Homework #5

Due: Tuesday, November 25, 1997, at 11:59PM
Points: 110

The UNIX Environment

Submitting: Put your answer into a file called *Answers* named as indicated and submit them as described in the **All About Homework** handout. Remember, we'll read it on the DECs, so be sure they work there! (**Note**: if you use ther file names, we will deduct 5 points.)

1. (*10* points) How would you print the date and time in the form

   ```
   Friday, May 13 1994 at 01:05AM PDT
   ```

   Give the UNIX command to do this.

The C Programming Language

Submitting: Put your program into files named as indicated and submit them as described in the **All About Homework** handout. Remember, we'll compile and run them on the DECs, so be sure they work there! (**Note**: if you use other file names, we will deduct 5 points.)

2. (*50 points*) Write a program called `choose` that prints all permutations of *m* numbers chosen from the set 1 ... *n*. Your program is to take two arguments, the first of which is m and the second, *n*. For example,

   ```
   choose 2 3
   ```

   would print

   ```
   1 2
   1 3
   2 1
   2 3
   3 1
   3 2
   ```

   (it may print these lines in any order). Put this into a file called *choose.c*. Remember to look for invalid command line numbers! (Both numbers must be positive, and $m \leq n$.)

   Hint: use recursion.

3. (*50 points*) The function *dyngets* is like the function *fgets*(3) except that it reads lines of arbitrary length, and returns a pointer to dynamically allocated memory containing the current input line. It is useful when the input line can be very, very long.

   A copy of this function is available. Unfortunately, it does not work right. Find, and fix, the error(s). Submit the corrected routine in a file called *dyngets.c*.

   To get this function, go to the directory *~cs40/hw5* in the CSIF. In that directory are the files *dynget.c* (the buggy source code) and *dyngets.man* (a manual page; it describes the arguments and the use of the function). The manual page's description of the function is what the debugged routine should conform to. Also present is a program *testdg.c* that can be used to test your *dyngets* function; it too has a manual page and source code in that directory.