PPA –WEEK 3

List(in long format, use `ls -l`) all the `.txt` files in the current working directory and redirect the output to a file named `textFiles.txt` and also print 'found' to the terminal(without quotes, do not print anything else).

If no `.txt` file exists redirect the error of your command to the file `noFiles.txt` and do not print anything.

Hint: Make use of redirection to file and operators to write solution in one line.

Ans.

```
script()
ls -l *.txt > textFiles.txt 2> noFiles.txt && echo found
}
```

Given a shell variable `month` supposed to contain a string value corresponding to some calendar month. Use the `cal` command to create a file named as `X.txt` where `X` is the string value in the variable `month`. Your command should also create a file named `error.txt` that should contain the error message if the string `month` does not correspond to any calendar month. Create all the files in the current working directory.

For example:

If the variable `month` contains the string "nov", your solution should create a file named `nov.txt` containing the calendar of november month and `error.txt` should be empty.

And if the variable `month` contains the string "garbage", your solution should create a file named `error.txt` containing the error from cal command and `garbage.txt` should be empty.

Ans.

```
script() {
read month
cal -m $month > $month.txt 2> error.txt
echo --$month.txt–
cat $month.txt
echo
echo --error.txt–
cat error.txt
}
```

Execute the commands given below in the sequence and collect the output/error into a file `errorlog` as described below.

Execute the command `test` and redirect the standard error to the file `errorlog`.

Execute the command `test -e` and append the standard error output to the file `errorlog`.

Execute the command `test -n`. and append the standard error to the file `errorlog`.

Ans.

```
script() {
test 2> errorlog
test -e 2>> errorlog
test -n 2>> errorlog
}
```

GRPA –WEEK 3

Add the string "EOF alpha" at the end of the file(starting at a new line) `alpha.txt` then append the contents of the file `numbers.txt` at the end of the file(starting at a new line) `alpha.txt`. `alpha.txt` and `numbers.txt` are located in the current working directory.

Ans.

```
script() {
echo "EOF alpha" >> alpha.txt; cat numbers.txt >> alpha.txt
}
```

Print the number of lines present in 'file1' and 'file2' combined, your solution should not print anything else. 'file1' and 'file2' are located in the current working directory.

Hint: Multiple files can be given as argument to 'cat' command.

Ans.

```
script() {
cat file1 file2 | wc –l
}
```

There are three files `master.txt`, `half1.txt` and `half2.txt` in the current working directory. Add first 2 lines of `half1.txt` to the file `master.txt` at the end(starting at a new line) then append the last 3 lines of the file `half2.txt` to the file `master.txt` at the end(starting at a new line). Append the lines in the sequence mentioned.

Ans.

```
script() {
head half1.txt -n2 >> master.txt
tail half2.txt -n3 >> master.txt
}
```

An observer wrote a script named `createTwingle` that produces a file `twingle` containing names of all the visible stars present in the sky at that instant. Every line in the file `twingle` is the name of a star. In your current directory the file `twingle` may or may not be present.

If the file `twingle` is present in the directory then print the number of lines in the file, else execute the command `createTwingle` it will create the file `twingle` in the current working directory then print the number of lines in the file `twingle`.

Hint: Try to use operators discussed in the lectures to give a single line solution for the task.

Ans.

```
script() {
wc -l twingle || (createTwingle && wc -l twingle)
}
```

Print the number of directories in the current working directory. Do not print anything else.

Hint: One solution is to make use of 'ls', 'wc' and pipes('|').

Ans.

```
script() {
ls -d */ | wc –l
}
```

The script `test` will print some text to the standard output, it can be run similar to any other command and does not accept any arguments.

Your task is to print the output after running `test` on the screen and also append the output at the end(starting at new line) of the file `log`. File `log` is located in the current working directory.

Hint: To solve it in one line check the man page of `tee` command for appending to the file.

Ans.

```
script() {
test | tee templog
cat templog >> log
}
```