

CSI 402 – Systems Programming – Spring 2012  
Programming Assignment I

**Date given:** Jan. 31, 2012

**Due date:** Feb. 10, 2012

**Weightage:** 5%

The regular deadline for this assignment is **11 PM, Friday, February 10, 2012**. With lateness penalty, the program will be accepted until **11 PM, Sunday, Feb. 12, 2012**. The assignment will *not* be accepted after 11 PM on Sunday, February 12, 2012.

Very important: Your source program must consist of three or more C files and you must also have a **makefile**. (Additional information regarding this requirement is given later in this handout.) The C files (with extension “.c”), header files (with extension “.h”) and the **makefile** must be submitted together using the **turnin-csi402** command. Instructions for using **turnin-csi402** and additional specifications for the **makefile** will be included in the **README** file for this assignment.

---

The total grade for the assignment is 100 points, with 85 points for correctness and 15 points for structure and documentation.

The purpose of this part is to observe the performance of binary search trees built from sets of strings. The necessary information about binary search trees will be presented in class. Since the goal is to get some quantitative information about binary search trees built from collections of strings, your program should *not* carry out any rebalancing operations on the tree.

The executable version of your program for Part (a) must be named **p1**. (Your **makefile** must ensure this.) It will be executed by a command line of the following form:

```
p1  infile  outfile
```

The command line parameters *infile* and *outfile* represent the names of the input and output files respectively. Both of these are text files. Each line of the input file has a string that must be inserted into the binary search tree, which is initially empty. You may assume the following. (There is no need to check these conditions in your program.)

1. There are no blank lines in the input file.
2. Each string in the input file has at least one character and at most 15 characters. Each character is an upper or lower case letter, a digit or underscore ('\_').
3. The strings in the input file are all *distinct*.

Your program should insert all the strings in the input file into the binary search tree in the specified order. (Thus, the first string in the input file becomes the string stored at the root of the binary search tree.) After inserting all the strings, your program should compute and print the following information about the resulting binary search tree **to the output file**.

- (a) The total number of strings in the input file.
- (b) The height of the binary search tree.
- (c) The number of leaves in the binary search tree.
- (d) The height of the left subtree of the root.
- (e) The number of strings in the left subtree of the root.
- (f) The height of the right subtree of the root.
- (g) The number of strings in the right subtree of the root.

You must use the library function `strcmp` (from `<string.h>`) to compare strings and decide their relative order.

Your program must detect the following fatal errors. In each case, your program should produce a suitable error message to `stderr` and stop.

- (1) The number of command line arguments is not equal to three.
- (2) The input or the output file specified on the command line cannot be opened.

**Structural requirements:** Your submission must have *at least three* C source files, zero or more header files and a `makefile`. Additional requirements on the C source files are as follows.

- (a) One source file must contain just the `main` function.
- (b) A second source file must contain only the function that inserts a string into the binary search tree.
- (c) The functions that compute various parameters of the binary search tree (e.g. height of the tree, number of leaves, etc.) must be in one (or more) source files.

---

**Information about README file:** The README file for this assignment will be available by 10 PM on Saturday, February 4, 2012. The name of the file will be `prog1.README` and it will be in the directory `~csi402/public/prog1` on `itsunix.albany.edu`.