COMPUTER SCIENCE 241
Spring 2023

Programming Assignment 7

Due by end-of-day on Thursday, April 27

In this class we have seen how the basic depth-first-search, tree-hugging traversal of a Binary Tree can be written very compactly as a recursive function. By augmenting this basic traversal algorithm, we can perform a variety of different computations on a binary tree. For this assignment you must implement the following binary tree functions.

```
bool  isHeapOrdered ( Node * curr );
int   numSingleChild ( Node * curr );
void  makeMirror ( Node * curr );
```

The provided source code contains a **stub** for each of these functions, together with a description of what the function should do.

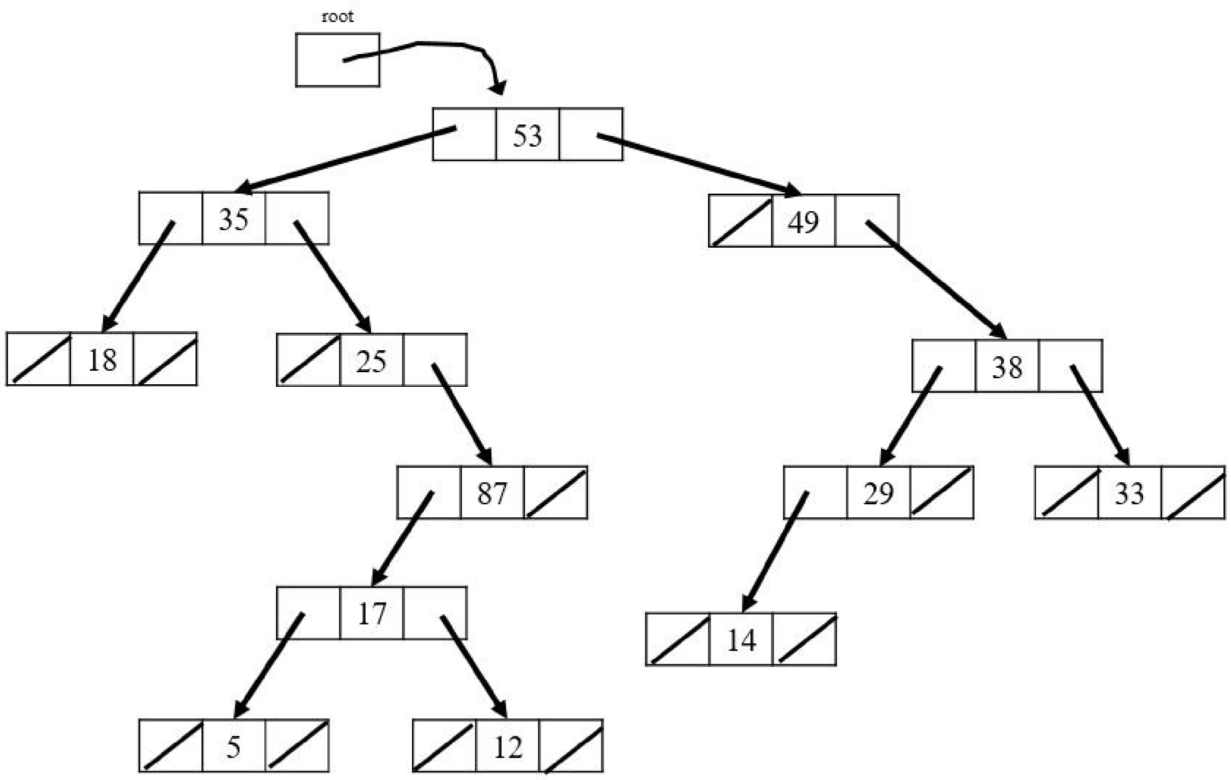
The **isHeapOrdered** function returns true or false depending on whether the binary tree is max-heap-ordered (i.e., if every node's value is greater-than-or-equal-to the values of its children). Note that we are **not** concerned at all about the shape of the tree. A tree can be heap-ordered even if it does not have a "complete" shape.

The **numSingleChild** function returns the number of nodes in the tree that have exactly one child. For example, in the tree below, the function would return the value 4.

The **makeMirror** function alters the shape of the tree (ie, it alters the left and right child pointers) so that the tree is now a mirror-image of the former shape. In other words, the tree has been "flipped" through an imaginary vertical line through the root of the tree.

The given source code contains the three functions **size, height,** and **contains** that we have studied in this class.

The main function creates and initializes the following binary tree that is used as a "test case".



You should implement each of these functions in the **most efficient manner** possible.

You should **modify** the code at: ~joanmlucas/cs241/programs/prog07/handouts/prog07.c You must complete the **body** of each of the required functions. Do **not** change the code anywhere else in the file. Your solution for these functions should be "self-contained". Your program **must not** use any **global** variables. Using global variables is a dangerous habit, since they are visible and modifiable throughout the file.

You should be sure to include **YOUR NAME** in a comment at the top of your source code file. Your source code must use proper **style**, i.e., variables should be well named (name is not too short, not too long, and is meaningful), and bodies of loops, if's, etc. should be properly indented. You should submit your program for grading by copying it into my account using the following command (assuming your name is J. Smith).

```
$ cp   smith_j_prog07.c   ~joanmlucas/cs241/programs/prog07/submission
```

The timestamp of this copied file will indicate when you submitted this assignment. **After** you have copied the file, you must **be sure to set the permissions** on this copied file as follows:

```
$ chmod  644  ~joanmlucas/cs241/programs/prog07/submission/smith_j_prog07.c
```

You can verify that your file was successfully copied using the ls (list) command:

```
$ ls -l  ~joanmlucas/cs241/programs/prog07/submission/smith_j_prog07.c
```