```
/**
 *  Author: Evan Krimpenfort
 *
 *  Project: Assignment_3.c (will be turned in as a pdf)
 */

/*
 * 1. Assuming a correct program show_args which prints its command-line
arguments to standard output,
 *    one per line, give the output generated by the shell command line:
 *
 *    $ ./show_args one two three four
 *    one
 *    two
 *    three
 *    four
 *
 * 2. Start with the following environment:
 *
 *    $ stty kill '@'
 *    $ stty erase '#'
 *    $ stty lnext '\'
 *    $ sh
 *
 *    Explain the results of each of the commands in the following
transcript:
 *
 *    $ date\@              // after typing '\', the '@' character is treated
like normal so its date@
 *    date@: not found
 *    $ date                // just a normal date
 *    Fri Sep 2 09:10:45 EDT 2005
 *    $ #date               // '#' deletes a character before it so date can
still go on as date
 *    Fri Sep 2 09:10:45 EDT 2005
 *    $ \#date              // again, the '\' character treats the next like
normal so its #date
 *
 * 3. Assume the file junk.
 *    In one sentence each, explain the output of each of the following
command lines (there are 10):
 *
 *    $ ls junk     // listing anything that is junk
 *      junk
```

```
 *    $ echo junk   // is echoing the word junk
 *      junk
 *    $ ls /        // listing anything in the base directory
 *      bin  boot  dev  etc  home  lib  lib64  lost+found  mnt  opt //
 *      proc  root  run  sbin  selinux  srv  sys  tmp  tmphome  usr  var
 *    $ echo /      // is echoing the character '\'
 *      /
 *    $ ls          // listing the current directory
 *      a1  a2  a.out  assignment_1.c  assignment_2.c  assignment_3.c
junk.txt
 *    $ echo        // echoing nothing
 *      <nothing>
 *    $ ls *        // listing the current directory
 *      a1  a2  a.out  assignment_1.c  assignment_2.c  assignment_3.c
junk.txt
 *    $ echo *      // is echoing the character '*'
 *      *
 *    $ ls ' * '    // trying to list a string of characters but that doesn't
work
 *      ls: cannot access ' * ': No such file or directory
 *    $ echo ' * '  // echoing the string of characters ' * '
 *      ' * '
 *
 * 4. Show and explain the output of the following Korn shell commands:
 *
 *    echo 'Go $HOME'
 *    Go $HOME      // only prints that and not the $HOME variable because
its within single quotes
 *
 *    echo "$5.00 is too much!"
 *    .00 is too much!  // prints everything but the $5 because double quotes
still allow variables to pass
 *
 *    echo $(who | wc -l) users is not very many
 *    1 users is not very many  // prints the amount of users on the machine
piped into the wordcount -line command
 *
 * 5. Give the output of the following command lines (assume there are 9
files in the current working
 *    directory, /home/linda, and x=10):
 *
 *    a) $ echo 'Send output of "command" to file descriptor 2'
 *         'Send output of command to file descriptor 2'
 *    b) $ echo "Well, isn't that \"special\"?"
```

```
 *         Well, isn't that "special"?
 *    c) $ echo "You have $(ls | wc -l) files in $(pwd)"
 *     You have 9 files in /home/linda
 *    d) $ print "You have \$(ls | wc -l) files in \$(pwd)"
 *         You have $(ls | wc -l) files in $(pwd)
 *    e) $ echo 'You have $(ls | wc -l) files in $(pwd)'
 *         'You have $(ls | wc -l) files in $(pwd)'
 *    f) $ echo "The value of \$x is $x"
 *         The value of $x is 10
 *    g) $ print "The value of $x is \$x"
 *         The value of 10 is $x
 *    h) $ echo 'Go $HOME'
 *         'Go $HOME'
 *    i) $ echo "$5.00 is too much!"
 *         .00 is too much!
 *    j) $ echo $(who | wc -l) users is not very many
 *         1 users is not very many
 *
 * 6. Suppose a command mystery writes its output to stderr. Give a single
command line which would
 *    pipe this output to wc -l.
 *    $ (mystery | wc -l) >&2
 */
```