Computer Engineering 171
Homework 1: Imperative Programming

Due: April 18th at 9:00 am

The goal of this assignment is to introduce you to writing code in two imperative language designed for different purposes. Use the Linux machines in the Engineering Computing Center for this assignment. The Pascal compiler is `fpc` and the C compiler is `gcc`.

1 Quicksort

On Camino you will find an implementation in Pascal of ***quicksort***, an efficient sorting algorithm. Translate the program to C, keeping the same procedure and function names and parameters, for all but the top-level program, which will become the function `main` in C. Call this program `sort.c` and submit it using Camino.

Goal: To learn about methods of passing parameters and recursion.

Hints: None.

2 Binary Search Trees

A ***binary search tree*** is either empty, or it consists of a node with two binary search trees as subtrees. Each node holds an integer. The elements in a binary search tree are arranged so that smaller elements appear in the left subtree of a node and larger elements appear in the right subtree. On Camino, you fill find an implementation of a binary search tree in C. Translate the program to Pascal, keeping the same function names and parameters, for all but the function `main`, which will become the top-level program in Pascal. Call this program `tree.p` and submit it using Camino.

Goal: To learn about types and data representation.

Hints: You will need to use a record or structure to represent a tree node, and tree nodes need to be dynamically allocated. It is easiest to have both functions be recursive. You will find an example in Pascal of a stack built using a linked list (`stack.p`) on Camino as well.

3 Coding Guidelines

You should not mimic Pascal code in C, nor C code in Pascal, but rather use the best mechanisms provided by each:

- C does not have a boolean type so `int` is typically used instead; Pascal has the built-in type `boolean`.
- Arrays in C always begin at index zero, and although arrays in Pascal can be begin at any index, they traditionally begin at index one.
- C traditionally uses a `#define` for a constant, whereas Pascal uses `const`.
- Since C does not support call-by-reference, a common idiom used to modify a parameter is to pass the parameter to the function by value and return the updated value; Pascal natively supports call-by-reference through `var` parameters, which should only be used for this purpose or for reasons of efficiency. Note that C passes arrays as pointers, but Pascal allows arrays to be passed by value though they typically should be passed by reference for efficiency.

- Pascal allows functions and procedures to be defined within other functions or procedures, thereby limiting their visibility; C does not allow such definitions, but does allow a function to be visible only within the file by using the keyword `static`.
- You can use `typedef` to give a name to an array type of a specified size, allowing the name to be used in multiple places.