# Gotta catch 'em all

Java May'19 DSA Set&Map 2 - 4 days 07:41:47

You are given the task to implement a system that ranks different pokemons. Each one of them has a name, type, power and current position on the leaderboard. The system should be able to add new pokemons, find all pokemons of a given type and get the ranking in some range (start and end positions).

Here are the commands that must be implemented:

- **add POKEMON_NAME POKEMON_TYPE POKEMON_POWER POKEMON_POSITION** - adds new pokemon to the rank list;
  - **POKEMON_NAME** can be any sequence from 1 to 20 characters and may not be unique;
  - **POKEMON_TYPE** can be any sequence from 1 to 10 characters and may not be unique;
  - **POKEMON_POWER** can be any integer between 10 and 50;
  - **POKEMON_POSITION** can be any integer between 1 and the current pokemons count plus one (e.g. if there are 2 pokemons already, **POKEMON_POSITION** can be 1, 2 or 3). If a pokemon is inserted to an already used position, all pokemons' positions from this position till the end are incremented by one (e.g. if the ranking system has Pokemon1 in position 1, Pokemon2 in position 2 and is inserting Pokemon3 in position 1 => Pokemon1 goes to position 2 and Pokemon2 goes to position 3).
  - Prints: **Added pokemon POKEMON_NAME to position POKEMON_POSITION**;
- **find POKEMON_TYPE** - finds the top 5 pokemons, first ordered by name in ascending order and then by power in descending order;
  - Prints the results in the following format "**Type POKEMON_TYPE: POKEMON; POKEMON; POKEMON**" where **POKEMON** should be printed in the format `POKEMON_NAME(POKEMON_POWER)`. If no pokemons are found, just print `Type POKEMON_TYPE:` (ending whit one space).
- **ranklist START END** - prints the rank list from **START** to **END** positions;
  - **START** can be any integer between 1 and current pokemons count;
  - **END** can be any integer between 1 and current pokemons count (and will be greater than or equal to START);
- **end** - marks the end of the commands and no other commands will follow afterward.

## Input

Submit solution

All submissions
Best submissions

**Points:** 100 (partial)
**Time limit:** 1.5s
Java: 4.0s
**Memory limit:** 128M
Java: 128M
**Author:** viktor

**Tags**
Hash Tables
**Difficulty**
Intermediate

**Allowed languages**
java

will be valid (as described in the above list), in the specified format, within the constraints given **check the input data explicitly.**



## Output

- The output data should be printed on the console.
- For each command from the input sequence print at the standard output its result as a single line.

## Constraints

- All **POKEMON_NAME** and **POKEMON_TYPE** will consist of letters and digits only. No spaces are allowed.
- The total number of lines in the input will be in the range [1 ... 100000].

## Sample tests

### Input

Copy

```
add Pikachu Electric 49 1
add Spearow Flying 25 2
add Jigglypuff Fairy 30 3
add Squirtle Water 22 2
add Squirtle Water 40 1
find Water
ranklist 1 5
add Charmander Fire 33 2
find Fire
ranklist 1 3
end
```

### Output

```
Added player Jigglypuff to position 3
Added player Squirtle to position 2
Added player Squirtle to position 1
Type Water: Squirtle(40); Squirtle(22)
1. Squirtle(40); 2. Pikachu(49); 3. Squirtle(22); 4. Spearow(25); 5. Jigglypuff(30)
Added player Charmander to position 2
Type Fire: Charmander(33)
1. Squirtle(40); 2. Charmander(33); 3. Pikachu(49)
```

## Clarifications

No clarifications have been made at this time.