Ski pass

The users of a ski resort are charged at the end of the day for accessing the ski facilities. The price is computed automatically based on the time slots the user accessed the facilities. The data are stored in two distinct files

The first file, named "*users.txt*", contains, one per row, the data of the first and last usage of the facility of each user on different days. The format is the following:

```
<id> <yyyy-mm-dd> <hh:mm:ss> <HH:MM:SS>
```

where `<id>` is an alphanumeric code identifying the user, `<yyyy-mm-dd>` is the day of corresponding to the row, `<hh:mm:ss>` is the time of the first usage of the facility and `<HH:MM:SS>` is the time of the last usage of the facility.

A second file, named "*prices.txt*", contains the information on prices of different time slots (one slot per row). The format is the following

```
<slot code> <hh:mm>-<HH:MM> <price>
```

where `<slot code>` is an alphanumeric code representing the time slot, `<hh:mm>` and `<HH:MM>` are the start and end time of the slot (note that these two times are separated by a dash "-"), and `<price>` is a real value representing the time slot price.

Write a Python program to compute the price for each user for a given date, specified in a global variable `TARGET_DATE` as a string in the format `<yyyy-mm-gg>` . In particular, the program must compute the time slot in which the facilities where used by each user and compute the corresponding price. Since the time slots may overlap, multiple time slots may be compatible with the user passages. In this case, the price for the user must be the lowest of all time slots that are compatible with he user passages. For example, with the files below, a user whose first passage was at 9:00 and last passage was at 11:00 would be compatible with both the Morning and the Full time slots. Since the Morning time slot has a lower price, he should be charged the price of the Morning slot.

The program must compute and print:

- The cheapest slot and the corresponding price for each user (for the specified date, in no particular order)
- The total income for the specified date
- The percentage of users for each time slot on the specified date (in no particular order)

Example

Given the file *users.txt*

```
A112233445 2019-12-20 09:01:58 10:30:00
A997733445 2019-12-20 09:39:53 11:00:53
A112233445 2019-12-21 09:01:58 16:04:58
A992233445 2019-12-21 10:02:58 12:02:58
A882233445 2019-12-21 10:15:03 11:15:00
A445566778 2019-12-21 12:35:44 16:35:44
A882233448 2019-12-21 15:15:03 16:00:02
```

and the file *prices.txt*

```
Morning 08:00-13:00 30
Afternoon 12:00-17:00 35
Full 08:00-17:00 50
```

and defining `TARGET_DATE = "2019-12-21"` the program should print

```
Skipass A112233445: Full - 50.00 EURO
Skipass A992233445: Morning - 30.00 EURO
Skipass A882233445: Morning - 30.00 EURO
Skipass A445566778: Afternoon - 35.00 EURO
Skipass A882233448: Afternoon - 35.00 EURO

Total income: 180.0

User stats:
Morning: 40.0%
Afternoon: 40.0%
Full: 20.0%
```