CSE 3430, Assignment #3/Lab1
Due: April 11, 2016

Two points about the homework 3 that I assigned:

First, some of you indicated that you had a hard time with the second problem on homework 3 or were not able to do it at all.

Second, I tried reading through some of the homeworks you handed in last week and I realized that given that I am not a very good C compiler:-), it was a bad idea to have you just give me the code without necessarily testing them. I imagine the grader would really hate me if I asked him to grade them! I should have asked you to get your code debugged and properly running on stdlinux and then submit them online.

So I would like to do the following. We will change that homework (revised slightly, see below) to a programming lab that is due at 11:59 pm on Monday, 4/11. It should be submitted by that time using the "submit" command as follows:

```
submit c3430aa lab1 histogram.c tail10.c
```

where `histogram.c` and `tail10.c` are the two files in which you have your complete source code for the two problems respectively. (See '`man submit`' if you have not used the `submit` command before.)

1. (20 points). (1-14): Write a program that reads input from the standard input file (stdin), one line at a time. For each line, the program should count the number of occurrences of each *alphabetic* (i.e., 'A' through 'Z') characters, treating uppercase and lowercase characters equally; i.e., an occurrence of 'a' should be considered equivalent as an occurrence of 'A'. Your program should make effective use of the *isalpha*() and the *toupper*() functions from the standard library. When you reach end-of-file, print, to stdout, a histogram of the different characters by outputting a row of '*'s to represent the number of occurrences of each character with each '*' representing ten occurences (or fraction of ten for the last '*'). If a given character does not appear in the input stream at all, print a '?' against that character. So the output from your program should consist of 26 lines, one for each character of the alphabet.

2. (30 points). (5-13): Write a program `tail`, which prints out, on stdout, the last 10 lines of the input from stdin. You should not use a two-dimensional array of fixed size. You may, e.g., use the following approach: Have an array `char *X[10]`; note that this only allocates space for the pointers, not for the actual character strings. Read in read in lines from stdin using `getline()` and store them in consecutive elements of the array (wrapping around the array). As you read in each line, you will have to copy it into in the next location in the `X[]` array; but first you have to use `alloc()` to allocate needed space for `X[k]` before using `strcpy()` to do the copying; but even before that you will have to *release* the space (if any) currently allocated to `X[k]` by using `free()`.

   When you finally reach end-of-file, you will have to print, on stdout, the last ten lines with the last line of the file appearing as the last line of your output. Once you do that, you should use `free()` to release the space occuped by the lines. Make sure your code works correctly even if the input contains *fewer* than ten lines.

   Puzzle: suppose the problem required you to print out the *longest* ten lines of the input rather than the last ten lines, how would you change your program? (Don't submit this; just think about it.)