Data Structure Assignment [6]

General information

- Deadline: 2020/12/3 12:00 noon time.
- Submit your programming assignment to the Moodle system.
- Submitted file format: student-ID Name.zip (e.g. F12345678_王曉明.zip.) Please name the filename of your submitted compressed file after your **student ID number. Otherwise, 20 points will be deducted.**
- Your submitted files have to be organized in a directory-structured manner as follows. Otherwise, **5 points will be deducted.**

  | -- F12345678_王曉明
  | -- F12345678_王曉明.pdf
  | -- code
  | -- xxxxx.c
  | -- xxxxx.c
- Your submitted file must contain **Source Code** & **Readme file** (Program description in .pdf format)
- **Late submission will not be accepted.**

There is a “zero tolerance” for plagiarism. You will receive a score of zer o if you get caught plagiarizing.

Course Provisions

1. Program execution environment: Windows (Git Bash)
   Command: ***./hw.exe < input.txt > output.txt***
2. Programming language: C (standard: C11)
   (**Languages other than C are not accepted**)
3. Submitted programming homework must include **source code** in .c data type, and **readme document** in .pdf data type. You are required to address the **(1) result screenshot, (2) program structure, (3) program functions** in readme file. Do not just write the pseudo code or even just copy and paste your code!
4. Homework grading is divided into two parts: 80% for the code and 20% for the readme file. The remaining grading standards are determined by the TAs.

TA office hours of the course:

Wed. 11:30 – 13:00

Lab location: CSIE Bldg. Room 65302

If you have any question, please make an appointment in advance.

You can also mail us about your questions.

TA e-mail: ta_@dblab.csie.ncku.edu.tw

Programming homework

AVL tree implementation

In this assignment, you are required to develop a program to construct an **AVL search tree** for fast searching. After building the AVL tree, you need to print out the nodes of the AVL tree in **Pre-Order**.

The input will contain two parts, each part starting with one letter. The first part of the input, starting with "D", is the data used to construct the AVL search tree. There will be N records in the data, and each record will contain a name and a phone number. Each node of your AVL tree will contain only one record. Hence, each node will contain only one name and one phone number. You need to build the AVL search tree **using the names only**. The insertion order has to be the same as the order of names given in INPUT. After constructing the AVL search tree, print out the content of the nodes of the AVL search tree in **Pre-Order.**

The second part of the input, starting with "S", is a series of searching requests. You need to develop a function to search your AVL tree based on the given name in the INPUT file and print out the answer, which contains the given search name and its phone number. Your printout has to be in exactly the same way as shown in the following OUTPUT file. If a given search name is not found in the tree, then a **"null"** string should be printed in the phone number field as shown in the following OUTPUT file.

A letter "E" will be given at the end of the INPUT file to indicate that it is the end of the INPUT file.

In terms of the format of the output, please refer to the output0_windows.txt in HW6.zip for details.

Input:

The input contains following sections of data.

"D" indicates the start of a number of " name" and "phone number" pairs which are separated by a "Blank". Each such pair is ended by an "Enter".

"S" indicates the start of a number of searches, which are some given names separated by an "Enter".

"E" stands for "End of Input".

Output:

Use the **Pre-Order** traversal to traverse the AVL tree you constructed and print out the names of the visited nodes and separate each name with a "Blank", as shown in the following OUTPUT file.

Then, print out the name and phone number pairs separated by a "Blank" if the given search name is found in your tree. If the given search name is not found in the tree, print out the name followed by a "Blank" and a **"null"** string.

Note:

- We suggest that you use **scanf()** to get the input content and **printf()** to print out the results.
- The format of the output has to be exactly the same as in the following example OUTPUT file. Your answer will be considered incorrect if any redundant spaces or extra lines exist after the last output string.

Execution:

Your homework must use the following command to execute in the **Windows Git Bash terminal**, and your output file format must be the same as that in the given output file. In addition to the given input data files, there will be another two hidden input data files used to grade your homework.

Step 1. gcc -std=c11 ./*.c -o hw6
Step 2. ./hw6.exe < input0_windows.txt > ans_output0_windows.txt
Step 3. diff ./output0_windows.txt ./ans_output0_windows.txt

Input

```
D
Bob 0900000000
Alice 0900000001
Paul 0900000002
David 0900000003
Ruby 0900000004
Charlie 0900000005
S
Paul
Amy
Ruby
E
```

Output

```
David Bob Alice Charlie Paul Ruby
Paul 0900000002
Amy null
Ruby 0900000004
```