# Readme

1. Complete the printout it without looking stuff up. The tasks are not connected so if you don't know one answer, simply move on.
2. Test your answers (10 points for each correct answer) using the Org-mode version of the quiz (at tinyurl.com): redirection-pop-org. Solution: redirection-pop-solution.
3. An answer is correct (10 points) if it runs and returns the correct result. You can give yourself extra points if you were close!
4. Grade yourself and submit your original paper copy to me no later than Tuesday, April 2, 2.30 pm, with your name and pledge.

# Task 1: Locate Standard Device Files

Use the ls command to list all files in the `/dev` directory that include `"std"` in their names.

```bash
ls -l /dev/* | grep std
```

```
lrwxrwxrwx   1 root root            15 Mar 24 18:06 /dev/stderr -> /proc/self/fd/2
lrwxrwxrwx   1 root root            15 Mar 24 18:06 /dev/stdin -> /proc/self/fd/0
lrwxrwxrwx   1 root root            15 Mar 24 18:06 /dev/stdout -> /proc/self/fd/1
```

# Task 2: Understanding `rev` Command

Demonstrate how the `rev` command can reverse a string. Use the string "OpenAI" as an example.

Tip: remember that `rev` operates on `stdin` - so you need to pipe the input into `rev` for this to work in the code block.

```
echo "OpenAI" | rev
```

```
IAnepO
```

# Task 3: Redirect Output to a File

Redirect the listing of all files in the `/usr/bin` directory to a file named `list.txt`, then display only the bottom five lines of the file.

```
ls -l /usr/bin > list.txt
tail -n 5 list.txt
```

```
-rwxr-xr-x 1 root root      875096 Mar 24  2022 zstd
lrwxrwxrwx 1 root root           4 Mar 16 11:44 zstdcat -> zstd
-rwxr-xr-x 1 root root        3869 Mar 24  2022 zstdgrep
-rwxr-xr-x 1 root root          30 Mar 24  2022 zstdless
lrwxrwxrwx 1 root root           4 Mar 16 11:44 zstdmt -> zstd
```

# Task 4: Check File Type

Determine the type of the previously created `list.txt` file.

```
file list.txt
```

```
list.txt: ASCII text
```

# Task 5: View Top Lines of a File

Display the first 5 lines of the `list.txt` file, `tee` the output off to a file `top.txt`, and count the number of words.

```
head -n 5 list.txt | tee top.txt | wc -w
```

```
38
```

# Task 6: Redirect Non-Existing Directory Listing to a File

Attempt to list a non-existing directory (`/non_existing_dir`) and redirect the error message to a file named `error.txt`, then display the file content.

```
ls -l /non_existing_dir 2> error.txt
cat error.txt
```

```
ls: cannot access '/non_existing_dir': No such file or directory
```

# Task 7: Create an Empty File

Create an empty file named `empty.txt` without using `touch`, just redirection. List the file to make sure it's empty.

```
> empty.txt
ls -l empty.txt
```

```
-rw-rw-r-- 1 marcus marcus 0 Mar 25 23:42 empty.txt
```

# Task 8: Append Text to a File

Put the text "Nothing to see here" in a file `append.txt`. Now append the text "Adding more content" to the end of the file and display the file.

```
echo "Nothing to see here" > append.txt
echo "Adding more content" >> append.txt
cat append.txt
```

```
Nothing to see here
Adding more content
```

## Task 9: Display Environment Variables

On one line, write a list of all environment variables to a file `env.txt` and count the number of variables (the displayed output).

Tip: the command to print all environment variables is `printenv`.

On the next line count the lines of the file `env.txt` you created.

On the last line, display only the first line of `env.txt` using the `--lines` flag.

```
printenv | tee env.txt | wc -l
cat env.txt | wc -l
head --lines 1 env.txt
```

```
43
43
SHELL=/bin/bash
```

## Task 10: Use a Loop to Create Files

Put the line "Error message:" in a file `error2.txt`. Now list a non-existing directory (`/non_existing_dir`) and append both `stdout` and `stderr` to `error2.txt`. Lastly, count the number of lines of `error2.txt`.

```
echo "Error message" > error2.txt
ls -l /non_existing_dir &>> error2.txt
cat error2.txt | wc -l
```

```
2
```

Author: marcus

Created: 2024-03-25 Mon 23:42

Validate