CSCB09 2025 Summer – Assignment 2
Due: July 10 midnight
This assignment is worth 10% of the course grade.

In this assignment, you will practice stdio.h binary file I/O in C.

As usual, you should aim for reasonably efficient algorithms and reasonably well-organized, well-factored, comprehensible code.

Code correctness (mostly auto-marking) is worth 90% of the marks; code quality is worth 10%.

Binary Search Tree on Disk

Storing a binary search tree in a file may be impractical (there are better choices such as B-trees), but we will do it anyway for simplicity. In this assignment, the file format consists of 1 or more nodes, each node defined by this struct (bst.h):

```
typedef struct node {
  char key[40];         // key, NUL-terminated C string
  unsigned long count;  // how many nodes in this subtree including self
  long left;            // file position of left child; -1L if none
  long right;           // file position of right child; -1L if none
} node;
```

The root node is always at file position 0. Other nodes may be anywhere else in the file.

File size may be larger than absolutely necessary for the actual number of nodes; unused areas may be anywhere in the file.

Question 0: Dump (1 mark, automarking only)

This question encourages the good habit of writing more code than required, especially code and tools that help you check results and debug. It is worth only token marks because you should do it even if it were worth 0 marks.

Write and hand in bst-dump.c to print out the nodes in a given file. The pathname is given as the 1st command-line argument. The nodes should be printed in in-order. The output format is this for each node:

```
key=\"%s\" pos=%ld count=%lu left=%ld right=%ld\n
```

The value for `pos=` should be the file position of the node itself.

A sample tree file is provided (sample-tree.bin). A sample output file for that tree is provided (sample-dump.txt).

Automarking will use only valid files and existing pathnames. But you should still try to catch and report common errors for your own sanity. (E.g., in your own testing, if you have a typo in the filename, who else will remind you?) Please use stderr for error messages and your own debugging messages.

Question 1: The `count` field (10 marks)

A former student made a mistake when producing tree files! Some of the `count` fields have wrong values. Fortunately, all other fields are done right.

Write and hand in bst-fix-count.c to correct all the `count` fields in a tree file. The pathname is given as the 1st command-line argument.

An easy way to stay within linear time is a post-order traversal. If you have already fixed both children (if they exist), it takes just $O(1)$-time to fix their parent.

A sample tree file with wrong counts is provided (sample-wrong-counts.bin). After fixing, it should be identical to sample-tree.bin.

Automarking will use only valid files and existing pathnames. Please use stderr for error messages and debugging messages, if you have any.

This part with be tested with large files under limited memory, time, and disk space. This should be no worry if you have a good implementation.

End of questions.