

Python weekend Kiwi.com - Entry task

By solving this task, you will prove to us and yourself, that you are familiar with the basics of python, in which case we believe that you will enjoy the weekend workshop and it won't be too hard or too boring for you.

Requirements

- python-2 or 3 (but you know the right choice here ;-))
- all modules are allowed

Description

You have data about flights (segments). Your task is to find all combinations of flights for passengers with *no bags*, *one bag* or *two bags* are able to travel, having 1 to 4 hours for each transfer between flights. The columns in table of input data are explained bellow:

- **source**, **destination** are the code of airport the flight is departing from and arriving to
- **departure**, **arrival** are times of departure and arrival
- **price** is the price of flight per person (without baggage)
- **bags_allowed** the number of bags passenger is allowed to take with them
- **bag_price** additional price per each bag passenger would like to take with them
- **flight_number** is the unique identifier of each flight

For easy navigation in offered flight combinations (*itineraries*), it would be nice to show total prices to passengers that already include the additional price for bags, given they input how many bags they wish to take when they search for flights.

Input data (example)

```
source,destination,departure,arrival,flight_number,price,bags_allowed,bag_price
USM,HKT,2019-05-11T06:25:00,2019-05-11T07:25:00,PV404,24,1,9
USM,HKT,2019-05-12T12:15:00,2019-05-12T13:15:00,PV755,23,2,9
USM,HKT,2019-05-12T21:15:00,2019-05-12T22:15:00,PV729,25,1,14
USM,HKT,2019-05-11T14:50:00,2019-05-11T15:50:00,PV966,21,1,17
USM,HKT,2019-05-12T00:35:00,2019-05-12T01:35:00,PV398,24,1,14
USM,HKT,2019-05-12T05:15:00,2019-05-12T06:15:00,PV870,19,1,13
USM,HKT,2019-05-12T10:00:00,2019-05-12T11:00:00,PV320,22,1,18
USM,HKT,2019-05-12T13:40:00,2019-05-12T14:40:00,PV540,26,2,13
USM,HKT,2019-05-12T09:30:00,2019-05-12T10:30:00,PV290,19,2,8
USM,HKT,2019-05-11T02:40:00,2019-05-11T03:40:00,PV876,25,2,16
USM,HKT,2019-05-12T09:35:00,2019-05-12T10:35:00,PV275,24,2,17
```

```

HKT,USM,2019-05-12T10:35:00,2019-05-12T11:30:00,PV996,23,1,15
HKT,USM,2019-05-11T15:45:00,2019-05-11T16:40:00,PV243,22,1,6
HKT,USM,2019-05-11T19:05:00,2019-05-11T20:00:00,PV146,21,2,5
HKT,USM,2019-05-12T16:00:00,2019-05-12T16:55:00,PV634,21,1,12
HKT,DPS,2019-05-12T00:00:00,2019-05-12T03:40:00,PV961,70,1,39
HKT,USM,2019-05-12T00:20:00,2019-05-12T01:15:00,PV101,18,2,7
HKT,DPS,2019-05-11T12:00:00,2019-05-11T15:40:00,PV100,96,1,40
HKT,USM,2019-05-12T22:05:00,2019-05-12T23:00:00,PV672,24,2,5
HKT,USM,2019-05-11T06:30:00,2019-05-11T07:25:00,PV442,17,1,11
HKT,USM,2019-05-12T07:15:00,2019-05-12T08:10:00,PV837,18,1,12
BWN,DPS,2019-05-11T06:10:00,2019-05-11T08:30:00,PV953,48,1,25
BWN,DPS,2019-05-12T14:35:00,2019-05-12T16:55:00,PV388,49,1,30
BWN,DPS,2019-05-11T05:35:00,2019-05-11T07:55:00,PV378,59,1,29
BWN,DPS,2019-05-12T10:35:00,2019-05-12T12:55:00,PV046,50,1,25
BWN,DPS,2019-05-11T13:40:00,2019-05-11T16:00:00,PV883,51,1,26
BWN,DPS,2019-05-12T19:10:00,2019-05-12T21:30:00,PV999,54,2,23
BWN,DPS,2019-05-11T16:15:00,2019-05-11T18:35:00,PV213,55,2,22
BWN,DPS,2019-05-11T02:35:00,2019-05-11T04:55:00,PV873,46,1,34
BWN,DPS,2019-05-11T01:15:00,2019-05-11T03:35:00,PV452,57,1,33
BWN,DPS,2019-05-12T08:45:00,2019-05-12T11:05:00,PV278,41,2,22
BWN,DPS,2019-05-12T22:50:00,2019-05-13T01:10:00,PV042,56,2,31
DPS,HKT,2019-05-12T08:25:00,2019-05-12T12:05:00,PV207,83,1,38
DPS,BWN,2019-05-12T17:15:00,2019-05-12T19:40:00,PV620,43,2,25
DPS,BWN,2019-05-11T13:15:00,2019-05-11T15:40:00,PV478,47,1,23
DPS,HKT,2019-05-11T09:15:00,2019-05-11T12:55:00,PV414,67,1,49
DPS,HKT,2019-05-12T08:25:00,2019-05-12T12:05:00,PV699,78,2,41
DPS,HKT,2019-05-12T15:20:00,2019-05-12T19:00:00,PV974,85,1,38
DPS,HKT,2019-05-11T00:20:00,2019-05-11T04:00:00,PV519,79,2,44
DPS,HKT,2019-05-11T08:50:00,2019-05-11T12:30:00,PV260,89,1,43
DPS,BWN,2019-05-12T16:45:00,2019-05-12T19:10:00,PV451,57,1,24
DPS,BWN,2019-05-11T22:10:00,2019-05-12T00:35:00,PV197,50,1,30

```

Output

- Output data should be in a format that is suitable for further processing
- Don't make passengers travel through the same cities in same trip:
 - A->B->A->B is not a valid combination
 - A->B->A is a valid combination

Solving

- The task has been designed such that it provides a challenge to someone who has only recently familiarized themselves with python and provides a chance to use basic data types and control structures (*if*, *for*, etc.). Most people who have been programming in other languages won't have problems solving it and get a chance to try the python syntax.
- The solution should be simple
- We don't expect the solution to be optimized for computational complexity or memory usage (however if it's your thing, you may)

Usage

Input data will be fed into your program through `stdin` so it should be possible to run it in command line via a command such as `cat input.csv | find_combinations.py`. The output of your program will be printed to `stdout` and any errors will go to `stderr`.