

Data Structure Assignment [4]

General information

- Deadline: **2020/11/12 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 zero 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:

Mon. 15:00 - 17:00

Wed. 11:00 – 12:00

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

Binary Search Tree – Deletion

In this assignment, you are required to build a **binary search tree** and delete nodes (i.e., keys) from the tree. After the deletion, you are required to print out the remaining nodes of the binary search tree in **Level Order**.

There will be two rows in the input. The **first row** gives the keys to be built into a binary search tree, which **HAS TO be built according to the order of the keys given in the first row (from left to right)**. All the keys in the data file will be between 1 and 100 (i.e., $1 \leq \text{key} \leq 100$), and there will be **no repeated key values**.

The **second row** of the input gives the keys we are going to delete from the tree. You **HAVE TO delete the keys in the binary search tree according to the given order shown in the second row (from left to right)**. If a deleted key, say X, has two child nodes, then you must **replace X with the node having the smallest key value in X's right subtree**.

After accomplishing the deletion operation, print out the remaining nodes (i.e., keys) of the binary search tree **in Level Order**. **NOTE: ONLY print out the remaining EXISTING nodes. For a non-leaf node, say Y, that has only one child, JUST print out the key of this child. DO NOT print out a NULL to indicate Y's the other non-existing child node. Your answer will be considered incorrect if a NULL is shown in your answer.**

In terms of the format of the output, please refer to the output_windows.txt in HW4.zip for details.

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 hw4`

Step 2. `./hw4.exe < input0_windows.txt > ans_output0_windows.txt`

Step 3. `diff ./output0_windows.txt ./ans_output0_windows.txt`

Note:

- Just use **scanf()** to get the input content and **printf()** to print out the results.
- The format of the output is required to be the same as in the example. Your answer will be considered incorrect if any redundant spaces or extra lines exist after the integers.

Input

```
1 3 5 7 2 4
5 7
```

Output

```
1
3
2
4
```