

The company you work for has been doing well recently and wants to hire a lot more employees! It is interested in software developers, particularly senior ones. Data about candidates who successfully passed the recruitment process is saved in the table `candidates`.

```
create table candidates (  
    id int primary key,  
    position varchar not null,  
    salary int not null  
);
```

Each record in this table contains a unique `id` for each candidate, their `position` (which is either "junior" or "senior") and their negotiated `salary` in dollars per month. Candidates have increasing ids and are sorted in ascending order by salaries.

Your company's monthly budget for new employees' salaries is 50,000 dollars. It wants to hire as many senior developers as possible, and then use the remaining money to hire as many junior developers as possible.

You were assigned the task of writing a SQL query determining how many employees can be hired for each position. It should return an `employees` table with two columns: `juniors` and `seniors`. The table should contain one row with the number of new employees in each position.

Examples:

1. Given table:

id	position	salary
20	junior	10000
30	senior	15000
40	senior	30000

Your query should return:

juniors	seniors
0	2

The company can hire both senior candidates for 45,000 dollars in total, but there will not be enough money left to hire the junior developer candidate.

2. Given table:

id	position	salary
1	junior	5000
2	junior	7000
3	junior	7000
4	senior	10000
5	junior	10000
6	senior	20000
7	senior	30000

Your query should return:

juniors	seniors
3	2

The company can hire two of the three senior developer candidates. Hiring all three of them would cost 60,000 dollars. After hiring candidates with ids 4 and 6, there will be 20,000 dollars left for hiring junior developer candidates. This is enough to hire candidates with ids 1, 2 and 3.

3. Given table:

id	position	salary
1	junior	15000
2	junior	15000
3	junior	20000
4	senior	60000

Your query should return:

juniors	seniors
3	0

The company can't afford to hire the candidate with id 4, but can afford all the others.

Assume that:

- column `position` contains only values "junior" and/or "senior";
- column `salary` contains only positive integers not greater than 100,000;
- in table `candidates` values of the column `id` are in increasing order and values of the column `salary` are in non-decreasing order.