1. $ cp hello.sh average.sh
   $ nano average.sh
   $ ./average.sh 2 6 10

```
#!/bin/bash
#initialize average and total sum
avg=0
total=0
number_of_args=$#
# Check if enough arguments given
if [ $# -lt 2 ] ; then
  echo -e "I need at least 2 command line args\n"
  echo -e "Syntax: $0: number1 number2 ... numberN\n"
  break
fi
#calculate the average of numbers given on command line as cmd args
for i in $*
 do
  #total sum of all the numbers
  total=`expr $total + $i `
 done
#calculate average from total sum and number_of_args
avg=`expr $total / $number_of_args `
echo "Average value of the numbers is $avg"
```

2. $ cp hello.sh reverse.sh
   $ nano reverse.sh
   $ ./reverse.sh 123456

```
#!/bin/bash
# condition to inform user how the script should be running
#e.g. too many parameters, etc.
if [ $# -ne 1 ]
then
   echo "Execute: $0   number"
   echo "        Script will find reverse of given positive integer"
   echo "        For eg. $0 123, it will print 321"
   exit 1
fi
number=$1
#initialize reversed number and
rev=0
while [ $number -gt 0 ]
do
rev=`expr $rev \* 10 + $number % 10`
number=`expr $number / 10`
#if you wish to follow intermediate steps, you can print it the loop
```

```
#e.g. uncomment next line
#echo $rev $number
done
echo  "Reverse number is $rev"
```

3.
```
$ cp hello.sh factorial.sh
$ nano factorial.sh
$ ./factorial.sh

#!/bin/bash
#factorial n! is the product of all positive integers less than or equal to n.
#For example 5 ! = 5 × 4 × 3 × 2 × 1 = 120.

#initialize factorial and the number
n=0; rn=0
fact=1
echo -n "Enter number to find factorial : "
read n

#remember the value of read number
rn=$n
#iterate over every number starting from $n
until [ $n -eq  0 ]
do
 fact=`expr $fact \* $n`
 n=`expr $n - 1`
done
echo "$rn! = $fact"
```

4.
```
$ cp hello.sh sort.sh
$ nano sort.sh
$ ./sort.sh 2 9 5 11 0

#!/bin/bash
if [ $# -ne 5 ]
then
   echo "Usage: $0   num1 num2 num3 num4 num5"
   echo "Numbers will be sorted"
   exit 1
fi
# Declare the array of 5 subscripts to hold 5 numbers
numbers=($1 $2 $3 $4 $5)
# Print the number before sorting process
echo "Original (not yet sorted) numbers in array:"
for (( i = 0; i <= 4; i++ ))
do
 echo ${numbers[$i]}
done
```

```
#Sort the numbers
for (( i = 0; i <= 4 ; i++ ))
do
#iterate all the numbers for comparison for each number
#make sure that numbers already sorted are not taken any more
  for (( j = $i; j <= 4; j++ ))
  do
#substitute current number by the smallest number left
#(number not considered before)
     if [ ${numbers[$i]} -gt ${numbers[$j]}  ]; then
         temp=${numbers[$i]}
         numbers[$i]=${numbers[$j]}
         numbers[$j]=$temp
     fi
  done
done
# Print the sorted numbers
echo -e "\nSorted numbers (ascending order):"
for (( i=0; i <= 4; i++ ))
do
  echo ${numbers[$i]}
done
```