Society of Software Engineers
350 Practice Practicum 1

Introduction

For this practicum, you will be working on a **Binary Tree**. You will be provided with several files, where you will be providing the functionality noted in headers. The program will be completed in phases, which will match to make targets.

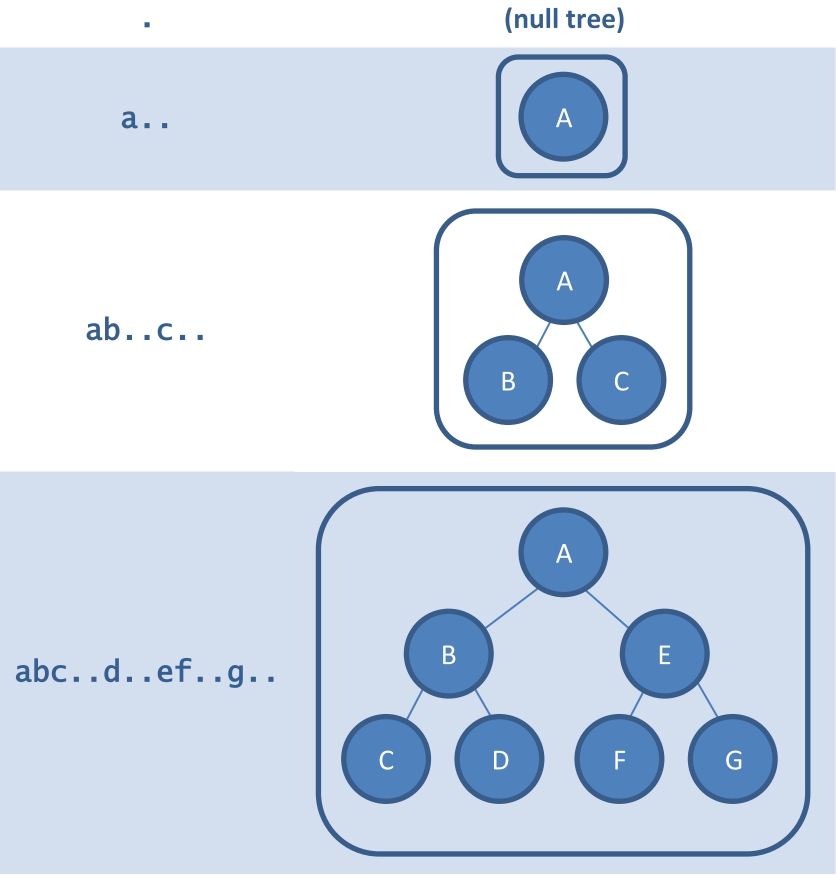
You will be expected to complete the following files:

1. make_tree.c
2. release_tree.c
3. identical.c
4. similar.c
5. parse.c

Program Overview

The program accepts in two lines from standard in, converts them into binary trees and compares the two trees.

The input string is given by the in-order traversal order of a tree. For the purposes of the program, only **alpha characters** will be used to represent node data. The character '**.**' is used to represent a null child.



Society of Software Engineers
350 Practice Practicum 1

The program compares the trees and outputs 1 of 3 different statuses:

1. The trees are **identical**
2. The trees are **similar**
3. The trees are **uncomparable**

Input / Output

```
    $ ./prog
    > abc..d..ef..g..
    > abc..d..ef..g..
    The trees are identical.
```

Steps

At each step, all code other than the file you are directed to edit is provided. For each step, you will functional

1. **make_tree.c**
   a. Complete the code in **make_tree.c**, conforming it to the header of **make_tree.h**
   b. Compile using the command **make prog_make_tree**
2. **release_tree.c**
   a. Complete the code in **release_tree.c** conforming to the header of **release_tree.c**
   b. Comile using the command **make prog_release_tree**
3. **identical.c**
   a. Complete the code in **identical.c** conforming to the header of **identical.c**
   b. Comile using the command **make prog_ identical**
4. **similar.c**
   a. Complete the code in **similar.c** conforming to the header of **similar.c**
   b. Comile using the command **make prog_ similar**
5. **parse.c**
   a. Complete the code in **parse.c** conforming to the header of **parse.c**
   b. Comile using the command **make prog_ parse**

All the elements should integrate together with the target **prog**.

Additional Requirements:

- All code should be properly styled and commented.
- The code should be free from any memory leaks.