

CS 350-Spring 2005

Project 2 Requirements

v. 2/9/05

Purpose: Extend assignment 1 in two ways:

1. Test more than 2 programs
2. Count the number of matches and mismatches, rather than using diff.

Parameters and Sample Use:

As before, the program should take several parameters passed on the command line:

1. the number of programs to be tested,
2. the base name (see below) for the set of programs, and
3. the name of the input data file.

You should assume that the programs to be tested all start with the same letters, but end with a two digit integer starting with "01".

Assume your program is called "tester2," and the executables to be tested are "vers01," "vers02," "vers03," "vers04," "vers05," and the test data file is "testfile." Then your program could be run with:

```
./tester2 5 vers testfile > TestReport
```

Error handling:

1. Your program should check for the correct number of parameters
 2. Your program should check that all of the files passed really exist.
 3. Your program should check that the first parameter is an integer.
- For each error, your program should generate a reasonable error message and terminate.

Counting Mismatches:

You should assume that all of the programs being tested produce only numeric output (this is to simplify your programming effort). After your tester has run all of the programs, it should then compare the first number produced by each of the programs and count mismatches, then compare the second value of each program, etc. until it has compared all values. Your program should then report the number of mismatches of each of the programs with the output of program 1, and the total number of mismatches.

General Comments:

1. You must use the submit command to submit your code. See the handout on the class web site for how to use this command.
2. You must test your program thoroughly.
You must design your program for reuse.
3. This assignment involves both program creation and program testing. If your test data are inadequate, you will lose points on the assignment.

Test Data:

You must submit several test data sets. Each test data set should consist of several files, executable versions of the programs to be tested, and two files: input file for each program being tested should read and an expected output file. This file contains what a correct tester should produce when it tests the provided programs with the provided test data.

Other Requirements:

1. You must also submit a README file which includes the following:
 - a. Information on what this code does.
 - b. Instructions about compiling your code.
 - c. Any other special information which a user of your code should know.In general, the grader will just scan the file for completeness. Because of the size of this class, the grader will NOT do anything special to compile and run your code (see below). The README file is included as a standard convention for documenting code for other programmers; it is not instructions to the grader on how to compile or run your code.
2. The grader will compile and run your programs. It is your responsibility to make this easy for the grader. The grader will not debug your code and because of the number of programs to grade, the grader will compile your code with the following command:

```
g++ -o prog1 -lm <Source file>
```

And then test your submission with his own test programs and data.

Other Information:

Check the class web site for due dates, grading criteria and required PSP forms.