COS 126 Midterm 1 Programming Exam, Spring 2008

This part of the exam is like a mini-programming assignment.   You will create a program, compile it, create test data for it and run it on your laptop. Debug it as needed. This exam is open book, open browser. You may use code from your assignments or code found on the COS126 website. When you are done, submit it via the course website using the Social Activities link called Precept Exam 1.

 Put your name, login ID, and precept number on this page (*now*), and write out and sign the Honor Code pledge before turning in this paper. Note: It is a violation of the Honor Code to discuss this midterm exam question with anyone until after everyone in the class has taken the exam. You have 50 minutes to complete the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

_____

                                               Signature

```
                    /25
   1
```

Description of Program (25 points)

Define the "similarity" of two strings `a` and `b` as the number of indices `i` for which `a.charAt(i)` and `b.charAt(i)` are the same. Write a Java program `Match` that reads strings from <u>standard input</u> (one string per line) and prints one string whose similarity with the command-line argument is **maximal.** Also print the similarity number. If there is more than one input with the same maximal similarity number, print the first one found. For example, if the file `tiny.txt` contains

```
actgcta
ctgactggaa
acacaaactt
acacaatgtgtg
actacat
```

then you should get the following results:

```
% java Match actactg < tiny.txt
actgcta 5

% java Match cta < tiny.txt
ctgactggaa 2
```

Organize your program as a static method `similarity()` that takes two strings as arguments and returns an `int`, and a `main()` method that uses `similarity()` to compare each string on standard input with the command-line argument, keeps track of the **<u>best</u>** match seen so far, and prints the result. You will need to have `StdIn.java` or `StdIn.class` in your directory in order to compile and test your program.

Remember to submit your program `Match.java` on the course website.

Description of Program (25 points)

Define the "similarity" of two strings `a` and `b` as the number of indices `i` for which `a.charAt(i)` and `b.charAt(i)` are the same. Write a Java program `Match` that reads strings from <u>standard input</u> (one string per line) and prints one string whose similarity with the command-line argument is **minimal.**   Also print the similarity number. If there is more than one input with the same minimal similarity number, print the first one found. For example, if the file `tiny.txt` contains

```
actgcta
ctgactggaa
acacaaactt
acacaatgtgtg
actacat
```

then you should get the following results:

```
% java Match actactg < tiny.txt
acacaaactt 2

% java Match cta < tiny.txt
actgcta 0
```

Organize your program as a static method `similarity()` that takes two strings as arguments and returns an `int`, and a `main()` method that uses `similarity()` to compare each string on standard input with the command-line argument, keeps track of the **<u>worst</u>** match seen so far, and prints the result. You will need to have `StdIn.java` or `StdIn.class` in your directory in order to compile and test your program.

Remember to submit your program `Match.java` on the course website.