**Homework Guide**

**Summary:**

* All homework problems require you to write a class called **Solution** that has the method   
       **public static void run(BufferedReader in, PrintWriter out) throws IOException**
* All problems require you to run the test client I provided. To run the test client, type in the following at the command line:   
       java TestClient *username number*   
  where *username* consists of the first letter of your first name and the first 7 letters of your last name and where *number* is the problem number you are trying to solve. Note that both TestClient.class and Solution.class must be on your CLASSPATH.
* Your grade for a problem *N* can be found in the table available at the following URL:   
       http://wmarrerowww.cstcis.cti.depaul.edu/csc421/problem*N*.html   
  Use the last 4 digits of your DePaul campus connect ID number to look up your grade. Note that leading zeroes will be missing.
* All programming problems will require your solution to respond to multiple problems. For all of the programs you must follow the following protocol:
  1. Read just enough from the input to get the next problem (or to decide there are no more problems)
  2. Solve the problem and send the output back to the server
  3. Go back to 1 and repeat

In particular, *you cannot read any more input until you solve the current problem and send back the answer*.

**Gotchas:**

* Your homework will be graded by running a test client that connects to my testing server. This requires that your JVM be able to open network connections to my server. Connections failures could be a result of firewalls blocking communication between you and the server (perhaps even on your very own machine). This will most likely be the case if you try running the test client from a machine at work.
* **Don't forget to flush your output!!!** Your output is not actually sent to the server until you call the flush() method on the PrintWriter object. You need to make sure to call flush after writing out your output for each test.
* Don't forget to reload the web pages for your homework grade and/or the log error report every time you run you solution. Your browser is most likely caching the page and you will see information about a prior execution if you forget to reload the pages.
* The protocol for the grading server is simple, but exact. There is no room for error. If you get the protocol wrong, your solution will fail, even if you algorithm is correct.
* Your code did not create the input or output streams. You should not be closing them. If your code closes these streams, this could cause my test client code to crash or the server to timeout.
* Many algorithms require you to use a reasonable data structure. Make sure to choose your data structures based on the operations that need to be supported and on the API provided by Java. That documentation will usually tell you the efficiency/running time of the various Collection classes.
* The web server maintains the error log for your last execution across *all* problems. For example, if you run a solution to problem 3, this will overwrite the last error log you had, even if that was an error log for an execution of problem 2.

**Tentative Homework Problems:**

|  |  |
| --- | --- |
| **Name** | **Due Date** |
| Problem 1 | Week 2 (1/11) |
| Problem 2 | Week 3 (1/18) |
| Problem 3 | Week 3 (1/18) |
| Problem 4 | Week 4 (1/25) |
| Problem 5 | Week 6 (2/8) |
| Midterm | Week 7 (2/15) |
| Problem 6 | Week 9 (3/1) |
| Problem 7 | Week 10 (3/8) |

**Tentative Reading Schedule:**

|  |  |
| --- | --- |
| **Week** | **Sections covered** |
| Week 1 | 1.1, 1.2 |
| Week 2 | 2.1, 2.2, 2.3, 3.1, 3.2, 4.3, 4.4, 4.5 |
| Week 3 | 7.1, Tribbles (Fibonacci sequence), Subsets problem |
| Week 4 | 15.1,15.2, 15.3 |
| Week 5 | Class cancelled due to weather. |
| Week 6 | 15.4, String Edit (Problem 15-5), 16.1 |
| Week 7 | 16.2, 16.3, Arbitrage Problem |
| Week 8 | 22.1, 22.2, 22.3 |
| Week 9 | 22.4, 23.1, 23.2 |
| Week 10 | 24.1, 24.2, 24.3, 24.4 |