

CS 2213-001 Advanced Programming

Instructor [Dr. Turgay Korkmaz](#)

Homework 7

Due date: check BB

!!!! NO LATE HOMEWORK WILL BE ACCEPTED !!!

Total 5 points

(Command-line Arguments, Files, Abstract Data Type - Library)

Programming Exercise 8 from Chapter 8...

8. Probably because solving problems by computer can generate such intense frustration, computer science courses seem to generate more than their share of plagiarism. In several universities, the situation has gotten so bad that computer science departments have had to develop software to help detect cases of academic misconduct. The usual approach taken in such programs is to compare the structure of two programs, ignoring differences that are easy for students to change, such as the names of variables and procedures.

Consider, for example, the two program fragments shown below, each of which sums the elements in an integer array.

student1.dat

```
int Total(int array[], int n)
{
    int i, sum;

    sum = 0;
    for (i = 0; i < n; i++) {
        sum += array[i];
    }
    return (total);
}
```

student2.dat

```
int Sum(int list[], int nl)
{
    int j, total;

    total = 0;
    for (j = 0; j < nl; j++) {
        total += list[j];
    }
    return (total);
}
```

The names of the functions and many of the variables are different, but the two programs are otherwise exactly the same. In code samples this short, it is entirely possible that the programs were created independently, but one would start to get suspicious if the structural similarity went on for page after page.

Write a program that uses two instances of the **scannerADT** to perform a line-by-line comparison of two input files. The output of the program should be the percentage of lines in the files that "match." Two lines are defined as matching if their corresponding tokens match all the way across. Two tokens match if either of the following is true:

- The tokens are the same string.
- The tokens both begin with a letter.

For example, the tokens **"sum"** and **"total"** match because both begin with a letter. In the **student1.dat** and **student2.dat** sample files, every line matches perfectly under this definition, so the program should report that 100 percent of the lines match.

In Programming Exercise 8, you are simply asked to implement a driver/client program using the scannerADT library in Section 8.5. So you should first study Section 8.5 and copy `scanadt.h`, `scanadt.c` and `scantest.c` files from the class web page (follow the link “programs from the textbook” and then go to 08-Abstract-Data-Types).

After compiling and running `scantest` program, implement your `driver.c`. Your program should take two filenames as command-line arguments, open these files and compare them line by line as defined above. So we will run your program as follows

```
main212> driver student1.dat student2.dat
```

You can simply create the above data files in your current directory.

As always, make sure you release (free) the dynamically allocated memories if you allocate any memory in your programs. So, before submitting your program, run it with `valgrind` to see if there is any memory leakage... Also if you need to debug your program, compile your programs with `-g` option and then run it with `gdb` and/or `ddd`.

What to return: !!!! NO LATE HOMEWORK WILL BE ACCEPTED !!!

1. Create a directory, say `LASTNAME_hw7`, and do all your work under that directory.
 2. You will get scannerADT library (`scanadt.h` and `scanadt.c`) from the textbook and use it along with other libraries in your driver/client program, `driver.c`.
 3. To easily compile the library and driver program, you must have a `Makefile` and use “make” to compile your code.
 4. After compiling, run your program a few times with different data files that you should create and save the output (using script) into `output.txt` file. So you will have around 6-7 files in your `LASTNAME_hw6` directory.
 5. Go to parent directory of `LASTNAME_hw7`, and use

```
> tar -cf LASTNAME_hw7.tar LASTNAME_hw7
```

This will create a new file called `LASTNAME_hw7.tar` and it contains all of your files. So just submit this `.tar` file.
 6. Go to WebCT (BB), and just submit `LASTNAME_hw7.tar` as **attachment** before the deadline. DO NOT submit other `.h` or `.c` files individually.
- /* Don't forget to include comments about the problem, yourself and each major step in your program! */*

You must submit your work using Blackboard Learn and respect the following rules:

- 1) All assignments must be submitted as either a zip or tar archive file unless it is a single pdf file.

- 2) Assignments must include all source code.
 - 3) Assignments must include an output.txt file which demonstrates the final test output run by the student.
 - 4) If your assignment does not run/compile, the output.txt file should include an explanation of what was accomplished, what the error message was that prevented the student from finishing the assignment and what the student BELIEVES to be the underlying cause of the error.
-
