CSI 333 – Programming at the Hardware-Software Interface – Fall 2017  
Project II

The total grade for the assignment is 100 points.

You must follow the programming and documentation guidelines available in the Blackboard module Projects.

# Description:

You are required to write a C program

* whose input is an existing text file with a name specified in a command line as the first argument,
* whose output is a new text file with a name specified in a command line as the second argument; requirements to the new text file are given below.

Input and output files should have extension .txt but only names of the files are specified in a command line.

Example: suppose the executable version of your program is named p2.out. The program will be executed by a command line of the following form

p2.out inputfile outputfile

In this case file inputfile.txt must exist, outputfile.txt will be created.

Requirements for the output text file:

The output file should have copy of each line of the input file in the same order of lines. Each line of the output file should consist of the same words as the corresponding line in the input file but ordered backward.

In the input file words are separated by one or more spaces (' ') or by horizontal tab ('\t'). In the output file words should be separated by single space.

In the input and output files, each line ends with new line character (‘\n’). Lines do not exceed 255 characters including new line character.

The outline of your C program is as follows.

1. Read command line arguments and check the correctness.
2. Create a new file named second argument + “.txt”.
3. Proceed with converting input file lines into output file lines.
4. Stop and print out "Finished." once finish reading and writing the files.

Thus, each time your program is executed, it should handle just one input file.

Suggestions: Use fflush(stdout) after each call to printf.

# Electronic Submission:

There are two mandatory submissions for each project. Both submissions are the same file but to be used for different purposes. You may do both submissions at one go:

* source code for the evaluation using turnin-csi333 on the ITS Unix machine,
* source code to keep your records – the standard Blackboard procedure for the assignments.

Important Notes: ignoring any of the following rules will result in penalty or even ZERO grade for the project.

* 1. For Project 2 you must turn in the file **named** “**p2.c”**.
  2. At the top of each of your C source file the following information must appear in the form of comments:
  3. your name,
  4. your Unix login ID,
  5. the name of your lab instructor and
  6. the day and time of your lab class.
  7. Make sure that your programs compile and produce correct results on the Unix machines (itsunix.albany.edu) supported by Information Technology Services (ITS) unit of UAlbany. Programs that cause compiler or linker errors on the ITS Unix machines will NOT receive any credit.
  8. Using the turnin-csi333 program as discussed below is the ONLY acceptable way of submitting programming assignments in this course. You should NOT email the files to the instructor or to the TAs.
  9. Remember that you must submit only your C source files. DON'T turn in unnecessary files (e.g. object files with extension “.o” created by compiling C source files, executable files such as “a.out”, etc.).

To submit your files electronically, you must have the source files on one of the ITS Unix machines. For this project, the file p2.c must be in your working directory and you must be logged on to one of those machines to actually carry out the electronic submission.

To perform submission, you should type the following command to the Unix operating system:

turnin-csi333 -c csi333 -p hw2 p2.c

After you issue the above command, the system responds with:

The sections of csi333 are:

WE\_0920

WE\_0415

FR\_1130

FR\_0140

MO\_1025

Enter your section:

Depending upon the day and time of your weekly discussion section, you would type the appropriate section. For example, if your discussion class meets on Wednesdays at 09:20 AM, you would type WE\_0920 followed by the return key. The system will then respond with

Your files have been submitted to csi333, hw2 for grading.

In the above message, "hw2" refers to the name of the project that is currently active.

If you submit your program during two days after due date the name of the project should be "hw2-late":

turnin-csi333 -c csi333 -p hw2-late p2.c

Lateness penalty is 10 points per day. Attempts to submit the program after the two-day grace period will fail.

Additional information about the turnin program:

* 1. If you use the turnin command above again at a later time (before the deadline), then the files submitted previously would be completely replaced by the newly submitted files.
  2. At any time, you can obtain the names of the files that you have submitted to the current project using the following command:

turnin-csi333 -c csi333 -v

* 1. You cannot submit your work if the project is closed. The project is starting from the date the homework is on Bb till the due date. To see the list of open/closed projects, use the following command:  
     turnin-csi333 -c csi333 -l

# Some sample data to test your program:

Important Note: Some sample inputs that can be used to test your programs are given below. However, you should remember that when we compile and run your source files, we will use other data. Just because your programs work for the sample inputs given below, you shouldn't assume that they will work for all inputs. Therefore, you should test your programs thoroughly with other input files.

|  |  |
| --- | --- |
| **Input File** | **Output File** |
| Beside an assault scenario.  When will a friendly cube?  The precedent.  This percent abides without an arm.  The recipe multiplies.  The provoking football. | scenario. assault an Beside  cube? friendly a will When  precedent. The  arm. an without abides percent This  multiplies. recipe The  football. provoking The |

# Program Grading:

Programs will be graded using a script written by the TAs. The script will compile your source program, generate the executable version and run the executable on new test data. The TAs will grade the version that you submit; once the submission is closed, you won’t be allowed to make any changes to your program.

Points: 90 points for correctness and 10 points for structure/documentation.

# Example of program execution:

The following examples assume that the executable version of the program is in the file p2.out.

unix2> p2.out file1 file2

Finished.

unix2>