Homework 1

Due Date: Friday, April 12, 2002 **Points**: 100

UNIX System

1. (*10 points*) Give a one-line UNIX command that capitalizes the vowels (the letters "a", "e", "i", "o", and "u") in a file.

2. (*10 points*) Use a UNIX program to compute the remainder of 7358^349 when divided by 1345.

C Programming

3. (*25 points*) Write a program that asks the user to enter two integers, then calculates and displays their greatest common divisor (GCD). The input and output must look like this:

```
Enter two integers:  14 49
The GCD of 14 and 49 is 7.
```

   where "14" and "49" are numbers the user types and "7" is the corresponding GCD. Be sure to check the input for validity, and terminate the program when the user types an end of file character! (***Hint***: The classic algorithm for computing the GCD, known as Euclid's Algorithm, goes as follows. Let *m* and *n* be variables containing the two numbers. Divide *m* by *n*. Save the divisor in *m*, and save the remainder in *n*. If *n* is 0, stop; *m* contains the GCD. Otherwise, repeat the process, starting with the division of *m* by *n*.)

4. (*30 points*) Every computer is limited in the amount of precision it can represent for floating-point numbers. At some point, where *epsilon* is very small, the following expression will be true:

```
1.0 == 1.0 + epsilon
```

   Write a program to find the largest value of *epsilon* on your computer. Note that the value of *epsilon* may be different for floats and doubles. Find both values (and the value for long doubles if your compiler supports them). Also, use 1.0 rather than 0.0 to test *epsilon* because most computers have special hardware instructions for handling zero arithmetic. (***Hint***: for *gcc*(1), a long double is the same as a double.)

Debugging

5. (*25 points*) On the class web page is the source for a C program named *vis.c* that reads characters from the standard input and prints them on the standard output after expanding any non-printing characters to their C character escape sequence.

   The relevant characters, and the C escape sequences to be printed when those characters are encountered, are:

character	print as	character	print as
newline	\n	backslash	\\
horizontal tab	\t	vertical tab	\v
backspace	\b	carriage return	\r
form feed	\f	bell	\a
NUL	\0	anything else	\ooo

   The "anything else" entry means that any non-printing character other than the ones named in the table is to be printed as a sequence of three octal digits preceded by a backslash. When the escape sequence for a newline is printed, the program is to skip to the next line.

Unfortunately, the program as saved in *vis.c* will not even compile, let alone run. And the programmer thoughtlessly left off all the comments. Hence, your mission: comment the program, and fix it so it works as described above! You are to turn in a corrected source program, with comments describing the changes you made to get it to work.

Extra Credit

6. (*10 points*) In question 4, the hint says that a long double and a double use the same number of bits. Write a program to demonstrate that the size of a long double is the same as that of a double.