In a school, parents can schedule appointments with teachers for parent-teacher conferences through a web interface. In this task, you need to work with the data of one of the conference days. The conference day runs from 16:00 to 18:00, and the possible bookable time slots are: 16:00, 16:10, 16:20... 17:50. Each appointment lasts for 10 minutes. The scheduling program does not allow time slot conflicts.
The 'fogado.txt' file contains information about the availability of teachers. Each line contains the following data separated by spaces: the teacher's last name, first name, the booked time slot, and the date and time of the reservation. The teacher's name consists of exactly one last name and exactly one first name. In the file, you will find each of the hour, minute, month, and day data stored with exactly two digits. There are definitely fewer than 500 lines in the file, and the order of the data is random.

```
…
Nagy Marcell 16:30 2017.10.29-20:32
Fodor Zsuzsanna 17:10 2017.10.28-23:12
Lakatos Levente 16:00 2017.10.30-08:24
…
```

According to the first line of the example, the 16:30 time slot was booked with Mr. Marcell Nagy, on October 29, 2017, at 20:32.
Create a program that uses the data from the 'fogado.txt' file to answer the following questions! Save the source code of the program under the name 'receptionHr'! (When writing the program, you don't need to check the correctness or validity of the data provided by the user; you can assume that the available data conforms to the description.) Before displaying the results of subtasks that require writing to the screen, display the task number on the screen (e.g., Task 2:). When requesting input from the user, indicate on the screen what type of value is expected. Accents in text are also accepted.

1. Read and store the contents of the 'fogado.txt' file!
2. Display on the screen how many booking records the file contains!
3. Ask the user for a teacher's name, and then display on the screen, as per the example, how many appointments the specified teacher has! If there are no appointments for the given teacher – for example, Farkas Attila – then display the message 'A megadott néven nincs időpontfoglalás.' ('There are no appointments for the specified name.').
4. Ask the user for a valid time in the format found in the source file (e.g., 17:40)! The program should display on the screen the list of teachers booked at the specified time! One name should appear per line in alphabetical order. Write the sorted list of names to a file, with one name per line. Use the file name corresponding to the time, for example, store the data in the file '1740.txt' for 17:40! Ensure that the file name does not contain the colon character. (If you cannot create a file with this name, use the 'adatok.txt' filename instead!)
5. Determine and then display on the screen all the details of the earliest booked appointment! When displaying the information, follow the format provided at the end of the task exactly!
6. Display 'Barna Eszter' teacher's available time slots on the screen! We know that the teacher has at least one booked and multiple available time slots. When can the teacher leave school at the earliest? Clearly display the identified time on the screen!