

# CIS 213: Foundations of Computer Science—Lab 7

## Trie to Retrieve Words

Outline of activities:

- Discuss Tries
- Discuss characters versus integers
- Discuss Modularity of Trie data structure
- Experiment with the `retrieve` program
- Implement the `print` member function
- Thought question about whether we need the `letter` data member.

### Tries

Tries are trees used to store sets of words in a manner that is fast to *retrieve*. Imagine a dictionary where all the As were in one volume, all the Bs in another, and so forth. Now, imagine that each volume were broken up into books by the second letter. Now imagine each book were broken up into booklets by the third letter. And so forth. Some of the books would be empty, of course, but that's okay.

### Getting Started

Create a `lab7` directory in your `cis213` directory. Goto your `lab7` directory and Copy the file from the lab directory, `~hardnett/pub/cis213/lab7` into your `lab7` directory. The file is:

1. `Trie.cpp`
2. `Trie.h`
3. `driver.cc`
4. `Makefile`

The data structure is defined very simply. Look in `Trie.h` and study the idea. Draw pictures for all the examples.

### Characters versus Integers

In C and C++, a character is really a kind of small integer. This is based on the ASCII character set. Use `man ascii` for a table. Here are some examples:

A	65
B	66
Z	90
a	97
z	122
0	48
9	57
+	43

This means, you can do weird things in your C++ program, such as:

```
char c = 'A';
int x = c;
cout << c << " = " << x << endl;
x = 'A'-5;
c = 'A'+10;
```

What should each of these do?

### Modularity

What things do we want to have public about the Trie data structure? What things should be private? How is this enforced by the code?

### Experiment

Try using the `retrieve` program in the class directory. To keep the trees small, I've limited them to the letters a–g. Here are some words you can use:

a, add, cab, fad, fade, bad, bade, babbage,  
fab, baa, cafe, cage, caged

If you can think of others, let us know.

When you're happy with your experiments, copy the code and compile it. You can change the number of letters to 26 if you want. However, the code I've given you is missing the code for the `print` member function. Work on implementing that.

### Thoughts

While you're implementing, study the code and determine whether we need the `letter` data member at all. Can we get by without it? Would that be a good idea?