CSC209H Worksheet: Shell Programming

1. What are the values of the variables `foo` and `bar` after executing the following commands? Suppose that the user types `bip` on the standard input.

   - ```
     foo=bar
     bar=baz
     read foo
     ```

     `foo:` `bar:`

   - ```
     foo=bar
     bar=baz
     read $foo
     ```

     `foo:` `bar:`

   Figure this out in your head and then print your answers in the boxes above (without actually trying the commands.) Then type the commands on the computer to verify your answer.

2. Write a shell program in a file called `whatday.sh` that simply asks the user what day it is, reads what the user types from `stdin` and prints back a message saying `I love <day>.` where `<day>` is what the user entered. When it is run, your program should look something like this.

   ```
   $ sh whatday.sh
   What day is it?
   Wednesday        <-- user types this
   I love Wednesday.
   ```

3. What is the problem if you try to change your program so that the output looks like this? Notice the `s` after the day name this time. The videos didn't cover the easiest solution to this problem. If you already know how to solve it, go ahead and fix your program, if not move on to the next problem on this worksheet for now. Come back to work on this (with Google's help) after you have finished everything else.

   ```
   $ sh whatday.sh
   What day is it?
   Wednesday        <-- user types this
   I love Wednesdays.
   ```

4. Complete the table on the back of this handout by typing each expression into the bash shell. If the command produces an error, give the error message. Otherwise, show the result printed to standard output. In both cases, provide the return value. Remember that the variable `$?` will hold the return value of the last command that was executed. You can type `echo $?` to see this value.

CSC209H Worksheet: Shell Programming

Expression	Error? (Y/N)	`stdout` or Error message	Return value
`test 3 = 4`			
`tree=maple` `test $tree`			
`tree=maple` `test ($tree = oak)`			
`test $undefined = something`			

`echo yes > file1`
`echo no > file2`
`diff file1 file2`			

`cp file1 file3` `diff file1 file3`			
`grep es file3`			
`value=3` `expr value + 4`			
`value=3` `expr $value * 4`			

5. Copy your `whatday.sh` program to the file `weekday.sh` and change it to respond with either the message `"The weekend! Yipee!"` or `"Back to work :("` depending on the input from the user. Consider the input `Sunday` and `Saturday` as the only valid weekend selections.

6. See what happens when you run your `weekday.sh` program and press enter without entering a day at all. Fix your program so that when this happens, the program prints `I don't know what day it is.`

7. The same directory also has files named `chapter1` and `chapter-one` which both contain the first chapter of Sherlock Holmes. One of them has been corrected and the time-stamps have been corrupted and so don't tell you which one was the original. Which one has the mistake and which has the correction?

8. In `/u/mcraig/209/shell-programming/` there are files named `0` through `9`, that are the same. Except that one of them is not the same. Which one? Feel free to simply solve this from the shell prompt with a solution that only works because the number of files is small. What would you need to be able to find the mismatched files from a set of 100?