# CECS 429/529 - Homework 1

1. Download the file `NaiveIndex.java` from BeachBoard's Homework section. This file builds a simple boolean document-term matrix over a directory of .txt files. For this problem, you will extend the functionality in `NaiveIndex` to support a very simple information retrieval system. Follow these steps:

   (a) Create a new Java project in your favorite IDE and add `NaiveIndex.java` to the project.

   (b) Download the file `MobyDick10Chapters.zip` from BeachBoard. This file contains the first 10 chapters of Herman Melville's *Moby Dick*, each chapter separated into its own `.txt` file. Extract the ZIP file to the **root** of your project's directory.

   (c) **Run** `NaiveIndex.java`. The program should output a large matrix of 1's and 0's as a term-document index. Make sure you understand the output and what it means.

   (d) **Modify the** `main` **method** of `NaiveIndex` to build a simple search engine application:

      i. Remove the call to `index.printResults()`.

      ii. Write a loop that continually scans a word from the user, until they enter the word "quit". After scanning a word, print a list of indexed documents that contain the user's term by walking through the `mIndex` member of the `index` variable in `main`. You will **need** to use a binary search to get the column number of the user's term (see Java documentation for `Arrays.binarySearch()`) from the `mTermArray` variable. Then use a loop to walk through each entry in the corresponding column in `mIndex`, and if you find a `true` at that matrix position, print out the corresponding document ID (since that document contains the requested term).

         A. Program this loop intelligently using good programming practices. Do not, for example, **assume** that there are only 10 documents; write your loop in terms of the array's arbitrary length, whatever that may be.

      iii. Example behavior: (user input in *italics*)

```
Enter a term to search for:  whale
These documents contain that term:
chapter1.txt chapter2.txt chapter3.txt chapter7.txt chapter9.txt

Enter a term to search for:  captain
These documents contain that term:
chapter1.txt chapter7.txt chapter8.txt chapter9.txt

Enter a term to search for:  quit
Bye!
```

   (e) **You should only modify the** `main` **method of the program and nothing else.**