Exercise 1 SecondScript.sh:

```
#!/bin/bash
# script example list files in current directory
ls
```

Exercise 2 CountScript.sh:

```
#!/bin/bash
# script counting the lines in a file providing full path to file
wc -l home/manager/course_data/NGS_file_formats_and_data_QC/SRR19504912_1.fastq
```

Exercise 3 SecondScript.sh modified so that the path to a directory is provided as input

```
#!/bin/bash
# providing the path of a directory as an input
path=$1
ls $path
```

Exercise 4 CountScript.sh modified

```
#!/bin/bash
# counting the lines in any input file provided
file=$1
wc -l $file
```

Run using `./CountScript.sh reference.fasta`

Exercise 5 modifying HelloToYou.sh so that it takes two arguments

```
#!/bin/bash
# taking two input from the command line
a=$1
b=$2
c="$a $b"
echo "Hello $c"
```

Run using

```
./HelloToYou.sh Johann Mastropiero
```

Exercise 6 modifying the CountScript.sh to take two inputs (the pair of files)

```
#!/bin/bash
# counting the lines in two input files provided
file1=$1
file2=$2
file=”$file1 $file2”
#works beacuse wc take multiple files as input
#if not you have to set up a “for done” loop
wc -l $file
```

Run using

```
./CountScript.sh /home/manager/course_data/NGS_file_formats_and_data_QC/SRR19504912_1.fastq
/home/manager/course_data/NGS_file_formats_and_data_QC/SRR19504912_2.fastq
```

Exercise 7 modifying GetPairName.sh so the user provides the input

```
#!/bin/bash
filename1=$1
filename2=${filename1%_1.fq}_2.fq
echo $filename2
sample1=sample${filename1#SRR}
echo $sample1
```

Exercise 8 modifying GetPairName.sh to check that the input has _1.fastq at the end

```
#!/bin/bash
filename1=$1
end=${filename1: -8}
if [ "$end" == "_1.fastq" ];then
  filename2=${filename1%_1.fastq}_2.fastq
  echo $filename2
fi
```

Exercise 9 writing a Loop2.sh script that loops through fastq_sets and copies the files to your current directory

```
#!/bin/bash
for f in fastq_sets/*.fastq
do
  cp $f .
done
```

Run this script from outside the fastq_sets with the following command

```
./Loop2.sh
```

Exercise 10

```
#!/bin/bash
for f in fastq_sets/*.fastq
do
  mv $f ${f%.fastq}.fq
done
```

Run this script from outside the fastq_sets with the following command

```
./Loop3.sh
```

Exercise 11 loop through the fastq_sets and if the file has _1.fq, then count the number of lines in the file

```
#!/bin/bash
# counting the lines for _1.fq in fastq_setsfor filename in fastq_sets/*.fq
# the if – then – fi statement is nested within a for – do – done loop
for filename in fastq_sets/*.fq
do
  end=${filename: -5}
  if [ "$end" == "_1.fq" ];then
    wc -l $filename
  fi
done
```

Run this script from outside the fastq_sets with the following command

```
./CountScript.sh
```

Bonus track, but with it will require your improvement

```
#!/bin/bash
# perfor the mapping as we did in class
# improvement required:
# 1. Change filenames by variables and read their values from comand line argument
# 2. Use string manipulations to gess the name of trim_galore output
# 3. Use string manipulations to create an output file name for hte stats
# 4. Include bamqc quality controls
# 5. Max-level improvement. Wrap up the script in a loop to pass a directory
#                           as a command line argument, and made the script
#                           scan all paired .fq.gz files to map them
# clean de reads. Improve adding qc before and after trimming.
trim_galore -q 25 --length 50 --paired chikv.read2.fq.gz chikv.read1.fq.gz
# index the reference genome. Improve adding an if to bypass this if already done
bwa index chikv-genome.fasta
# map the reads from trimmed fq.gz files
bwa mem chikv-genome.fasta chikv*val_?.fq.gz > chikv-aln.sam
# convert sam to bam
samtools view -bS chikv-aln.sam > chikv-aln.bam
# sort and index the bamfile
samtools sort chikv-aln.bam -o chikv-aln.sorted.bam
samtools index chikv-aln.sorted.bam
# create a recycle-bin like folder to put files that can be deleted if all goes OK.
# improve checkin if it already exist before creation
mkdir safe-to-delete
# move files that should be deleted later
mv dengue-aln.sam safe-to-delete
mv dengue-aln.bam safe-to-delete
# print the stats. Improve adding redirection to a result file.
samtools view -c -F4 chikv-aln.sorted.bam
samtools view -c -f4 chikv-aln.sorted.bam
samtools idxstats chikv-aln.sorted.bam
samtools stats chikv-aln.sorted.bam > chikv_stats.txt
# to be done, include the bamqc control.
```