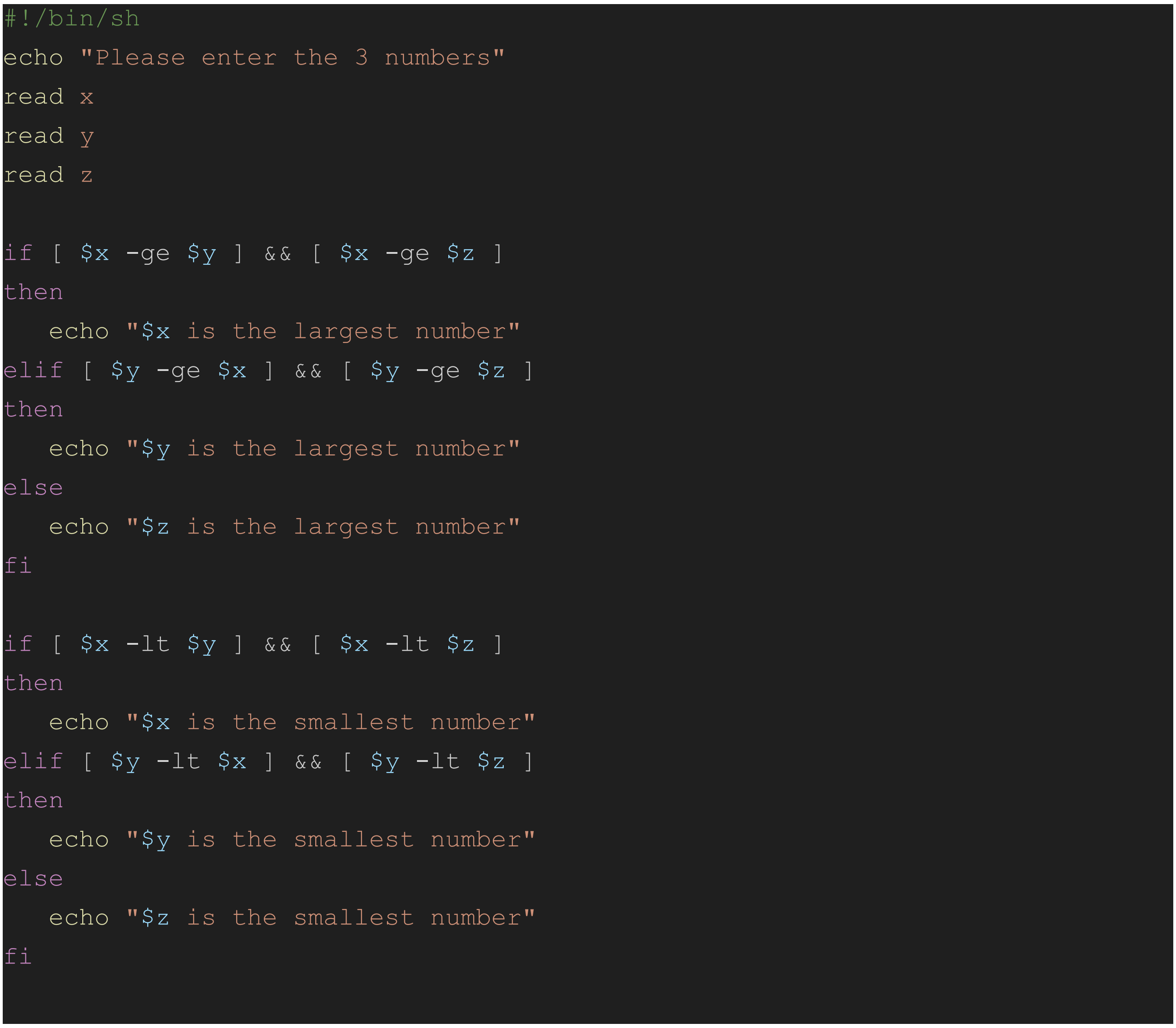
Linux - Unix Shell Programming Lab Programs

—---

A1. Implement a shell program to find and display largest and smallest of three numbers

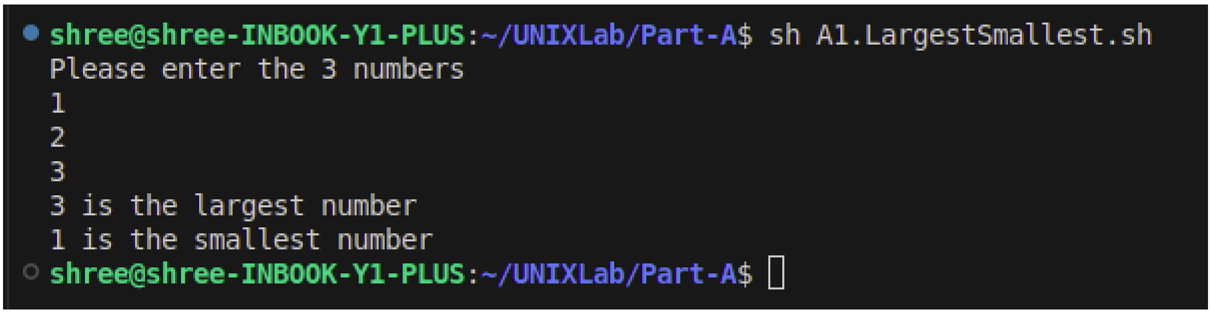
A1.LargestSmallest.sh



```
#!/bin/sh
echo "Please enter the 3 numbers"
read x
read y
read z

if [ $x -ge $y ] && [ $x -ge $z ]
then
   echo "$x is the largest number"
elif [ $y -ge $x ] && [ $y -ge $z ]
then
   echo "$y is the largest number"
else
   echo "$z is the largest number"
fi

if [ $x -lt $y ] && [ $x -lt $z ]
then
   echo "$x is the smallest number"
elif [ $y -lt $x ] && [ $y -lt $z ]
then
   echo "$y is the smallest number"
else
   echo "$z is the smallest number"
fi
```



```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A1.LargestSmallest.sh
Please enter the 3 numbers
1
2
3
3 is the largest number
1 is the smallest number
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$
```

A2. Find the number n is divisible by m or not using shell script. Where m and n are supplied as command line argument or read from key board interactively

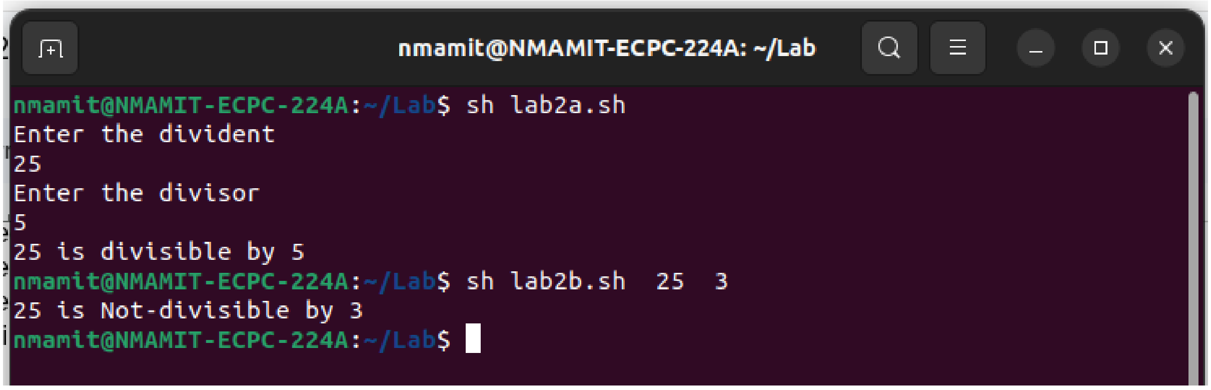
lab2a.sh

```
#!/bin/sh
echo "Enter the divident"
read n
echo "Enter the divisor"
read m

y=` expr $n % $m `
if [ $y -eq 0 ]
then
   echo "$n is divisible by $m"
else
   echo "$n is Not-divisible by $m"
fi
```

lab2b.sh using command Line arguments

```
#!/bin/sh
y=` expr $1 % $2 `
if [ $y -eq 0 ]
then
   echo "$1 is divisible by $2"
else
   echo "$1 is Not-divisible by $2"
fi
```



```
nmamit@NMAMIT-ECPC-224A:~/Lab$ sh lab2a.sh
Enter the divident
25
Enter the divisor
5
25 is divisible by 5
nmamit@NMAMIT-ECPC-224A:~/Lab$ sh lab2b.sh  25  3
25 is Not-divisible by 3
nmamit@NMAMIT-ECPC-224A:~/Lab$ 
```

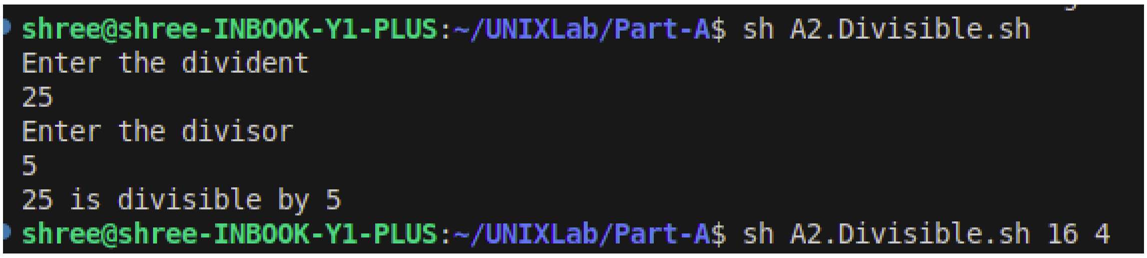
A2Divisible.sh Using Both CommandLine And Keyboard input

```
#!/bin/bash

if [ $# -eq 2 ]
then
   n=$1
   m=$2
else
   echo "Enter the divident"
   read n
   echo "Enter the divisor"
   read m
fi

y=`expr $n % $m`

if [ $y -eq 0 ]
then
   echo "$n is divisible by $m"
else
   echo "$n is NOT divisible by $m"
fi
```



```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A2.Divisible.sh
Enter the divident
25
Enter the divisor
5
25 is divisible by 5
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A2.Divisible.sh 16 4
```

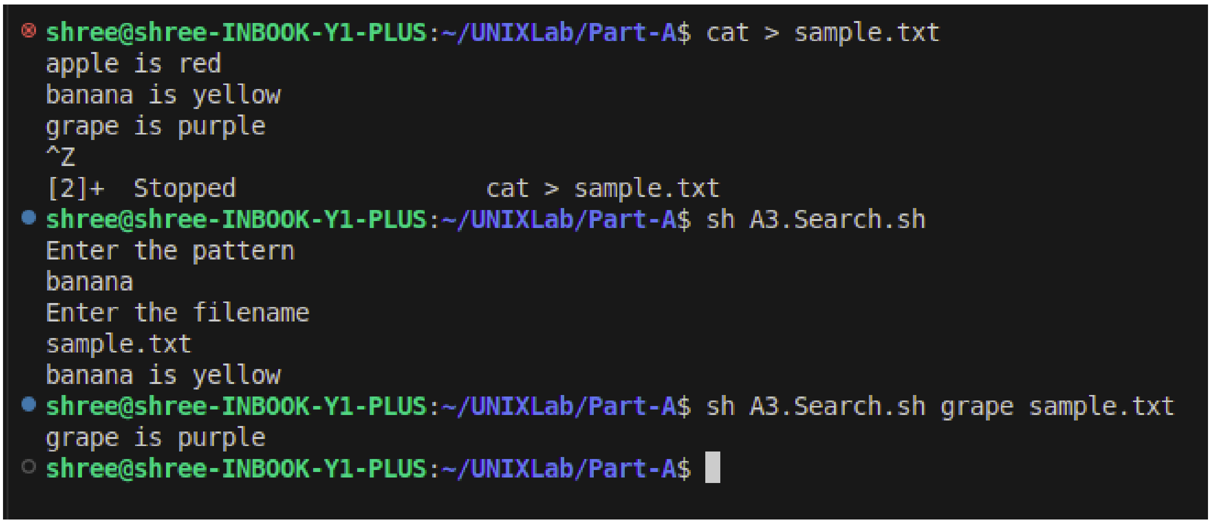
A3. Plan and implement a shell program to search a pattern in a file that will take both pattern and file name from the command line arguments.

A3.Pattern.sh

```
#!/bin/sh
if [ $# -eq 0 ]
then
   echo "No arguments"
else
   grep "$1" "$2"
fi
```

A3.Search.sh

```
#!/bin/sh
if [ $# -eq 0 ]
then
   echo "Enter the pattern"
   read pattern
   echo "Enter the filename"
   read filename
else
   pattern=$1
   filename=$2
fi
sh A3.Pattern.sh $pattern $filename
```



```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ cat > sample.txt
apple is red
banana is yellow
grape is purple
^Z
[2]+  Stopped                 cat > sample.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A3.Search.sh
Enter the pattern
banana
Enter the filename
sample.txt
banana is yellow
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A3.Search.sh grape sample.txt
grape is purple
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ 
```

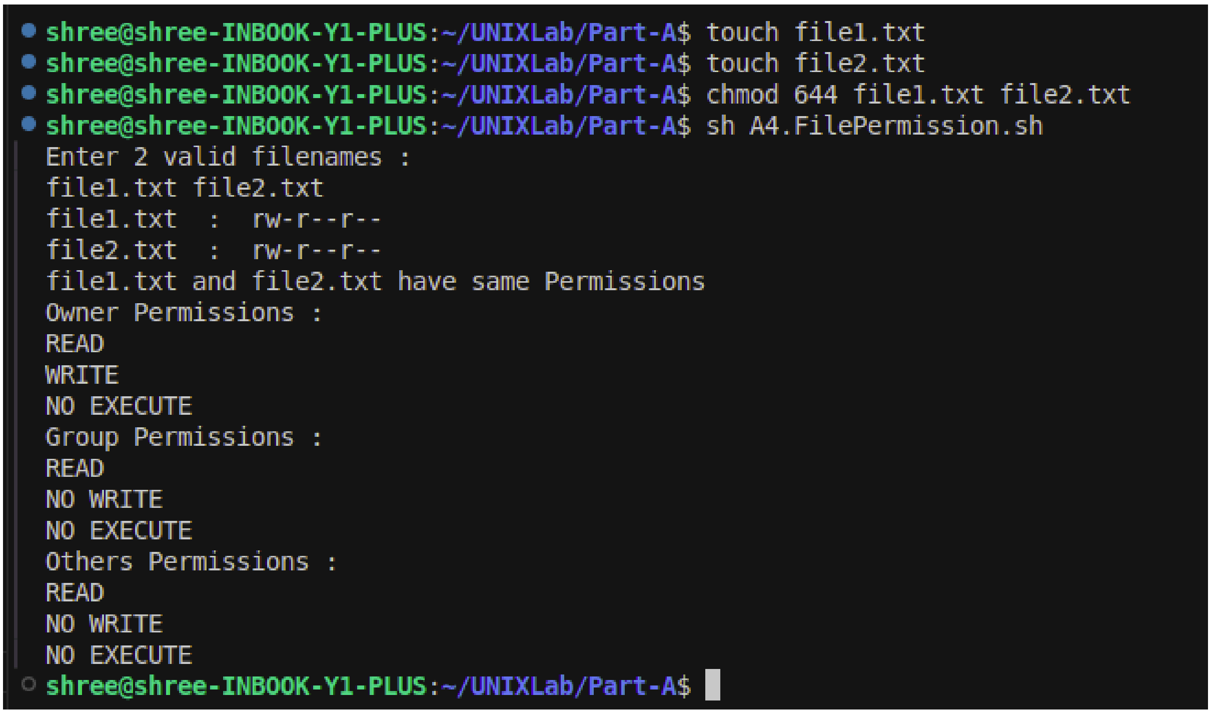
A4. Design a shell program that takes two file names, checks the permissions for these files are identical and if they are identical, output the common permissions; otherwise output each file name followed by its permissions.

A4.FilePermission.sh

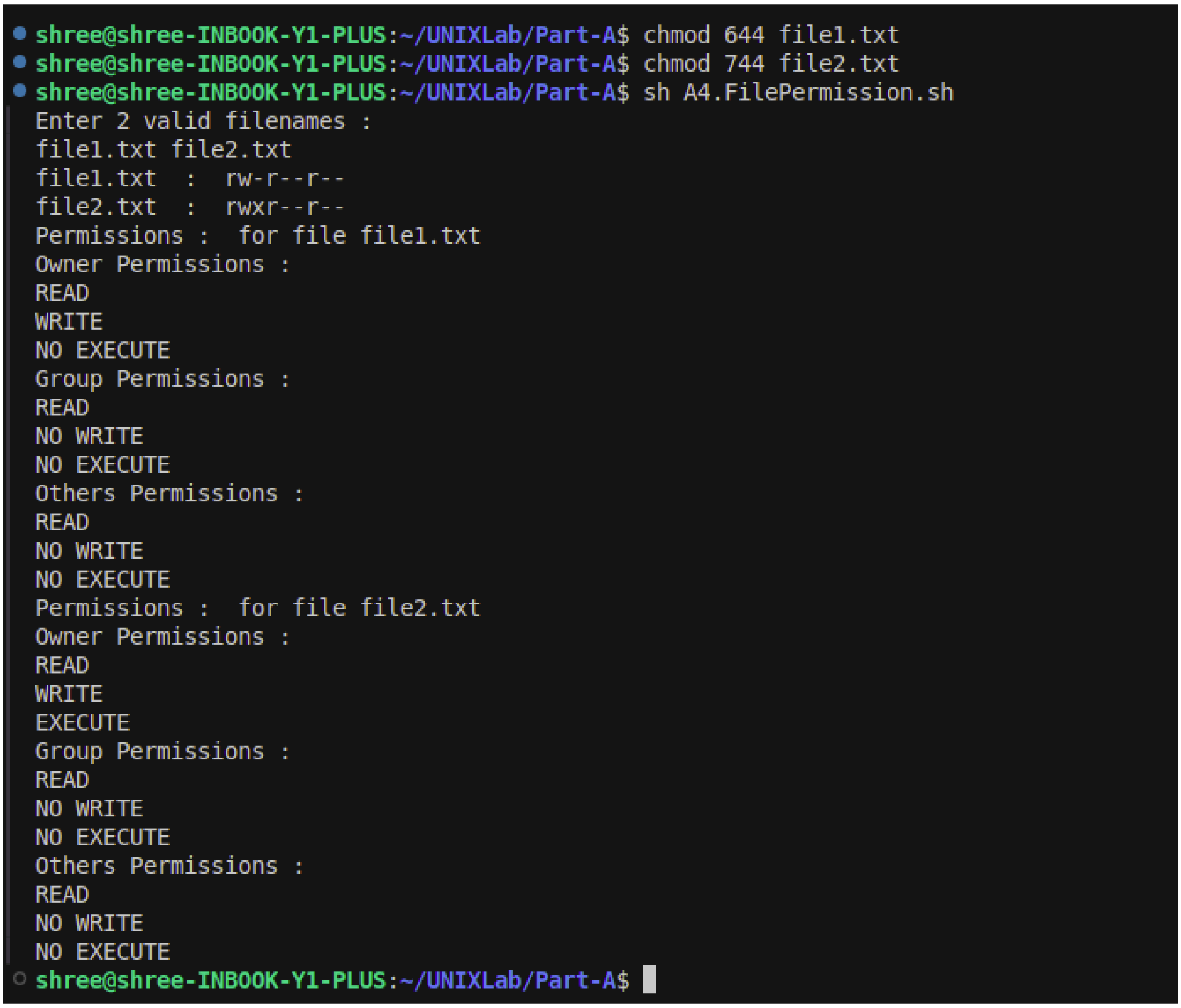
```
#!/bin/sh
display_perm()
{
  r=` ls -l $1 | cut -c 2 `
  w=` ls -l $1 | cut -c 3 `
  x=` ls -l $1 | cut -c 4 `
  echo "Owner Permissions : "
  if [ "$r" = "r" ]
  then
      echo "READ"
  else
      echo "NO READ"
  fi
  if [ "$w" = "w" ]
  then
      echo "WRITE"
  else
      echo "NO WRITE"
  fi
  if [ "$x" = "x" ]
  then
      echo "EXECUTE"
  else
      echo "NO EXECUTE"
  fi
  r=` ls -l $1 | cut -c 5 `
  w=` ls -l $1 | cut -c 6 `
  x=` ls -l $1 | cut -c 7 `
  echo "Group Permissions : "
  if [ "$r" = "r" ]
  then
      echo "READ"
  else
      echo "NO READ"
  fi
  if [ "$w" = "w" ]
  then
      echo "WRITE"
  else
      echo "NO WRITE"
  fi
  if [ "$x" = "x" ]
  then
      echo "EXECUTE"
  else
      echo "NO EXECUTE"
  fi
  r=` ls -l $1 | cut -c 8 `
  w=` ls -l $1 | cut -c 9 `
  x=` ls -l $1 | cut -c 10 `
  echo "Others Permissions : "
  if [ "$r" = "r" ]
  then
      echo "READ"
  else
      echo "NO READ"
  fi
  if [ "$w" = "w" ]
  then
      echo " WRITE"
  else
      echo "NO WRITE"
  fi
  if [ "$x" = "x" ]
  then
      echo "EXECUTE"
  else
      echo "NO EXECUTE"
  fi
}


echo "Enter 2 valid filenames : "
read f1 f2
if [ -e "$f1" -a -e "$f2" ]
then
  p1=` ls -l $f1 | cut -c 2-10 `
  p2=` ls -l $f2 | cut -c 2-10 `
  echo "$f1  :  $p1"
  echo "$f2  :  $p2"
  if [ "$p1" = "$p2" ]
  then
```

```
      echo "$f1 and $f2 have same Permissions"
      display_perm $f1
  else
      echo "Permissions :  for file $f1"
      display_perm $f1
      echo "Permissions :  for file $f2"
      display_perm $f2
  fi
else
  echo  "Invalid filenames"
fi
exit
```

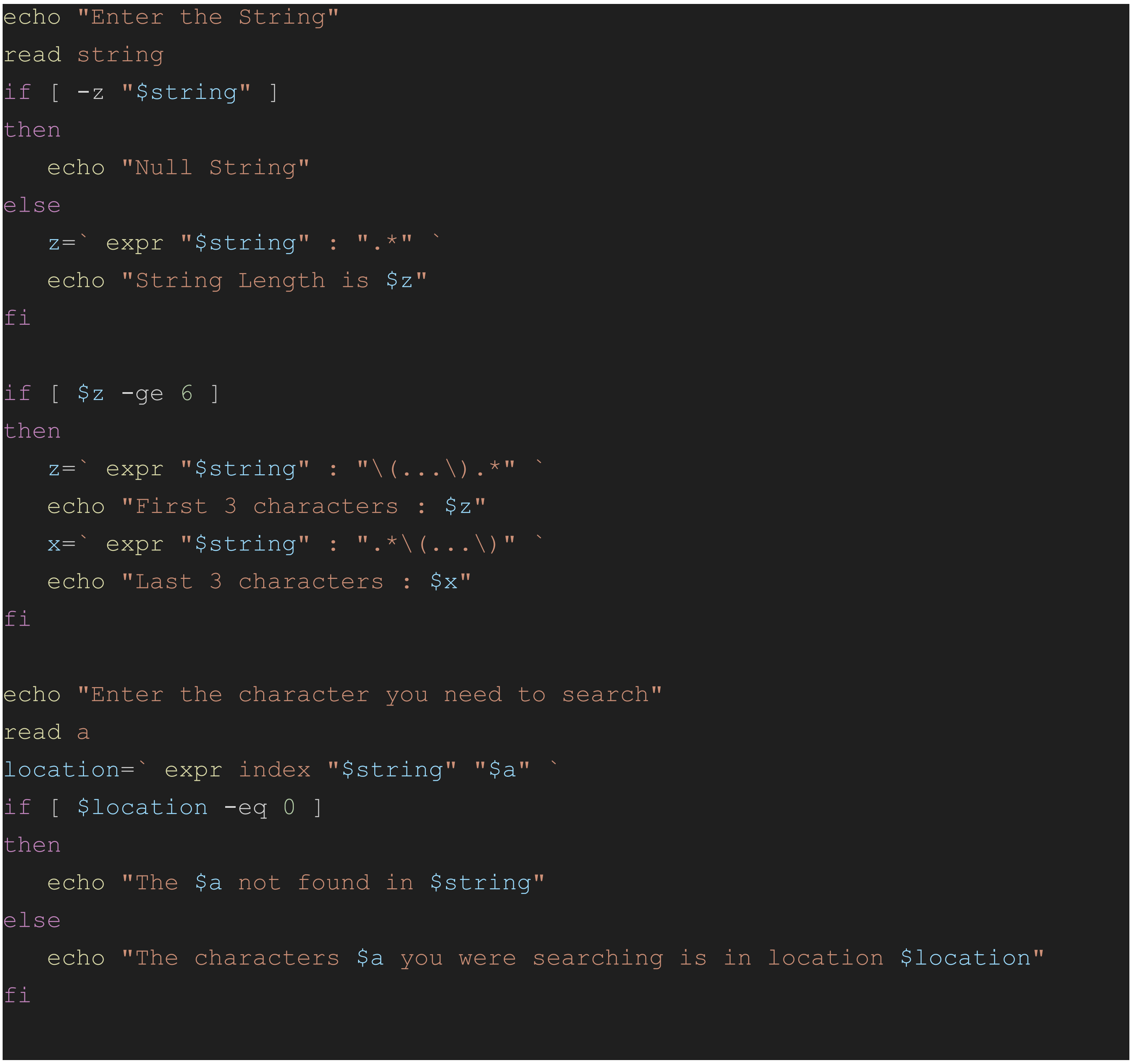


```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ touch file1.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ touch file2.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ chmod 644 file1.txt file2.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A4.FilePermission.sh
Enter 2 valid filenames :
file1.txt file2.txt
file1.txt  :  rw-r--r--
file2.txt  :  rw-r--r--
file1.txt and file2.txt have same Permissions
Owner Permissions :
READ
WRITE
NO EXECUTE
Group Permissions :
READ
NO WRITE
NO EXECUTE
Others Permissions :
READ
NO WRITE
NO EXECUTE
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$
```



```
● shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ chmod 644 file1.txt
● shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ chmod 744 file2.txt
● shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A4.FilePermission.sh
  Enter 2 valid filenames :
  file1.txt file2.txt
  file1.txt  :  rw-r--r--
  file2.txt  :  rwxr--r--
  Permissions :  for file file1.txt
  Owner Permissions :
  READ
  WRITE
  NO EXECUTE
  Group Permissions :
  READ
  NO WRITE
  NO EXECUTE
  Others Permissions :
  READ
  NO WRITE
  NO EXECUTE
  Permissions :  for file file2.txt
  Owner Permissions :
  READ
  WRITE
  EXECUTE
  Group Permissions :
  READ
  NO WRITE
  NO EXECUTE
  Others Permissions :
  READ
  NO WRITE
  NO EXECUTE
○ shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$
```

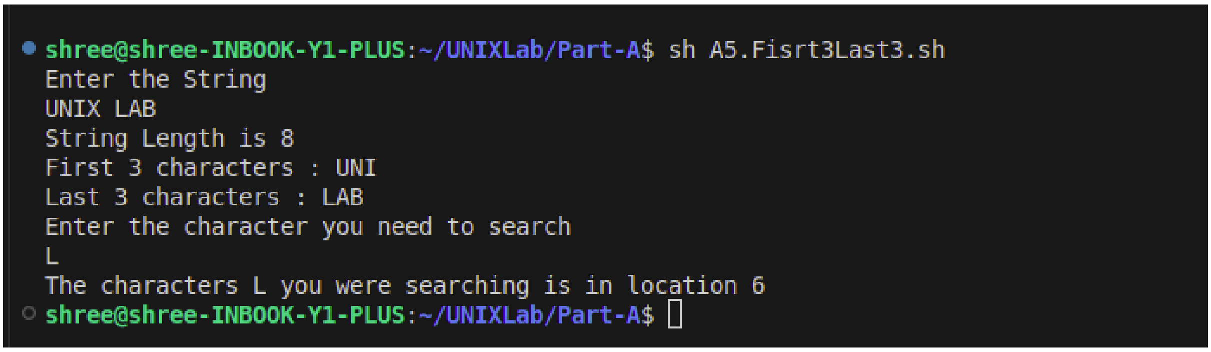
A5. Develop a shell script that performs following string handling operations
i) Calculate the length of the string
ii) Locate a position of a character in a string
iii) Extract last three characters from string
iv) Extract first three characters from the string



```
echo "Enter the String"
read string
if [ -z "$string" ]
then
   echo "Null String"
else
   z=` expr "$string" : ".*" `
   echo "String Length is $z"
fi

if [ $z -ge 6 ]
then
   z=` expr "$string" : "\(...\).*" `
   echo "First 3 characters : $z"
   x=` expr "$string" : ".*\(...\)" `
   echo "Last 3 characters : $x"
fi

echo "Enter the character you need to search"
read a
location=` expr index "$string" "$a" `
if [ $location -eq 0 ]
then
   echo "The $a not found in $string"
else
   echo "The characters $a you were searching is in location $location"
fi
```



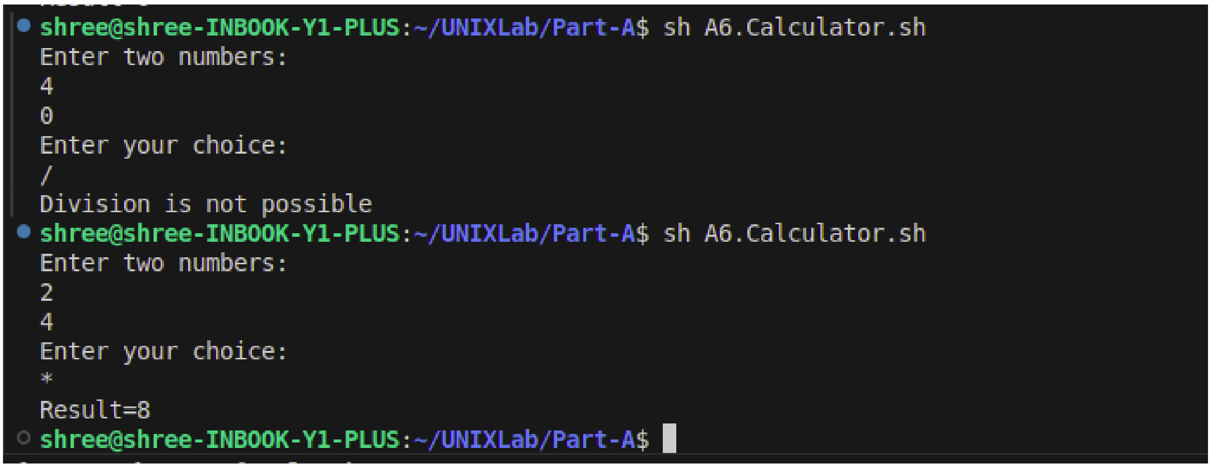
```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A5.Fisrt3Last3.sh
Enter the String
UNIX LAB
String Length is 8
First 3 characters : UNI
Last 3 characters : LAB
Enter the character you need to search
L
The characters L you were searching is in location 6
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$
```

A6. Write a shell program to implement simple calculator operations.

[A6.Calculator.sh]

```
#!/bin/sh
echo "Enter two numbers:"
read a
read b
echo "Enter your choice:"
read ch

case $ch in
   '+') y=`expr $a + $b`
        echo "Result=$y"
        ;;
   '-') y=`expr $a - $b`
        echo "Result=$y"
        ;;
   '*') y=`expr $a \* $b`
        echo "Result=$y"
        ;;
   '/') if [ $b -eq 0 ]
        then
           echo "Division is not possible"
        else
           y=`expr $a / $b`
           echo "Result=$y"
        fi
        ;;
    *)  echo "Invalid choice"
        ;;
esac
```



```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A6.Calculator.sh
Enter two numbers:
4
0
Enter your choice:
/
Division is not possible
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A6.Calculator.sh
Enter two numbers:
2
4
Enter your choice:
*
Result=8
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ 
```

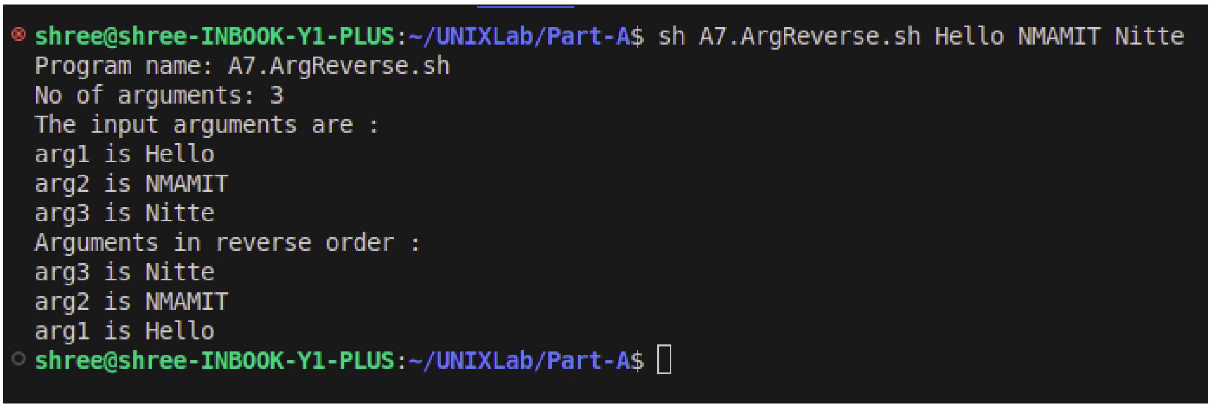
A7. Design a Shell Program that takes any number of arguments and prints them in the same order and in reverse order with suitable messages.

A7.ArgReverse.sh

```
#!/bin/sh
echo "Program name: $0"
if [ $# -eq 0 ]
  then exit
Fi

echo "No of arguments: $#"
echo "The input arguments are : "
num=1
for i in "$@"
do
  echo "arg$num is $i"
  num=`expr $num + 1`
done

echo "Arguments in reverse order : "
num=$#
while [ $num -ne 0 ]
do
  eval echo "arg$num is \$$num"
  num=`expr $num - 1`
done
```



```
⊗ shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A7.ArgReverse.sh Hello NMAMIT Nitte
  Program name: A7.ArgReverse.sh
  No of arguments: 3
  The input arguments are :
  arg1 is Hello
  arg2 is NMAMIT
  arg3 is Nitte
  Arguments in reverse order :
  arg3 is Nitte
  arg2 is NMAMIT
  arg1 is Hello
○ shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ ▯
```

A8. For the given path names (E.g., a/b,a/b/c), design a shell script to create all the components in that path names as directories.

A8.CreateDir.sh

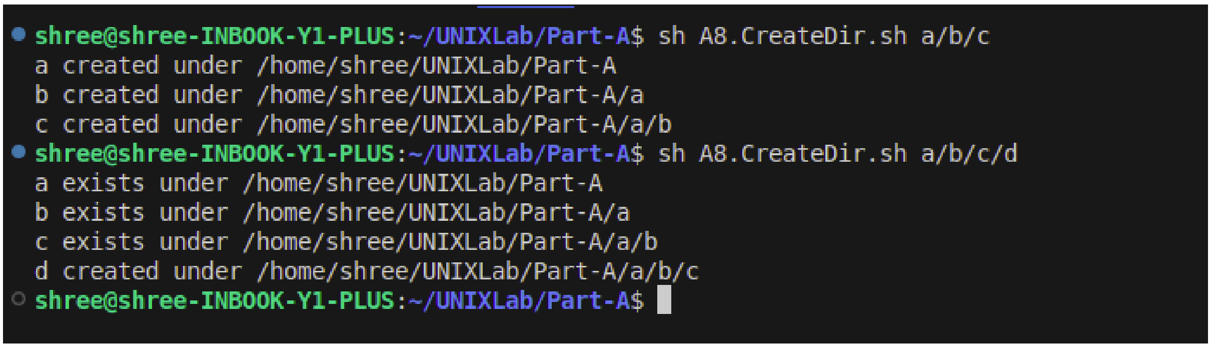
```
#!/bin/sh

if [ $# -ne 1 ]
then
  echo "no arguments"
  exit 1
fi

curdir=$(pwd)

for dir in $(echo "$1" | tr "/" " ")
do
  if [ -d "$dir" ]
  then
      echo "$dir exists under $(pwd)"
      cd "$dir"
  else
      mkdir "$dir"
      echo "$dir created under $(pwd)"
      cd "$dir"
  fi
done

cd "$curdir"
```

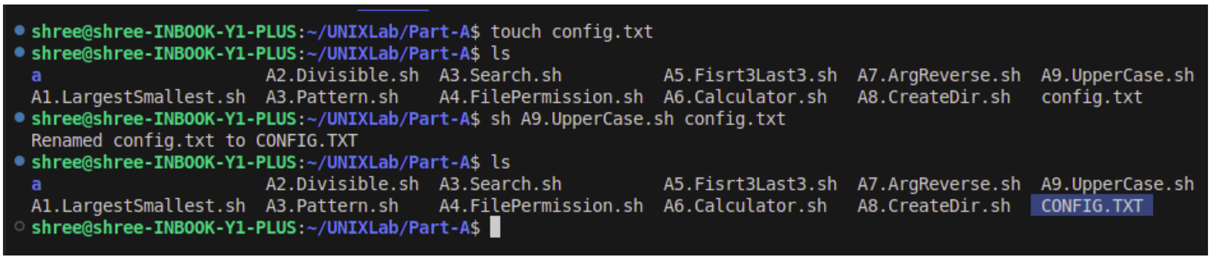


```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A8.CreateDir.sh a/b/c
a created under /home/shree/UNIXLab/Part-A
b created under /home/shree/UNIXLab/Part-A/a
c created under /home/shree/UNIXLab/Part-A/a/b
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A8.CreateDir.sh a/b/c/d
a exists under /home/shree/UNIXLab/Part-A
b exists under /home/shree/UNIXLab/Part-A/a
c exists under /home/shree/UNIXLab/Part-A/a/b
d created under /home/shree/UNIXLab/Part-A/a/b/c
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$
```

A9. For every filename, check whether a file exists in the current directory or not and then convert its name to uppercase only if a file with a new name doesn’t exist using shell script.

A9.UpperCase.sh

```
#!/bin/sh
for file in "$@"
do
  if [ -f "$file" ]
  then
      upper=$(echo "$file" | tr '[a-z]' '[A-Z]')
      if [ -f "$upper" ]
      then
          echo "$upper already exists"
      else
          mv "$file" "$upper"
          echo "Renamed $file to $upper"
      fi
  else
      echo "$file does not exist"
  fi
done
```



```
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ touch config.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ ls
a                      A2.Divisible.sh  A3.Search.sh          A5.Fisrt3Last3.sh  A7.ArgReverse.sh  A9.UpperCase.sh
A1.LargestSmallest.sh  A3.Pattern.sh    A4.FilePermission.sh  A6.Calculator.sh   A8.CreateDir.sh   config.txt
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ sh A9.UpperCase.sh config.txt
Renamed config.txt to CONFIG.TXT
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ ls
a                      A2.Divisible.sh  A3.Search.sh          A5.Fisrt3Last3.sh  A7.ArgReverse.sh  A9.UpperCase.sh
A1.LargestSmallest.sh  A3.Pattern.sh    A4.FilePermission.sh  A6.Calculator.sh   A8.CreateDir.sh   CONFIG.TXT
shree@shree-INBOOK-Y1-PLUS:~/UNIXLab/Part-A$ 
```