**Bash One liners:**

From sandbox/folders/numbered\_things:

* Copy all folders or files that start with 7 into destination\_7
* Move all folders that contain a 0 in the name into destination\_0\_folders
* Move all files where the filename without extension ends in a 1 to destination\_1.files
* Append the contents of all files that start with 4 to contents\_4\_files.txt
* Output the contents of all files in folders that contain 8 to contents\_8\_folders.txt, such that it contains nothing else afterwards
* Move all files containing nines to the folder in the parent directory called all\_the\_nines

From sandbox/folders:

* Merge the contents of sub\_b into sub\_a

From sandbox/files

* Find the lines across all of the files which contain all of the words “one” “two” and “three”

**Python scripts:**

*Printing:*

1. Output the text “hello world!” to the screen
2. Assign “hello” to one variable, and “world” to another. Output the text “hello world” to the screen using these variables
3. Using a single print statement, output “hello” and “world” on separate lines

*Conditionals:*

1. Given some integer variable my\_variable, tell the user whether the number is odd or even
2. Given some integer variable my\_variable, output “foo” if it is divisible by 3, “baz” if it is divisible by 5, and “foo-baz” if it is divisible by both 3 and 5
3. Do **5** again, but only testing for equality twice. (hint: use Boolean variables!)

*While loop:*

1. Collatz conjecture on some variable initial\_value

*Lists / For loop:*

1. Project Euler problem 1
2. Given a list of prime numbers, output a list that contains all of the prime factors of a number “product”
3. Project Euler problem 2

**Python functions:**

*Output:*

1. Write a function leet(input) that prints input in1337speek. (just replace all L’s with 1s, E’s with 3s, T’s with 7s). Have it accept from raw\_input(). Use a dictionary to determine how to replace.
2. Given a word list, write a function that prints out all legal anagrams of an input. Again, use raw\_input() to get the input.

*Return:*

1. Write a function that determines whether an input is a prime number, and returns True or False accordingly
2. Using the function above, write a function that returns a list of the first 100 prime numbers.

*Passing functions:*

1. Write code to reverse, transpose, and diagonalize a string.

By transpose, I mean that:

‘abc’

becomes

‘a

b

c’

by diagonalize, I mean that:

‘abc’

becomes

‘a  
 b  
 c’

Write another function that can apply a list of functions to a string. Try applying all three functions in a row to get a diagonal that goes from the top right to bottom left.

*Recursion (bonus):*

1. Generate the fibonacci sequence. In the Fibonacci sequence,

f(n) = f(n-1) + f(n-2)

where f(1) and f(2) are both 1