Bash:

1. Find common integers

Write a bash script that takes two input files, file1 and file2, each containing a list of integers, and outputs the common integers in both files. (files are not necessarily sorted)

Usage: bash common-integers.sh file1 file2 > common.txt

2. Array operations

Write a bash script to take an positive integer array of any length as an argument. Arguments should be space separated as shown below. Your script 'array.sh' should be able to display the length of the array in the first, then maximum number from the input array, and sum of the squares of each number in the same line separated by space.

Sample input:

bash array.sh 1 2 3 4 5

Output:

```
5 5 55
```

Explanation:

1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55

3. Rename photos with date created

photo-rename.sh Write a bash script called photo-rename.sh which creates one directory called "output". And copy all the files of the form ddmmyyyyANY.jpg present in the given directory. Also, rename that file name format to yyyy-mm-dd-ANY.jpg Here, 'dd', 'mm', 'yyyy' correspond to the two-digit date, two-digit month, and four-digit year when the photo was taken, and 'ANY' corresponds to any string (including the empty string). The first three fields consist only of digits 0 to 9.

- Your script should ignore jpg files which are not of the form ddmmyyyyANY.jpg.
- Your script should take the directory name as an argument
- Remember the extension has to be .jpg only.

Assumption:- Assume every file starting with 8 digits to be a valid date. No need for checks if the month/day is valid or not.

bash photo-rename.sh folder-name

AWK:

1. Write an awk script for the McDonald rhyme, which reads an input file which includes different types of ANIMAL and SOUND (see below example). You can use the below template as part of your “begin” in the awk script and then replace animal and sound with the input parameters. Your output should follow the template fully, including spacings, commas etc

Template:

Old McDonald had a farm ei-ei-o\n On that farm he had a ANIMAL ei-ei-o\n With a SOUND-SOUND here, SOUND-SOUND there\n Here a SOUND there a SOUND, everywhere SOUND-SOUND\n\n

For Example:

Input is:

```
pig:oink
cow:moo
dog:bow
```

Output is:

```
Old McDonald had a farm ei-ei-o
On that farm he had a pig ei-ei-o
With a oink-oink here, oink-oink there
Here a oink there a oink, everywhere oink-oink

Old McDonald had a farm ei-ei-o
On that farm he had a cow ei-ei-o
With a moo-moo here, moo-moo there
Here a moo there a moo, everywhere moo-moo

Old McDonald had a farm ei-ei-o
On that farm he had a dog ei-ei-o
With a bow-bow here, bow-bow there
Here a bow there a bow, everywhere bow-bow
```

To run the script:

```
./mcdonald.awk input.txt
```

2. Write an awk script to swap the second and third column of the a csv file of below given format(also try writing a sed script for the same, might need back referencing for it).

Note : After swapping the columns, each field should be space separated.

Input->

UserID	Name	Designation	loginShell
10001	ajay	manager	/bin/false
1001	sunil	clerk	/bin/false
101	varun	manager	Valid
60123	amit	manager	/bin/false
401	tarun	peon	Valid

Output->

```
UserID Designation Name loginShell
10001 manager ajay /bin/false
1001 clerk sunil /bin/false
101 manager varun Valid
60123 manager amit /bin/false
401 peon tarun Valid
```

Usage : ./swap_accounts.awk sample.txt

Sed:

1. Given an HTML file, we want to redirect all its URLs to www.google.com. We know that URLs will start within <a href=“”> tag with http:// or https:// or directly domain name. Write a code that will replace all URLs to www.google.com. This requires backward referencing, which is useful to know, has many usecases, like the example I covered in the slides involving "Welcome To The Course CS104". A bit complex and might be time consuming, but give a try.

Ex :

Input:

```
<html>
<body>
<a href="www.example.com/about">About Us</a>
<a href="https://www.example.com/contact">Contact Us</a>
<a href="http://blog.example.com/latest">Latest News</a>
</body>
</html>
```

Output:

```
<html>
<body>
<a href="www.google.com">About Us</a>
<a href="https://www.google.com">Contact Us</a>
<a href="http://www.google.com">Latest News</a>
</body>
```

</html>

Usage:- sed -i -f html.sed sample.html

2. You don’t need to know python for this, except for the fact that in python 2, code that has print statements uses the format print “text” , while in Python3 the print format is print(“text”).

Write a sed expression that takes a python2 input code and converts the print statements within to python3 print format.

Example:

Input:-
```
import numpy as np

if __name__== ‘__main__’:
                a = np.arange(5)
                print a
                b = np.arange(10)
                print b
                print “Hello”
```

Output:-
```
import numpy as np

if __name__== ‘__main__’:
                a = np.arange(5)
                print(a)
                b = np.arange(10)
                print(b)
                print(“Hello”)
```

Usage :- sed -i -f python23.sed script.py

3. Suppose you need to extract the month from a sequence of ID codes in a file 'input.csv', where each line is of the form: N...CCCMMMDD
Where N... is one or more numerical digits, CCC is 3 alphabetic characters, MMM is a 3 character month abbreviation, and DD is a 2 digit date.
Write a sed expression to output the MMM from a number of rows of this form, using the sed command

Example:

Input :-

```
1385ABCAPR19
9CHDMAR28
324598GLRMAY01
12346JULJUN07
8975JUNJUL10
774SATOCT31
```

Output:-

```
APR
MAR
MAY
JUN
JUL
OCT
```

Usage:- `sed -i -f extract-month.sed input.csv`

4. Ever felt the requirement of copying contents of a script or a ".csv" along with line numbers?? (even if not, you will need it here ;))

Task : You are given a "employment.csv" file and you need to modify the file to have line numbers. (You might need to pipe output of one sed command to another).
Well, this is somewhat boring..., let's REMOVE the header of the file and print only entries of "High_industry" and "Value", separated by ':' and double quotes
around entries of "High_industry" (see example for clarity)
You are allowed to use pipe('|') for this activity but brownie points for not using it.

Eg :
Input ->

```
Week_end,Indicator,High_industry,Value
2019-05-05,Number of paid jobs - 34 days,Film,2090110
2019-05-05,Number of paid jobs - 34 days,Agriculture,95150
2019-05-05,Number of paid jobs - 34 days,Banking,405050
2019-05-05,Number of paid jobs - 34 days,Manufacturing,1570740
....
```

Output ->

```
1 "Film":2090110
2 "Agriculture":95150
3 "Banking":405050
4 "Manufacturing":1570740
```

....

Usage :

```
script.sh > output.txt
```