

## Shell Scripting Assignment (Probably the 2nd or 3rd one)

### Q1. The Package.

You are the label head for Rocafella Records. Shawn Corey Carter has sent you a package containing the files with the lyrics for his latest yet-to-be-released album, code name 'Blueprint IV'.

In typical celebrity fashion he simply sent 20 files titled, song1.lyric, song2.lyric, ... song20.lyric and did not specify the titles for the songs.

Being the label head you would like to have the files correctly labeled according to their song name.

Fortunately, you use Linux and are a wizard with bash. So you whip up a script to do this for you. You will have to extract the file name from the .lyric file itself and then rename the file accordingly.

Example:

song1.lyric becomes Renegade.lyric, where Renegade is the song title/name.

The format specification for the lyric file is as follows:

Title: Renegade (Song name may contain spaces)

Artist: Shawn Corey Carter, Marshal Mathers (Comma separated list of artists)

Producer: Marshal Mathers, DJ Head (Comma separated list of tentative producers)

(Blank line here)

Lyrics:

[verse 1]

Lorem ipsum dolor sit amet, consectetur adipisicing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco  
laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum  
dolore eu fugiat nulla pariatur.

<Blank line after verse ends>

[verse 2]

(Text for verse two here)

...

...

... so on

[verse n] (n is an arbitrary positive integer)

(Text for verse n here)

## Q2. The Hype.

As part of the promotions for “Blueprint IV” you wish to secretly release the Tracklist of the album as teaser and to build up hype for the upcoming album. Again being the masterful shell artist you are, you quickly write down a script to generate a single file called: Tracklist.list which contains all the Track titles in alphabetical order preceded by the track number and a period and a space. For example the tracklist might look like:

```
01. 99 Problems
02. Izzo
03. Renegade
...
...
...
20. The Takeover
```

## Q3. Censorship

Artists being artists tend to use lots of expletives in their quest to convey emotions, sometimes though censorship is necessary. Partly to make a song more Radio friendly and partly to reach a larger audience, which might not take kindly to such “frank” language.

Your job as the label head includes censorship.

Being rich, famous and busy you have neither the time nor the wish to manually censor the lyrics, again, you turn to your trusty terminal and write a script that will censor the lyrics for you.

You have a file called “expletives.list” which contains a list of words you wish to censor.

A word is considered censored if it is replaced by \*\*\*\*\* (5 asterisks).

For example assume one of the files is called Renegade.lyric, you need to look for words listed by expletives.list in Renegade.lyric and censors them.

Assume expletives.list looks like

```
....
Lorem
tempor
amet
tincidunt
....
```

And Renegade.lyric contains:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

This line in Renegade.lyric should now read:

\*\*\*\*\* ipsum dolor sit \*\*\*\*\* , consectetur adipisicing elit, sed do eiusmod \*\*\*\*\*

Assuming you are given all 20 lyrics files named with the song title, You need to write a script do this for all the .lyric files.

#### **Q4. Spy Game.**

Sources inform you that Mr. Carter's arch-rival and nemesis Nasir Jones plans to steal the lyrics of "Blueprint IV" and release them to the public before release of the album.

If such a scenario were to come to fruition, it would have catastrophic consequences.

But you being the head honcho at Rocafella records didn't get to that position without effort, you know how to deal with theft.

Again, you crack your knuckles, chuckle and type away at the keyboard, this time to write a script that encrypts the lyrics and saves them in new files called <Song Title>.encrypted

You use the "md5sum" command to compute the encryption for your purposes. (ignore the fact that it computes an md5 checksum for the time being, you do not need to know what it is.)

You need to encrypt all the lyrics of a file. That is, the entire contents of the .lyric file the 5th line (after the blank line after producer) onwards.

The new .encrypted files are of the form:  
(Example Renegade.lyric)

Title: Renegade (Song name may contain spaces)

Artist: Shawn Corey Carter, Marshal Mathers (Comma separated list of artists)

Producer: Marshal Mathers, DJ Head (Comma separated list of tentative producers)

(Blank line here)

892a802bdfb33619c461130e9f9714e4 (This is the encrypted string returned by md5sum)

#### **Q5. Sales and Accounting**

So "Blueprint IV" was released without major mishap and the sales figures are turning in.

You need to write a memo to your staff giving them brief statistics.

You have a file with the number of sales per song. Assume all songs were sold through iTunes at the standard rate of \$0.99.

You need to report the following statistics in the same order. (Give only the actual figures or names, no unnecessary text)

1. Total number of songs sold.
2. Total money made.
3. Best selling song.
4. Worst selling song.

You have a file called stats.csv, which contains the data in the following general format:

<Song Title>, Numbers sold.

For example a sample stats.txt would be:

Renegade, 1920001

The Takeover, 500782

...

...

...

You need to write the statistics to a file called stats.memo

A sample stats.memo would look like:

3876521

\$3837755.79

Renegade

The Takeover

As always, you automate the procedure by writing a script.

## **Q6. Payday**

Now that records have been sold and money's been made, it's time to pay the Artists and producers.

Assume that 50% of the total earnings are to be distributed among the Artists, producers and ofcourse ... you. Assume that this 50% is distributed equally.

Assume you have the stats.memo file and the <Song Title>.encrypted files with you.

You simply need to compute the the amount each person should receive. Output this into a file called payday.dreday

Example payday.dreday file:

\$132100