Link to the challenge:

https://cmdchallenge.com/

Snail:

Challenge:

Your first challenge is to print "hello world" on the terminal in a single command.

Answer:

```
echo "hello world"
```

Butterfly:

Challenge:

Print the current working directory.

Answer:

```
pwd
```

Larva:

Challenge:

List names of all the files in the current directory, one file per line.

Answer:

```
ls
```

Ant:

Challenge:

There is a file named access.log in the current directory. Print the contents.

Answer:

```
cat access.log
```

Spider-Web:

Challenge:

Print the last 5 lines of "access.log".

Answer:

```
tail -5 access.log
```

Whale:

Challenge:

Create an empty file named take-the-command-challenge in the current working directory.

Answer:

```
touch take-the-command-challenge
```

Dolphin:

Challenge:

Create a directory named tmp/files in the current working directory

Answer:

```
mkdir tmp && mkdir tmp/files
```

Shark:

Challenge:

Copy the file named take-the-command-challenge to the directory tmp/files

Answer:

```
cp take-the-command-challenge tmp/files/take-the-command-challenge
```

Fish(1):

Challenge:

Move the file named take-the-command-challenge to the directory tmp/files

Answer:

```
mv take-the-command-challenge tmp/files
```

Fish(2):

Challenge:

A symbolic link is a type of file that is a reference to another file. Create a symbolic link named take-the-command-challenge that points to the file tmp/files/take-the-command-challenge.

Answer:

```
ln -s tmp/files/take-the-command-challenge take-the-command-challenge
```

Puffer Fish:

Challenge:

Delete all of the files in this challenge directory including all subdirectories and their contents.

Answer:

‘rm -rf * .*’ or ‘find -delete’

Wolf:

Challenge:

There are files in this challenge with different file extensions. Remove all files with the .doc extension recursively in the current working directory.

Answer:

```
find -type f -name "*.doc" -delete
```

Bee:

Challenge:

There is a file named access.log in the current working directory. Print all lines in this file that contains the string "GET".

Answer:

```
cat access.log | grep GET
```

Beetle:

Challenge:

Print all files in the current directory, one per line (not the path, just the filename) that contain the string "500".

Answer:

```
grep -l 500 *
```

(-l: print the filename that contains the specified string,

*: search in all files)

Grasshopper:

Challenge:

Print the relative file paths, one path per line for all filenames that start with "access.log" in the current directory.

Answer:

```
find -type f -name "access.log*" -exec echo {} \;
```

Spider:

Challenge:

Print all matching lines (without the filename or the file path) in all files under the current directory that start with "access.log" that contain the string "500"

Answer:

```
find -type f -name "access.log*" -exec cat {} + | grep 500
```

Scorpion:

Challenge:

Extract all IP addresses from files that start with "access.log" printing one IP address per line.

Answer:

```
find . -type f -name "access.log*" -exec cat {} + | grep -E -o '[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+'
```

(-E: extended regex, -o: print only matching part, and not the whole line)

Fly:

Challenge:

Count the number of files in the current working directory. Print the number of files as a single integer.

Answer:

```
ls -R | wc -l
```

Worm:

Challenge:

Print the contents of access.log sorted.

Answer:

```
cat access.log | sort
```

Amba:

Challenge:

Print the number of lines in access.log that contain the string "GET".

Answer:

```
cat access.log | grep GET | wc -l
```

Gorilla:

Challenge:

The file split-me.txt contains a list of numbers separated by a ; character.

Split the numbers on the ; character, one number per line.

Answer:

```
cat split-me.txt | sed "s/;/\n/g"
```

Dog:

Question:

Print the numbers 1 to 100 separated by spaces.

Answer:

```
awk 'BEGIN { for (i = 1; i <= 100; i++) printf "%d ", i }'
```

Fox:

Challenge:

This challenge has text files (with a .txt extension) that contain the phrase "challenges are difficult". Delete this phrase from all text files recursively.

Answer:

```
find -type f -name "*.txt" -exec sed "s/challenges are difficult//g" -i {} \;
```

or

```
find -type f -name "*.txt" -exec sed -i "/challenges are difficult/d" {} \;
```

Cheetah:

Challenge:

The file sum-me.txt has a list of numbers, one per line. Print the sum of these numbers.

Answer:

```
awk '{ sum += $1 } END { print sum }' sum-me.txt
```

Lion:

Challenge:

Print all files in the current directory recursively without the leading directory path.

Answer:

```
find -type f -exec basename {} \;
```

Tiger:

Challenge:

Rename all files removing the extension from them in the current directory recursively.

Answer:

```
mv * .*
```

Horse:

Challenge:

The files in this challenge contain spaces. List all of the files (filenames only) in the current directory but replace all spaces with a '.' character.

Answer:

```
“ls | sed ‘s/ /./g’ “
```

Unicorn:

Challenge:

In this challenge there are some directories containing files with different extensions. Print all directories, one per line without duplicates that contain one or more files with a ".tf" extension.

Answer:

```
find . -type f -name "*.tf" -exec dirname {} \; | sort -u
```

(-u flag = unique)

Cow:

Challenge:

There are a mix of files in this directory that start with letters and numbers. Print the filenames (just the filenames) of all files that start with a number recursively in the current directory.

Answer:

```
ls -R -p | grep "^[0-9]" | grep -v /
```

Pig:

Challenge:

Print the 25th line of the file faces.txt

Answer:

```
sed '25q;d' faces.txt
```

Mouse:

Challenge:

Print the lines of the file reverse-me.txt in this directory in reverse line order so that the last line is printed first and the first line is printed last.

Answer:

```
tac reverse-me.txt
```

Hamster:

Challenge:

Print the file faces.txt, but only print the first instance of each duplicate line, even if the duplicates don't appear next to each other.

Answer:

```
awk '!seen[$0]++'
```

Bear:

Challenge:

The file random-numbers.txt contains a list of 100 random integers. Print the number of unique prime numbers contained in the file.

Answer:

```
cat random-numbers.txt | sort | uniq | factor | awk 'NF==2' | wc -l
```

Panda:

Challenge:

access.log.1 and access.log.2 are http server logs.

Print the IP addresses common to both files, one per line.

Answer:

```
cat access.log.1 access.log.2 | awk '{print $1}' |  sort | uniq -d
```

Chick:

Challenge:

Print all matching lines (without the filename or the file path) in all files under the current directory that start with "access.log", where the next line contains the string "404".

Answer:

```
grep -h -B1 404 **/access.log*|grep -vE '404|--'
```

peacock:

Challenge:

Print all files with a .bin extension in the current directory that are different than the file named base.bin.

Answer:

```
diff *.bin --to-file base.bin | cut -d " " -f 3
```

Frog:

Challenge:

There is a file: ./.../ /. .the flag.txt

Show its contents on the screen.

Answer:

```
cat './.../  /. .the flag.txt'
```

Dragon:

Challenge:

How many lines contain tab characters in the file named file-with-tabs.txt in the current directory.

Answer:

```
cat file-with-tabs.txt | grep $'\t' | wc -l
```

(the $ and single quote are mandatory)

Cyclamen:

Challenge:

There are files in this challenge with different file extensions.

Remove all files without the .txt and .exe extensions recursively in the current working directory.

Answer:

```
find -type f ! -name "*.txt" ! -name "*.exe" -exec rm {} +
```

(\; also works instead of +)

(+ multiple arguments for single rm commands – such as rm arg1 arg2..

And \; is for single run of rm each time: rm arg1 rm arg2…)

Rose:

Challenge:

There are some files in this directory that start with a dash in the filename. Remove those files.

Answer:

```
find -type f -name "-*" -exec rm {} \;
```

Lily:

Challenge:

There are two files in this directory, ps-ef1 and ps-ef2. Print the contents of both files sorted by PID and delete repeated lines.

Answer:

```
cat ps-ef1 ps-ef2 | sort -k2 -n | uniq
```

(-k2 – sort by second field)

(-n sort numerical instead of alphabetically)

Groundsel:

Challenge:

In the current directory there is a file called netstat.out.

Print all the IPv4 listening ports sorted from the higher to lower.

Answer:

```
cat netstat.out | grep -w "LISTEN" | awk '{print $4}' | cut -d":" -f2 | sort -rn
```