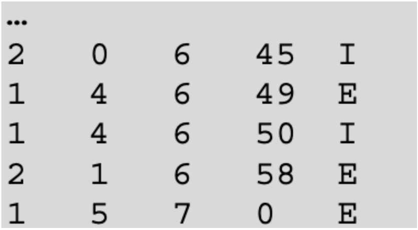
Schedule

Several years ago, in the eastern part of the country, a regular schedule was introduced. This means that trains depart from the terminal every hour and arrive at each station at the same time according to the schedule. Due to the well-planned travel, the number of travelers has increased.

The vonat.txt file records the departure and arrival data for some trains on the Szeged-Budapest route. In the lines of the file, there are five tab-separated values, four integers, and one character. The first number is the train identifier, the second is the station identifier, the third and fourth represent the hour and minute of a timestamp. The character indicates whether the train arrives (E) or departs (I) from the specified station at the given time.

The number of rows is at most 1000, and the identifiers for trains and stations are integers between 0 and 20. The hour values range from 0 to 23, and the minutes are between 0 and 59. Stations are numbered from 0 onwards in terms of distance, while trains are numbered sequentially in increasing order based on departure time, encompassing all values from 1. The file records the actual paths of the trains. The data is arranged in chronological order, and within that, except for the initial station, arrivals always precede departures. We know that every train departs from station 0 and reaches the final station. During this journey, they stop at each station, and no train overtakes another.



```
…
2    0    6    45   I
1    4    6    49   E
1    4    6    50   I
2    1    6    58   E
1    5    7    0    E
```

The first line indicates that the 2nd train departs from the starting station at 6:45 AM. In the following line, it states that the 1st train arrives at the 4th station at 6:49 AM. Create a program that, using the data from the vonat.txt file, answers the following questions! Save the source code of the program as 'schedule'. (When writing the program, you do not 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.)

1. Read and store the contents of the vonat.txt file.
2. Display on the screen the number of trains and stations stored in the file, including the starting and final stations!
3. Determine which station had the most trains stop at it! Provide the train and station identifiers, as well as the duration of the stop! If there were multiple instances, you can just provide one of them.
4. Read in a train identifier and a timestamp (hour and minute values). Use these for later tasks!
5. On this route, the prescribed travel time is 2 hours and 22 minutes. Display on the screen how many minutes the train with the read identifier deviates from this time! For example: 'The journey of train number 5 was 2 minutes shorter than prescribed,' 'The journey of train number 5 took exactly the prescribed time,' or 'The journey of train number 5 was 3 minutes longer than prescribed.
6. Write to the haladX.txt file, indicating when the train with the read identifier arrived at which station! Replace X in the filename with the read train identifier!
7. Determine the current location of the trains that have already started but have not yet reached the final station at the read timestamp! During testing, it is worth paying attention to the following timestamps: 6:50, 8:45, 9:05, 10:04, 10:20.