EC327 – "Introduction to Software Engineering"

Lab7 – Pointers and Arrays

Mo, 10/26/15 – Fri, 10/30/15

In Lab7, **we define three problems** (P1, P2, P3) of similar complexity. To pass Lab7, **must complete all problems**. You **ARE** required to solve all three problems!

The problems' solutions focus on the practical utilization of C++ **Makefiles**, **GDB, Pointers and Arrays.**

In your designated Lab hours you should demonstrate and explain your solution approach for the problems. You are not required to work outside Lab hours! We recommend starting to work early. For example, you can solve the problems at home, come to your Lab hours, demonstrate your solution, and then you're done. But, ensure that your code compiles and executes properly in the Lab! You must be in lab for at least an hour to get credit for the lab section. The three problems should take around an hour and you may utilize the second hour for PA2 questions or questions on the midterm you just took.

Read carefully the problem definitions. **Decide for yourself** when to start. And please **do not hesitate to ask questions**!

ENJOY!

PROBLEM #1: Learning about Makefiles!

Make is a Unix tool to simplify building program executables from many modules, make reads in rules (as a list of target entries) from a user created Makefile.

Please look at the slideshow and read through the informational links provided about Makefiles.

Then open *pointerDemo.cpp, print.h, print.cpp, reverse.h, reverse.cpp* and the *Makefile*.

You will need to add commands in the Makefile to compile the reverse function. You will have to write the functionality for the functions in *print.cpp* and *reverse.cpp*.

Problem #2: Arrays and Sorting with Pointers

Once you are comfortable with the functionality of the files, make another file *bubbleSorting.cpp* (with its appropriate header file), that defines

void bubbleSort(int * intArray, int size)

void bubbleSort(char * charArray, int size)

Which sorts an array in ascending order, using the bubble sort method.

Once those functions are defined, add functionality to the main() function that creates arrays of different lengths with cout statements that show the original arrays, and the arrays after sorting. You will also need to add commands to the Makefile to compile this file.

Problem #3: GDB, The GNU Debugger

As you may or may not know, debugging code with print statements that show the variable values at different steps sometimes is not the best way to find problems in your code. The GNU Debugger (GDB) is the standard debugger for the GNU software system (which is what you've been using on the Unix machines to compile your code).

GDB can do four main things to help find bugs:

1) Start the program, specifying anything that might affect its behavior.
2) Make your program stop at specified conditions.
3) Examine what has happened when your program has stopped.
4) Change things in your program, and see the effect of correcting a bug.

For this problem, read through the *inclass_lab7.pdf*, implement *test.cpp* (found in the documentation) and experiment with piping the output of the terminal to a file called *outputFile.txt*.