CS 2750	**Advanced Programming in Unix**	Test 3
Name:	Winter 2005	Maximum Points: 44

Important: This is an open book test. You can use any books, notes, or paper but no electronic device. *Do not log into the computer during the test*. Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. You will lose five points for not writing your name.

1. [12 pt] Give a Unix command, and the equivalents in `sed` and `awk` to achieve the following:

   (a) Print the file named `foobar` to `stdout` but only lines numbered 4 through 20 (inclusive).

   (b) Print all the lines in file `foobar` but only columns numbered 10 through 25 (inclusive).

2. [6 pt] In English language, every occurrence of the letter `q` is followed by the letter `u`. There are few exceptions to this rule, such as the words `Iraq` and `Iraqis`. Can you give me a command to search for such words in a document? Will your command work if the word `Iraq` occurs as the last word on a line? Give me a command that will recognize the letter `q` at the end of a word. The commands can be taken to be `sed` or `vi` commands.

3. [4 pt] Explain the use of flags `-I` and `-L` when building executables from source code. Indicate whether they are needed at compile time, link time, or both.

4. [4 pt] How does the use of an asterisk (`*`) differ in shell compared to regular expressions for search in `sed`?

5. [8 pt] Give me an `awk` command so that any line with more than 72 characters is split into two lines: first line with 72 characters and the next line with the remaining characters. If a line is less than 72 characters, display it with no change.

6. [10 pt] Examine the following code fragment and point out any errors:

```
int instance;
for ( instance = 0; instance < 10; )
{
    pid_t pid, cpid;
    if ( ( pid = fork() ) < 0 )
        perror ( "Could not fork a child" );

    if ( pid )
        cpid = wait ( &status );

    execl ( ".", "foobar", ( char * )( instance++ ), ( char * )( NULL ) );
}
```

Assume that there is an executable in the current directory named `foobar` that requires a number as command line argument.