.

Heroes:

== Stefan: ItShouldWork

Вход

Enroll Stefan

Enroll Peter

Enroll John

Learn Stefan Spell

Learn Peter Dispel

End

Изход

Heroes:

== Stefan:

Spell == Peter:

Dispel == John: