Problem 3. Listmon's Monsters

Input / Constraints

An evil genius never works alone. Listmon has own army. He wants to start tracking data for his army. You have been asked to create a data base for Listmon's monsters.

A **basic** monster will always have:

- **name - string of any asci character without space and comma**
- **id – integer number**
- **strength - integer point number**
- **ugliness - integer number**

In his army he accepts only two types of monsters: Hydralisks and Zerglings. Both types has the same parameters as a basic monsters but also:

- Hydralisk has another property – range - **string of any asci character without space and comma**
- **Zergling has another property – speed – integer number**

Listmon does not get in his army basic monsters. They must be one of the two types mentioned above.

You will start receiving input data in format:

*** Hydralisk({name}, {id}, {strength}, {ugliness}, {range})**

-if there is no fifth parameter you should print ***'__init__() missing 1 required positional argument: 'range''***

-if the fifth parameter is not a string you should print ***'Range must be string'***

In both cases you do not add the monster in DB

*** Zergling({name}, {id}, {strength}, {ugliness}, {speed})**

-if there is no fifth parameter you should print ***'__init__() missing 1 required positional argument: 'speed''***

-if the fifth parameter is not a string you should print ***'Speed must be integer'***

In both cases you do not add the monster in DB

Is it possible a basic monster to try to apply in the army. If you receive a command:

***BasicMonster({name}, {id}, { strength }, {ugliness})**

You must print **'Can't instantiate abstract class BaseMonster with abstract methods __init__'** and you must not add it in the DB.

Output

When you receive a command '**stopAddingArmy' you must print the information about the army in the following format:**

- Overall speed of army: {speed}
- Overall strength: {strength}
- Hydralisk: {count}; Zergling: {count}

Where {speed} is the sum of all Zerglings monsters speed in DB;

{strength} is the sum of strength of all monsters;

{count} is the number of all Hydralisks/Zerglings in the DB

Examples

Input	Output	Comments
Zergling('Pesho', 10, 10, 10) Zergling('Pesho', 10, 10, 20) Hydralisk('a', 100, 100, 100, 'min') Zergling('Pesho', 10, 10, 10, 30) stopAddingArmy	Overall speed of army: 60 Overall stength: 130 Hydralisk: 1; Zergling: 3	On the first row we receive a Zergling. We check that there are five parameters, and the last one is integer number, so we add it in DB. Same for the second row. Third is Hydralisk. We check if there are five parameters and the last one is string so we add it. Fourth is the same as the first two. After a command 'stopAddingArmy' we print the result. Overall speed of the army is all Zerglings speed and we sum the results – 10 +20+30 = 60. Overall strength is the sum of all strength of every soldier in the army: 10+10+100+10 = 130. After that we should count the amount of Hydralisks which in this case is 1, and the amount of Zaerglings which is 3.

Input	Output	Comments
BaseMonster('A12', 150, 200, 300) Zergling('Pesho', 10, 10, 10, 'min') Hydralisk('a', 100, 100, 100, 10) Zergling('Pesho', 10, 10, 10) Hydralisk('a', 100, 100, 100) stopAddingArmy	Can't instantiate abstract class BaseMonster with abstract methods __init__ Speed must be integer Range must be string __init__() missing 1 required positional	

	argument: 'speed' __init__() missing 1 required positional argument: 'range' Overall speed of army: 0 Overall stength: 0 Hydralisk: 0; Zergling: 0	