

COS126 Spring08 Programming Exam 1 (best match version)

```
// Spring 08 Programming Exam 1
// Best Match version

public class Match {

    // returns number of indices where a and b
    // have same character in same position
    public static int similarity(String a, String b) {
        // shortest String determines loop condition
        int N = Math.min(a.length(), b.length());
        int matches = 0;
        for (int i = 0; i < N; i++) {
            if (a.charAt(i) == b.charAt(i)) matches++;
        }
        return matches;
    }

    public static void main(String[] args) {
        String a = args[0];
        // number of matched characters can't be lower than zero
        int best = -1;
        String bestMatch = "";

        // read strings from standard input
        // remember best match found so far and its similarity number
        while (!StdIn.isEmpty()) {
            String b = StdIn.readString();
            int sim = similarity(a, b);
            if (sim > best) {
                best = sim;
                bestMatch = b;
            }
        }

        // output best match and its similarity
        StdOut.print(bestMatch);
        StdOut.print(" ");
        StdOut.print(best);

    }
}
```

COS126 Spring08 Programming Exam 1 (worst match version)

```
// Spring 08 Programming Exam 1
// Worst Match version

public class Match {

    // returns number of indices where a and b
    // have same character in same position
    public static int similarity(String a, String b) {
        // shortest String determines loop condition
        int N = Math.min(a.length(), b.length());
        int matches = 0;
        for (int i = 0; i < N; i++) {
            if (a.charAt(i) == b.charAt(i)) matches++;
        }
        return matches;
    }

    public static void main(String[] args) {
        String a = args[0];
        // number of matched characters can't be higher than a.length()
        int worst = a.length() + 1;
        String worstMatch = "";

        // read strings from standard input
        // remember worst match found so far and its similarity number
        while (!StdIn.isEmpty()) {
            String b = StdIn.readString();
            int sim = similarity(a, b);
            if (sim < worst) {
                worst = sim;
                worstMatch = b;
            }
        }

        // output worst match and its similarity
        StdOut.print(worstMatch);
        StdOut.print(" ");
        StdOut.print(worst);

    }
}
```