Grading

After the examination period, the class teacher of a class in secondary education wants to determine the student with the highest grades in his class. This is the student with the highest weighted average. The software that manages the exam results can only generate a list containing the score of each students per subject. The average grades per student must therefore still be calculated, taking into account the weight of each subject. The class teacher has requested your help to automate the process of determining who the student with the highest grades is. Therefore, write a function

```
def top_student(grades, courses):
```

which given a list of tuples, called Grades, of the form ('student','course',grade) and a list of tuples, called Courses, of the form ('course',weight) should return the name of the student at the top of the class, i.e. the student with the heigest weighted average.

You can assume that in the given lists grade and weight are integers, where grade is always a number between 0 and 10 (both inclusive), and weight is always a number between 1 and 9 (both inclusive). In addition, you may assume that the names of the persons are spelled the same everywhere, including capials. You can also assume that every student follows every course, that every course in Grades is also in Courses, and that there is always exactly one student with the top grades. Note that the given lists Grades and Courses are not necessarily sorted by student, course and/or weight.

The function `top_student(grades, courses)` will be passed the valid parameters, two lists of tuples in this case.

Example

For example, consider the following lists:

```
s1 = [('Alice','Bio',1), ('Bob','NED',8), ('Bob','Bio',0), ('Alice','LO',7), ('Alice','NED',9), ('Bob','LO',10)]
v1 = [('Bio',3), ('NED',5), ('LO',2)]
```

To determine a student's weighted average, first calculate the sum of the scores per subject multiplied by the weight of this subject. Then divide this sum by the total weight. This result is then the weighted average. To illustrate, calculate the weighted averages of *Alice* and *Bob* in the example above:

$$Alice: \frac{(1*3)+(9*5)+(7*2)}{3+5+2} = \frac{62}{10} = 6.2$$

$$Bob: \frac{(0*3)+(8*5)+(10*2)}{3+5+2} = \frac{60}{10} = 6.0$$

Since *Alice* has the highest grades, then function call should return *Alice*.

As an additional example, consider the lists:

```
s2 = [('Liam','Lat',6), ('Liam','Math',5), ('Violet','Lat',7), ('Nora','math',5), ('Nora','Lat',2), ('Violet','math',8)]
v2 = [('Lat',1), ('Math',4)]
```

The weighted averages of Liam, Violet and Nora are 5.2, 7.8 and 4.4 respectively. So the function call `top_student(s2, v2)` returns Violet.

Working

```
>>>top_student(s1, v1)
'Alice'

>>>top_student(s2, v2)
'Violet'
```