Homework #1

Due: Tuesday, October 7, 1996, at 6:00 PM
Points: 130

The UNIX Environment

Submitting: Type your answers into a file called *Answers*, and submit it as described in the **All About Homework** handout. Remember, we'll read it on the DECs, so be sure it can be read on them! (**Note**: if you use another file name, we will deduct 5 points.)

The purpose of these questions is to have you use the UNIX system and environment.

1. (*10 points*) Type the following command, replacing *yourname* with your login name:

   ```
   ypcat passwd | grep ^yourname
   ```

   Show the output of this command. Notice the ':'s in the output. The 13 characters between the first and second colon are your encoded password; the UNIX system uses it when you log in. Now, type

   ```
   man yppasswd
   ```

   to learn how to change your password. Then change it — and remember to avoid pronounceable words, your initials, friends' names, and other obvious choices.

   Finally, wait at least 15 minutes, and retype the command

   ```
   ypcat passwd | grep ^yourname
   ```

   How does this differ from the previous output?

   Reminder: Be sure you change your password by Monday of the second week, or the system staff will disable your account, and you will need to go to them to get it reinstated!

2. (*10 points*) What option should you give to *rm*(1) to make it ask you verify that each file is to be deleted? Is it possible to restore a file that has been removed?

3. (*10 points*) Send a letter to the class account *cs40* from your CSIF account. The letter is to contain one line in the body, and must look *exactly* like:

   ```
   name<tab>student id number<tab>CSIF account name
   ```

   Note that `<tab>` means a tab character. Do ***not*** put any spaces around the tabs! Your name should begin in the first column in the line. (Do not put anything in your *Answers* file for this exercise.)

The C Programming Language

Submitting: Put your program into a file called *main.c*, and submit it as described in the **All About Homework** handout. Remember, we'llcompile and run it on the DECs, so be sure it works there! (**Note**: if you use another file name, we will deduct 5 points.)

(*100* points) The purpose of this question is to get your feet wet with respect to programming in C and the C compiler. You also will need to use an input function, a control loop, and printing functions (and some ingenuity).

Input Specifications

Read from the standard input. The input is supposed to be a C program. You need not check that it is, in fact, a legal C program; your program should simply assume it is.

If the first non-whitespace character on the line is a "#" and the line is not in a comment, the line is a *preprocessor directive*. The next maximal sequence of alphanumeric characters and underscores "_" is the command to the preprocessor. (For example, the line

```
#include <stdio.h>
```

is a preprocessor directive, and the preprocessor command is "include".)

Output Specifications

Write to the standard output. In the first 6 columns of each output line, print a line number followed by a period ".". Print the input line beginning in column 9 (one tab from the left margin)..

If the line is a preprocessor directive, the next output line should contain "*"s in the first 8 columns, followed by the phrase "Preprocessor directive: " and the directive *only* on the same line.

Example Input

```
/* next line is a preprocessor directive */
#include <stdio.h>
/* next line is not:
#include <something.h>
because the comment is 3 lines long */

int main(void)
{
#ifdef mips
      printf("2/3 = %d, 2.0/3.0 = %18.14f, 2.0F/3.0F = %18.14f\n",
            2/3, 2.0/3.0, 2.0F/3.0F);
#endif
}
```

Example Output

```
    1.  /* next line is a preprocessor directive */
    2.  #include <stdio.h>
********Preprocessor directive: include
    3.  /* next line is not:
    4.  #include <something.h>
    5.  because the comment is 3 lines long */
    6.
    7.  int main(void)
    8.  {
    9.  #ifdef mips
********Preprocessor directive: ifdef
   10.      printf("2/3 = %d, 2.0/3.0 = %18.14f, 2.0F/3.0F = %18.14f\n",
   11.              2/3, 2.0/3.0, 2.0F/3.0F);
   12.  #endif
********Preprocessor directive: endif
   13.  }
```

Extra Credit

Submitting: Put your answer at the end of your *Answers* file for the first part.

Compile and execute the program in the example input. Explain why the results of the divisions are all different, especially the last two. It is ***not*** sufficient to note the middle one is double precision, and the last one floating point; you need to explain how that causes a difference.