

Interlude: PA1 - Solutions

These Practice Exercises are meant to help you review for our first Intermediary Exam: IE1.

$+(\dots \mid \dots)$

What it actually means:

- 1 or more occurrences of EITHER pattern
- $+(aa|bb)$ would expand into aaaaaa bbbbbbb but also **aabbbbbaa**

What would we use instead in order to list only aaaaaa or bbbbbbb but not aabbbbbaa?

```
ls {+(aa),+(bb)}  
ls @(+(aa)|+(bb))
```



Replacement Commands

You receive information that one of your servers was cracked, the cracker probably replaced the **ls** command. You know that the **echo** command is safe to use. Can **echo** replace **ls**? How can you list the files in the current directory with **echo**?

```
Echo *
```

Is there another command, besides **cd**, to change directories?

```
pushd or popd
```

Hacker Wannabe

It is a good day for a prank.

A friend left their workstation unlocked with a terminal running bash opened. We want to make it so that, when they use `ls`, the following message is displayed instead of running the command;

```
tux@tuxbox$ ls
```

```
ERROR 0xFF42 (bad sector \\ data corrupted).  
Contact NSA for a "backup" copy of your data.
```

```
tux@tuxbox$ alias ls="echo ERROR 0xFF42 \ (bad  
sector \\\ data corrupted\). Contact NSA for  
a \"backup\" copy of your data."
```

Aliases of aliases of....

- Define an alias `justdoit` that echo a message on the screen
- Define an alias `dontdoit` that calls the previous alias. What happens?
- Redefine the alias `justdoit` so that it calls the alias `dontdoit`. What happens?

- The 1st two items work as expected; the 2nd alias calls the 1st that is replaced by its command
- The last item does not cause the infinite recursion that we might have expected

Brace Yourself: Defining multiple aliases to the same command

- We want to create aliases `one`, `two` and `three`, that all correspond to `echo this is pretty cool`
- However, we want to issue only a single `alias` command

```
tux@tuxbox$ alias {one,two,three}='ls -l '
```

The Folders Factory

Create the following folders in one command each:

- Weird"folder
- Weird folder → yes, there is a space in the name of that one folder
- Weird\folder
- Weird/folder → yes, that's the actual name of the one folder
- Weird/folder → this time folder is a subfolder of Weird, which does not exist yet

If you bump into something impossible, explain why it is so.

```
tux@tuxbox$ mkdir Weird\"folder Weird\ folder
Weird\\folder
tux@tuxbox$ mkdir -p Weird/folder
```

All in one, and one in all!

We have a folder containing the following subfolders

- COP4610
- COP3353
- COP2512
- COP2513
- CIS4930

We want to move, in a single command, all folders except COP3353 into CIS4930. **Do so without having to list explicitly all folders that you want to move.**

```
tux@tuxbox$ mv !(CIS4930) CIS4930/
```


Pause... Resume...

1. Execute a sleep 600 process in the background
2. Make sure it is running in the background
3. Find out its PID
4. Pause its execution by sending an appropriate signal to its **PID**
5. Display information confirming that it is now frozen
6. Resume its execution by sending an appropriate signal to its **Job ID**
7. Display information confirming that it is resumed
8. Ask it politely to terminate by sending it an appropriate signal

```
sleep 600 &  
jobs  
ps or jobs -l  
kill -s STOP 20599  
jobs  
kill -s CONT %1  
jobs  
kill -s TERM 20599
```

You may “quote me”

Provide the `echo` command you would type in your shell to display each of the following outputs;

- The dog & the cat; a tale of getting along just fine I remember now...
- "Bash is fun", they said. Liars!
- It's rather annoying (and even at times infuriating) to see quoting fail
- Why are we using `\\` when we want just a `\` to be displayed

```
echo "& ; \" ! ' ( ) \\\\ \\"
```

Hidden treasures

- Let us list all files that end with the suffix .exe
- How do we make it so that the hidden files are **also** displayed?
- How do we make it so that **only** the hidden files matching the above pattern are displayed?

```
ls *.exe  
shopt -s dotglob  
ls *.exe  
ls .*.exe
```

Ending with single digit

Let us list all files that start with a lowercase letter and end with a **single** digit:

Possible matches:

 afilelikethis9

 a9

Not matching:

 Afilelikethis9

 afilelikethis42

```
ls [a-z]*[^0-9][0-9]
```

The case of the secret filename

How would you use globbing patterns to match filenames containing the word secret or SECRET?

Same question if the words can also be sEcReT and SeCrEt?

What if the spelling include all possible combinations of cases?

```
shopt -s nocaseglob  
ls -l *secret*
```

Extglob FTW!

Use extended globbing patterns to list the files that

- Start with the name of a color (red, blue, or green), followed by the name of an animal (cat, dog, wolf, panther)
- Contain 1 or more consecutive occurrences of a pattern consisting of the word project, followed by an underscore, followed by a 4 digit number.
- Are spelled using between 3 and 5 characters
- Start with the word number followed by a series of digits that are in ascending order. E.g., number123 would work, so would number233455. However, number231 would not be matched.

```
@(red|blue|green)@(cat,dog,wolf,panther)
```

```
+(project_[0-9][0-9][0-9][0-9])
```

```
@(???|????|?????)
```

```
Number*(1)*(2)*(3)*(4)  
) and so on so forth
```