CISC105 Fall 2006 Lab12

- Write a program for each of the following problems. Be sure to save every separate program. All programs must be properly commented and indented (see Assignment Standards on the class website).
- Every semester I watch as scores of C students type big chunks of code into computers and then spend hours or days trying to debug it. If you take one programming habit away from this course, I hope it is to program something short and simple, then test it thoroughly before adding one or two additional lines of code.
- Name each program lab12.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab12.1.c. Put the files in some random directory because your TA won't be looking at them.

Programs

1. Use your Dog struct from lab 12. Dynamically allocate 2 instances of a Dog in main(), then give them values for each member. Write a function that can take a Dog pointer as an argument and print the Dog.

   Consider the print function prototype:

   ```
   void printDog(Dog * dp);
   ```

   While this works, and is nice because it only passes an address, not the all of the values in a whole struct, it is considered bad style because the function should not need to change a Dog, and yet it can. See page 387 of your text for a nice way to prevent the function from accidentally changing the Dog it is supposed to print.

2. Note: the array you make for this problem is not the same type as the array from lab 11 or project 3.

   Declare an array of pointers to Dogs, as follows:

   ```
   Dog * pack[10];
   ```

   This array does not contain any dogs, only room for the addresses of Dogs. Use a loop and malloc to set each address in the array to point to a Dog you make on the heap. Then use another loop to allow the user to input values for each dog. After all Dogs are initialized, print the Dogs.

3. Declare an array of 10 pointers to type char (similar to the argv array used by main()). Read in ten words from the user, and set each array pointer to the address of a char array you allocate on the heap with malloc. Make sure the space on the heap is big enough for the final character in the string!. Copy each user word into thee space you made for it. After all words are in, print them out.

You should have a total of 3 programs named lab12.1.c to lab12.3.c. Make a single script file where you show the result of compiling and running an empty file.

Submit nothing *and* your script of nothing on WebCT, and give the paper version of the complete script file **only** to your TA at the beginning of your next lab (all Friday labs) or in lecture Friday (Wednesday labs **only**). Note: Cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.