A WEB server stores all access on the WEB performed by all employees' computers. The format of the file is the following:

178.1.192.32 goodguy [11/01/2021:15:21:43] POST /intranet/login 200
178.1.192.42 nastyguy [10/01/2021:18:56:01] POST /intranet/login 200
178.1.192.33 goodguy [10/01/2021:13:55:36] GET index.html 200
34.52.1.33 badguy [10/01/2021:14:04:10] POST /services/ask 200
34.52.177.48 okguy [10/01/2021:14:32:00] POST /intranet/login 401
178.1.192.41 nastyguy [10/01/2021:18:29:01] POST /intranet/login 401
34.52.1.33 badguy [10/01/2021:14:06:20] POST /intranet/login 401
123.154.48.1 worstguy [11/01/2021:00:21:32] GET /services/list 200

Where the different fields report:

- The IP address of the host performing the WEB access
- The user name, the data, and time of the access,
- The type of access (GET or POST)
- The name of the resource searched by the host: 200 when the request was accepted and 401 when it was not accepted.

Suppose a reasonable format and a maximum length for each one of these fields. IP addresses and user names are unique within the database.
The candidate must write a C program able to store all information contained in the file in a data structure suitable to print all accesses performed by an IP address or a user name using as a search key either the IP address or the user name. More specifically, given an IP address **or** a user name, the data structure must guarantees searches with **logarithmic time costs**, in the average case, using either the IP address **or** the user name as a key. Moreover, once the entry of the IP address or the user name has been found, displaying the list of accesses performed by this IP or user must have a linear cost in the number of WEB accesses.

For example, given the previous file and the IP 34.52.1.33 the search must return the name of the user, i.e., badguy, and all information about his/her 2 accesses to the WEB, i.e., the ones of the [10/01/2021:14:04:10] and the one of the [10/01/2021:14:06:20]. Similar information has to be displayed if the search key is the name, i.e., badguy.

The candidate must:

- Define in C code the ADTs required to satisfy these constraints.
- Write in C code the function **read**, which receives the name of the file containing the information, stores it in the data structure, and returns the pointer to such a structure. Notice that the function **read** must make appropriate use of all required library functions.