Problem F
Young, Poor and Busy
Input: young.txt

Ken and Keiko are young, poor and busy. Short explanation: they are students, and ridden with part-time jobs. To make things worse, Ken lives in Hakodate and Keiko in Tokyo. They want to meet, but since they have neither time nor money, they have to go back to their respective jobs immediately after, and must be careful about transportation costs. Help them find the most economical meeting point.

Ken starts from Hakodate, Keiko from Tokyo. They know schedules and fares for all trains, and can choose to meet anywhere including their hometowns, but they cannot leave before 8am and must be back by 6pm in their respective towns. Train changes take no time (one can leave the same minute he/she arrives), but they want to meet for at least 30 minutes in the same city.

There can be up to 100 cities and 2000 direct connections, so you should devise an algorithm clever enough for the task.

Input

The input is a sequence of data sets.

The first line of a data set contains a single integer, the number of connections in the timetable. It is not greater than 2000.

Connections are given one on a line, in the following format.

Start_city HH:*MM Arrival_city HH*:*MM price*

Start_city and *Arrival_city* are composed of up to 16 alphabetical characters, with only the first one in upper case. Departure and arrival times are given in hours and minutes (two digits each, separated by ":") from `00:00` to `23:59`. Arrival time is strictly after departure time. The *price* for one connection is an integer between 1 and 10000, inclusive. Fields are separated by spaces.

The end of the input is marked by a line containing a zero.

Output

The output should contain one integer for each data set, the lowest cost possible. This is the total fare of all connections they use.

If there is no solution to a data set, you should output a zero.

The solution to each data set should be given in a separate line.

Sample Input

```
5
Hakodate 08:15 Morioka 12:30 2500
Morioka 14:05 Hakodate 17:30 2500
Morioka 15:30 Hakodate 18:00 3000
Morioka 14:30 Tokyo 17:50 3000
Tokyo 08:30 Morioka 13:35 3000
4
Hakodate 08:15 Morioka 12:30 2500
Morioka 14:04 Hakodate 17:30 2500
Morioka 14:30 Tokyo 17:50 3000
Tokyo 08:30 Morioka 13:35 3000
18
Hakodate 09:55 Akita 10:53 3840
Hakodate 14:14 Akita 16:09 1920
Hakodate 18:36 Akita 19:33 3840
Hakodate 08:00 Morioka 08:53 3550
Hakodate 22:40 Morioka 23:34 3550
Akita 14:23 Tokyo 14:53 2010
Akita 20:36 Tokyo 21:06 2010
Akita 08:20 Hakodate 09:18 3840
Akita 13:56 Hakodate 14:54 3840
Akita 21:37 Hakodate 22:35 3840
Morioka 09:51 Tokyo 10:31 2660
Morioka 14:49 Tokyo 15:29 2660
Morioka 19:42 Tokyo 20:22 2660
Morioka 15:11 Hakodate 16:04 3550
Morioka 23:03 Hakodate 23:56 3550
Tokyo 09:44 Morioka 11:04 1330
Tokyo 21:54 Morioka 22:34 2660
Tokyo 11:34 Akita 12:04 2010
0
```

Output for the Sample Input

```
11000
0
11090
```