HW2 - C programming #1

Due 2/8/17

For this assignment, you will write 2 programs in C, and compile and run them from the command line.

Program 1: simple statistics

Write a C program that prompts the user to enter an integer indicating the count of numeric values to follow. After this, read in *count* double values and load them into an array. Use `printf` and `scanf` for IO.

Calculate and print out the min, max, average, median and mode of the numeric values entered.

Make sure to properly free array memory before the program exits.

I suggest coding your own insertion sort to sort the input as it is entered. This will facilitate the calculation of median and mode.

Program 2: random text square

Write a C program that takes two numeric command line arguments: `width` and `height`. This program should print out a block of random **text letters** with the prescribed dimensions: `width` characters per line, `height` lines.

Command line parameters can be accessed via in the `char*[]` array that is the formal parameter for the `main` subroutine. (Remember, `char*` is the datatype we use for strings, so this parameter effectively stores an array of strings – you will need to convert these to `int`).

This program should allocates a `char` array of the appropriate size – ((width+1)*height+1) to store all random characters, newlines and the NULL at the end. Then it should fill it randomly, as appropriate, print out the buffer and free up the allocated array. Make sure to NULL-terminate the string.

Note, the `char` literal '\n' can be used to include newlines within a NULL-terminated string. You must load an array for full credit. Don't just print out random text.

Tips

- Make sure to compile using the `-Wall` flag to alert you to all compiler warnings.
- You'll need to look up documentation for different functions and subroutines. Here are some that you may find helpful:
  - `atoi` – converts a NULL-terminated string to an `int`
  - `rand` / `srand` – generate pseudorandom numbers / seed the random number generator (seeding is pretty interesting!)
- Ask me for help! Start early enough to have time to get the needed help.
- Make sure you are doing your own work, and not using code from the internet. Turning in code that you did not write yourself is a violation of the honor code and you will be referred to the UTC Honor Court.

Turn it in.

Put all C files in a zip file and submit to blackboard. Code must compile to be eligible for points. Include screenshots illustrating the programs in action.