CSE230

HOMEWORK - SPRING 2015

HOMEWORK 3 - due Tuesday, March 24th no later than 5:00PM

For this assignment you will be required to write a "Grader" program which accepts the paths of a C source file, an input text file, and an expected result file. Your task is to compile the C source file by using gcc (or any other C compiler of your choice) through the system() function and then run the resulting program with the given input file. You will then compare the output file generated by the execution of this program with the expected result file to determine whether or not the program is correct.

Your program must also follow these requirements:

- **You must prompt the user for the name of the source file, input file, and expected result file which the user will then enter by using the keyboard**
- **The actual output file path will be predetermined by your program (do not prompt the user for this)**
- **The number of lines in the output will have a fixed number of lines (5 lines) that should be defined as a constant with an appropriate #define statement**
- **You may compare the output file with the expected result file by using a string comparison library function on the contents of each file**
- **If there is an error reading from the files or if the C program has errors during compilation (indicated by a non-zero value returned by the system function) you must inform the user**
- **Your program should make good use of functions and be well commented**
- **You may implement this assignment by using either C or C++**

Here is a sample usage of the system function used to compile a source file named myProg.c and to run the resulting program with an input file called input.txt:

```
system("gcc -o myProg myProg.c");
system("./myProg input.txt > output.txt");
```

EXTRA CREDIT (Optional - Maximum 25 points):

- **Allow your program to generate a numerical grade for each assignment based upon what percentage of the input is correct. As long as the grading system you devise is reasonable, you may handle the details of this on your own.**
- **Improve the user interface for the grader so that he or she is capable of entering multiple source files without needing to re-enter the names of the input and expected result files.**
- **You may make your program capable of handling expected output files of variable length (not just limited to 5 lines).**

SAMPLE INPUT AND OUTPUT

Note that the input file is in green and the output follows in blue:

```
//Sample source code, "myProg1.c"

#include <stdio.h>
```

```
#define  MAX_VALUES    3
#define  OUTPUT_LINES   5

int main(int argc, char **argv)
{
    int values[MAX_VALUES];
    int i, j;
    FILE *inputFile;
    if ( (inputFile = fopen(argv[1], "r") ) == NULL) {
         printf("Error opening input file.\n\n");
         exit(1);
    }
    for(i = 0; i < MAX_VALUES; i++)
        fscanf(inputFile, "%d", &values[i]);
    for(i = 0; i < OUTPUT_LINES; i++){
       for (j=0; j < MAX_VALUES; j++)
          printf("%d ", values[j]*(i+1) + j);
       printf("\n");
    }
    return 0;
}

//Sample source code, "myProg2.c"

#include <stdio.h>
#define  MAX_VALUES    3
#define  OUTPUT_LINES   5

int main(int argc, char **argv)
{
    int values[MAX_VALUES];
    int i, j;
    FILE *inputFile;
    if ( (inputFile = fopen(argv[1], "r") ) == NULL) {
         printf("Error opening input file.\n\n");
         exit(1);
    }
    for(i = 0; i < MAX_VALUES; i++)
        fscanf(inputFile, "%d", &values[i]);
    for(i = 0; i < OUTPUT_LINES; i++){
       for (j=0; j < MAX_VALUES; j++)
          printf("%d ", values[j]*(i+1) + (j < 2 ? j: 3));
       printf("\n");
    }
    return 0;
}

//Input file "input.txt"
4 6 8

//Output generated by "myProg1.c"
4 7 10
8 13 18
12 19 26
16 25 34
20 31 42

//Output generated by "myProg2.c"
4 7 11
8 13 19
12 19 27
16 25 35
20 31 43

//expected result file "results.txt"
```

```
4 7 10
8 13 18
12 19 26
16 25 34
20 31 42

//Two program executions

Please enter the name of the source file: myProg1.c
Please enter the name of the input file: input.txt
Please enter the name of the expected result file: results.txt

The output of the program is correct.

//The extra credit would give a grade of 100

Please enter the name of the source file: myProg2.c
Please enter the name of the input file: input.txt
Please enter the name of the expected result file: results.txt

The output of the program is not correct.

//The extra credit might give a grade of an 67 or so
```

GRADING KEY

- **(5 pts): Proper use of Comments (Include your name, assignment number and a brief description of the program at the top of main program, and brief description of each function. Also, include name of the C compiler that you have used).**
- **(5 pts): Good use of Functions**
- **(5 pts): Use of Multiple Source Files**
- **(5 pts): Inclusion of a makefile**
- **(5 pts): File names are read correctly from keyboard**
- **(5 pts): Appropriate #define statements are included**
- **(10 pts): Error message displayed when a problem reading information from a file or compilation has been encountered**
- **(20 pts): The system function correctly is used to compile the source program**
- **(20 pts): The output of the compiled program is correctly generated**
- **(20 pts): The generated output file is correctly compared with the expected result file**
- **EXTRA CREDIT (Optional - Maximum 25 points):**
  - **(up to 20 pts): The program is numerically graded based upon the amount of the grade that is correct**
  - **(7 pts): The user interface allows multiple source files per input and expected result file**
  - **(10 pts): The program handles outputs of a variable number of lines**

SUBMISSION INFO

1. **Create the necessary source files and label them accordingly. For example, hw3.c, etc.**
2. **As part of a comment at the beginning of each file, include your full name, Stony Brook Solar ID# and your email address. Also, include a brief description of the program.**
3. **Login to your grading account and click "Submit Assignment" to upload and submit all files of your assignment. Note, if you're taking more than one course with me, your username is different for each course.**