BOWLING

Create a program structured in Python language that manages the score of a bowling game. The match score is recorded in the “bowling.txt” file which shows the score of all registered players; each line of the file stores information about a single player. The format of each line is as follows:

```
<surname>;<name>;<shot_score_1>;<shot_score_2>; ... ;<shot_score_n>
```

Make the following assumptions:

- the number of players is not known a priori;
- the number of shots by a player is not known in advance and is not the same for all players (it depends on how many strikes a player did during the match);
- the fields of a row are separated from each other by a semicolon;
- cases of homonymy (same name) are not possible;
- the file format is correct.

The program must:

- upload the information contained in the “bowling.txt” file
- return the ranking ordered by decreasing scores by printing the fields: `<surname> <name> <final score>` in columns
- return the players who have collected more ‘10’ (if they exist) and more ‘0’ (if they exist)

Example of “bowling.txt” file:

```
Rossi;Massimo;7;10;6;5;10;4;9;9;5;10;10
Verdi;Giuseppe;10;10;6;6;7;9;9;8;9;9;10;10;10
De Piscopo;Tullio;9;9;8;8;7;7;6;6;0;5
Montalbano;Salvo;10;10;9;10;10;10;9;10;10;10
```

Corresponding screen output generated by the program

The program must print on the screen:

```
Verdi Giuseppe 113
Montalbano Salvo 98
Rossi Massimo 85
De Piscopo Tullio 65
Montalbano Salvo has knocked down all the pins 8 times
De Piscopo Tullio has missed all the pins 1 time (s)
```