Ankara University
Computer Engineering
COM2067 LAB 4
2024-2025 Fall

Assume that you have a list of people. Each person has a phone number. You will get the person and phone information from the user, the person entry will end with -1. You will get one character from the user at each step and list all the records in the list that match the name. This process will continue until the user enters -1 and exits, or returns a single name as a result, or no person is found (**nobody** will be printed on the screen).

For example, your list of people (person name and phone number) is as follows:

mehmet 123
eda 124
remzi 125
rana 126
taner 127
tuana 128
pelin 129
baran 130
dursun 131
emin 132
derda 133
melih 134
veysel 135
elif 136
emel 137

When the user enters the character ‘e’:

‘e’ :
eda
elif
emel
emin

should be listed. The list should be printed in alphabetical order. Since the above termination criteria are not met, the program continues to receive characters.

‘em’ :
emel
emin

‘emi’ :
emin

The program ends.

You must write the solution in C using a **binary search tree.**

Submission:

Name your source file as <StudentID>.c. For example, if your ID is 22290777, then you will submit **22290777**.c file. For the correct output format, carefully examine the sample input and output files provided to you. You can perform the following operations to check the correctness of your program.

Testing:

We provide a sample input/output text file pairs for you to test your codes at Ubuntu. Please carefully review the sample input and output files given to you for the correct output format.

We recommend you to use input redirection mechanism of your operating system to test your programs. For example, if your executable is called as Lab3, redirect the input.txt file to standard input using < operator and redirect your outputs to a file using > operator such as:

```
> ./Lab4 < input1.txt > output1.txt
```

This kind of execution enables your programs to read inputs from a file without writing any file related functions. In other words, scanf reads data from the redirected files instead of the std. input in this way (e.g. keyboard).

Automatically compare your own output with the expected output by using the diff myOutput1.txt output1.txt command. If a warning as shown below does not appear on the screen after executing this command, this means that your program is working correctly. If you see a warning in the command system after executing the command, this indicates that there is a problem with your output.

Test your program for different inputs that you will create yourself. Please note that the input files given to you and the input files used during the evaluation may differ from each other.