EmbedUR Systems Private Limited

Bash Shell Scripting Training

Module-5

By:

A.Buvaneshkumaar

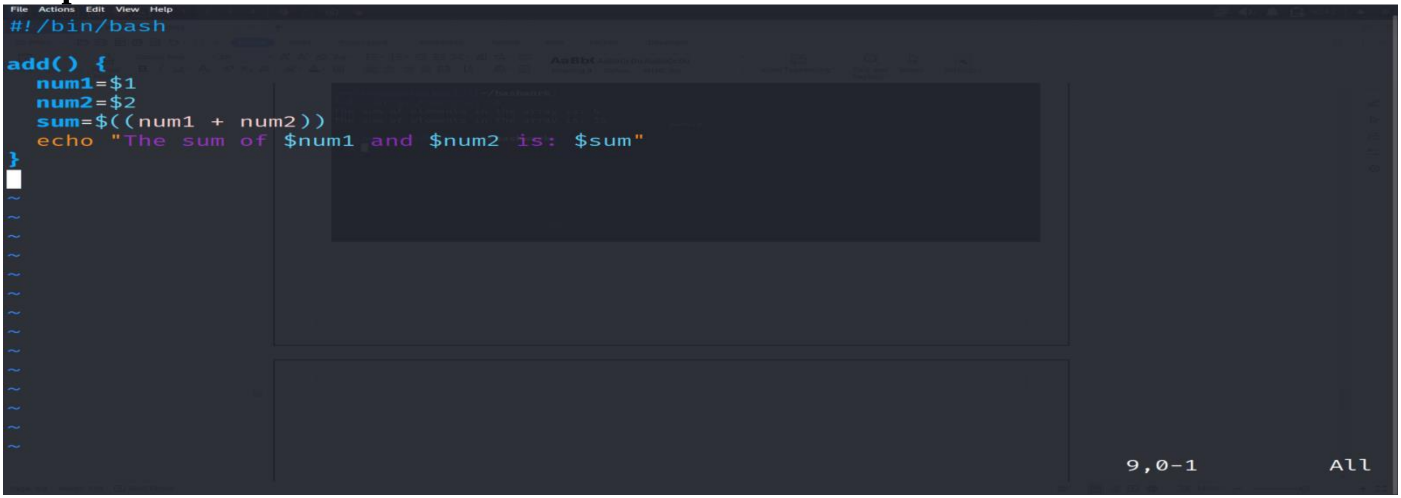
Mepco Schlenk Engineering College

Sivakasi,Virudhunagar District.,

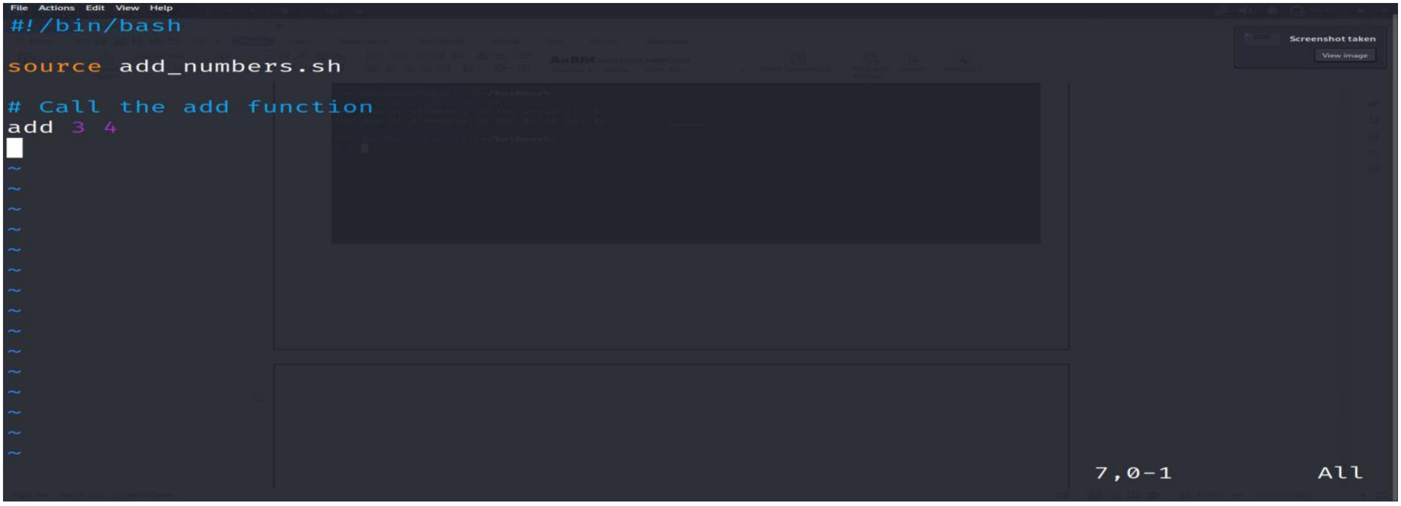
Advance topics in a function

1) Write a function add to add two numbers and call the function in another file.

script:

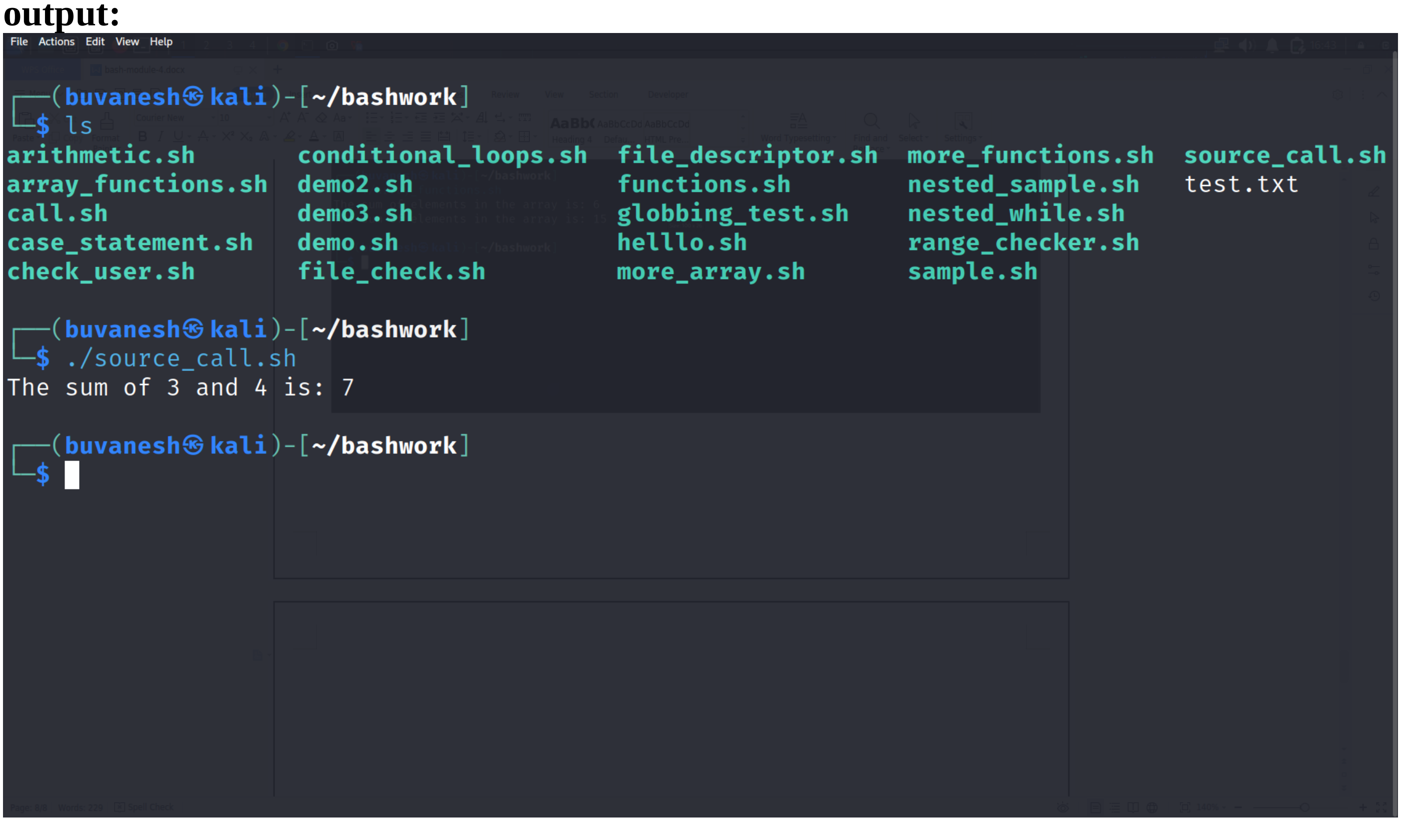


```
#!/bin/bash

add() {
   num1=$1
   num2=$2
   sum=$((num1 + num2))
   echo "The sum of $num1 and $num2 is: $sum"
}

~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                   9,0-1         All
```



```
#!/bin/bash

source add_numbers.sh

# Call the add function
add 3 4

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                   7,0-1         All
```

output:

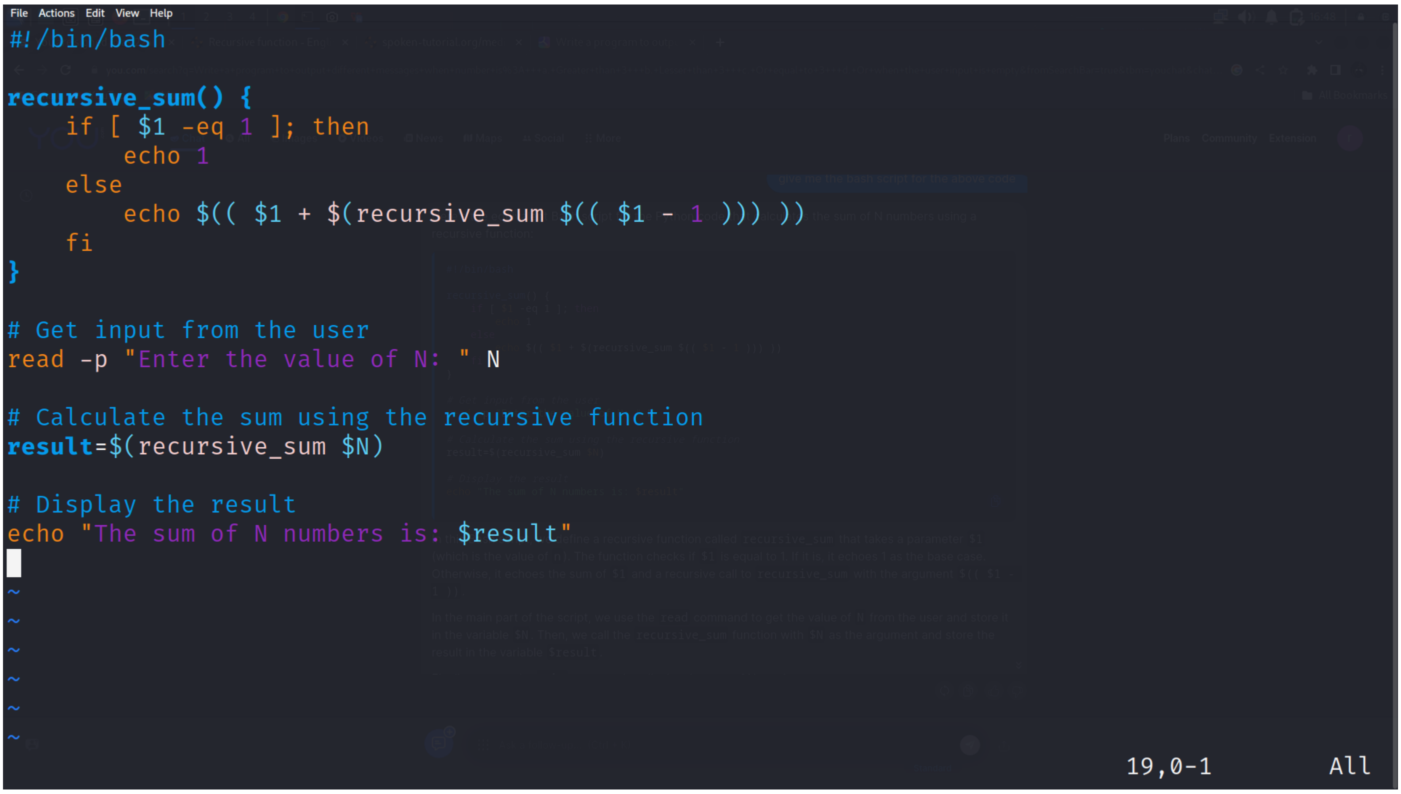


```
┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ls
arithmetic.sh        conditional_loops.sh  file_descriptor.sh  more_functions.sh  source_call.sh
array_functions.sh   demo2.sh              functions.sh        nested_sample.sh   test.txt
call.sh              demo3.sh              globbing_test.sh    nested_while.sh
case_statement.sh    demo.sh               helllo.sh           range_checker.sh
check_user.sh        file_check.sh         more_array.sh       sample.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ./source_call.sh
The sum of 3 and 4 is: 7

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ 
```

Recursive function

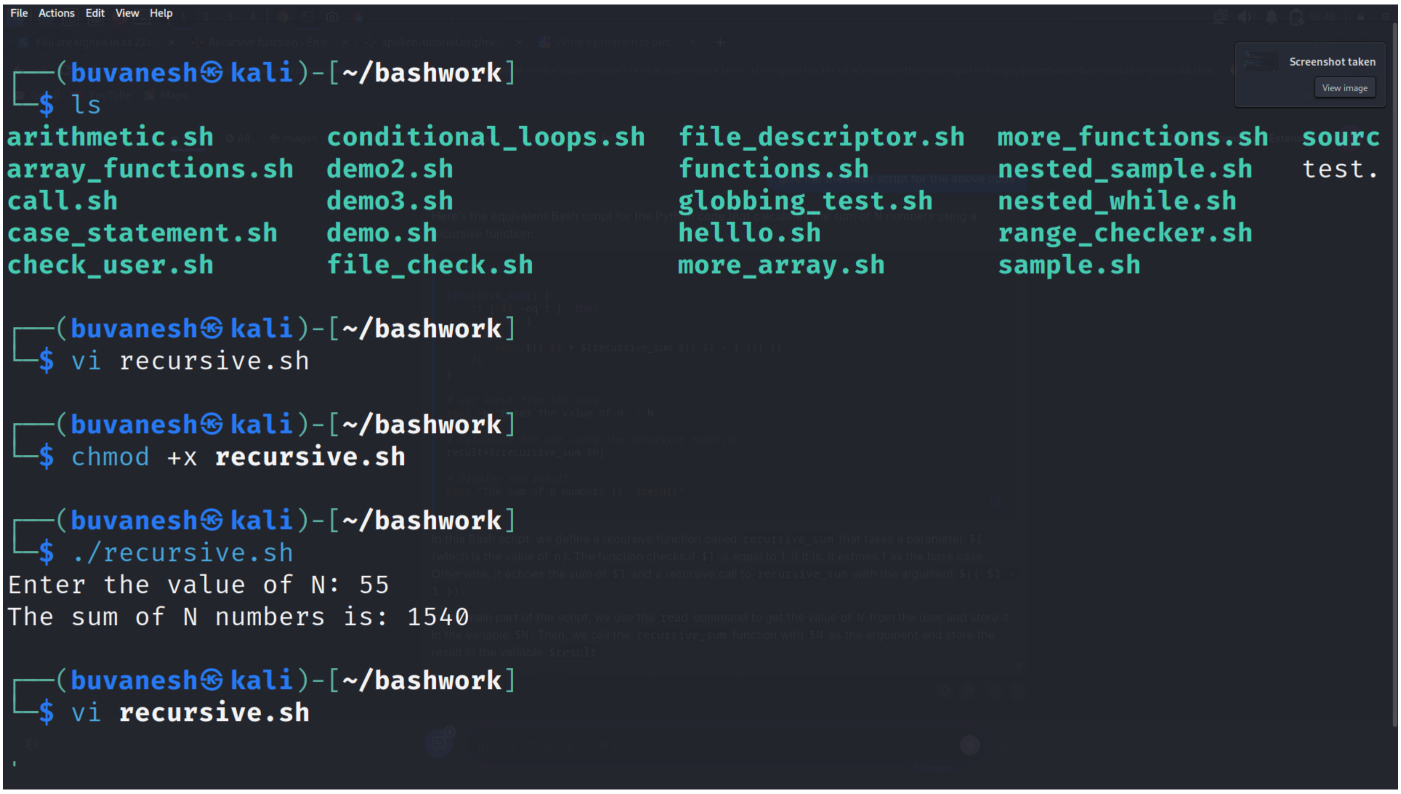
1) Write a program where the recursive function calculates the sum of N numbers

script:



```
#!/bin/bash

recursive_sum() {
    if [ $1 -eq 1 ]; then
        echo 1
    else
        echo $(( $1 + $(recursive_sum $(( $1 - 1 ))) ))
    fi
}

# Get input from the user
read -p "Enter the value of N: " N

# Calculate the sum using the recursive function
result=$(recursive_sum $N)

# Display the result
echo "The sum of N numbers is: $result"

~
~
~
~
~
~
                                                                    19,0-1        All
```

output:

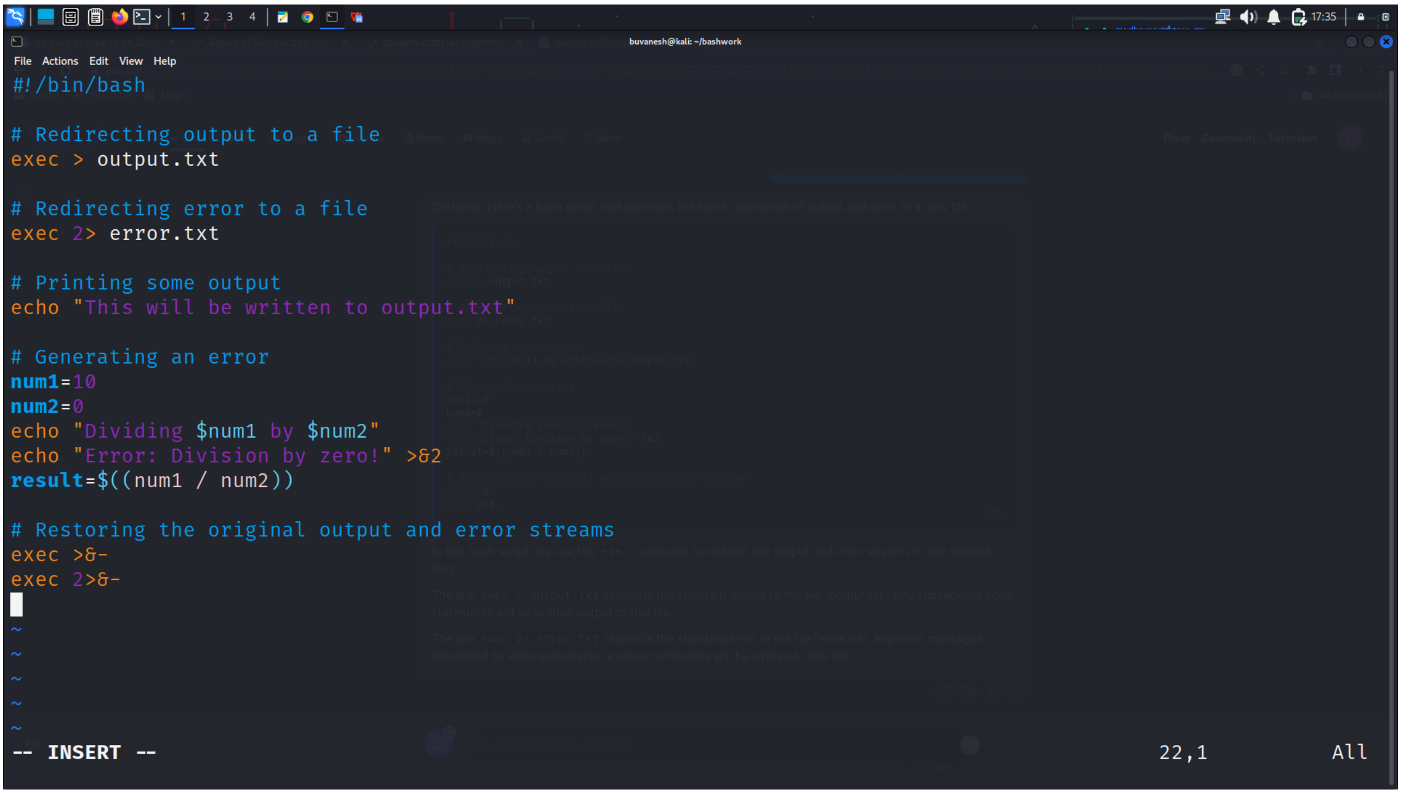


```
┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ls
arithmetic.sh       conditional_loops.sh  file_descriptor.sh  more_functions.sh  sourc
array_functions.sh  demo2.sh              functions.sh        nested_sample.sh   test.
call.sh             demo3.sh              globbing_test.sh    nested_while.sh
case_statement.sh   demo.sh               helllo.sh           range_checker.sh
check_user.sh       file_check.sh         more_array.sh       sample.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi recursive.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ chmod +x recursive.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ./recursive.sh
Enter the value of N: 55
The sum of N numbers is: 1540

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi recursive.sh
```

Basics of Redirection (error handling)

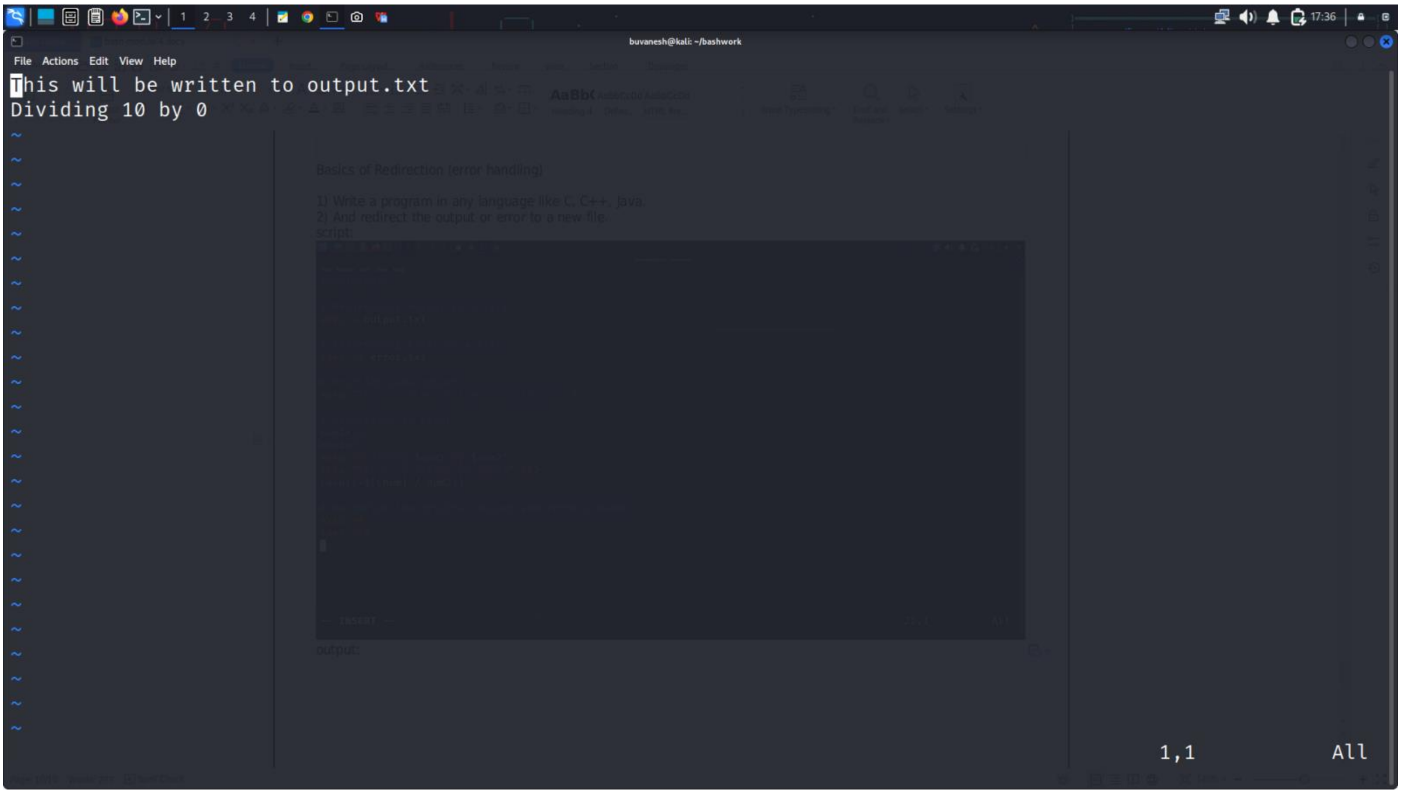
1) Write a program in any language like C, C++, Java.
2) And redirect the output or error to a new file.

script:

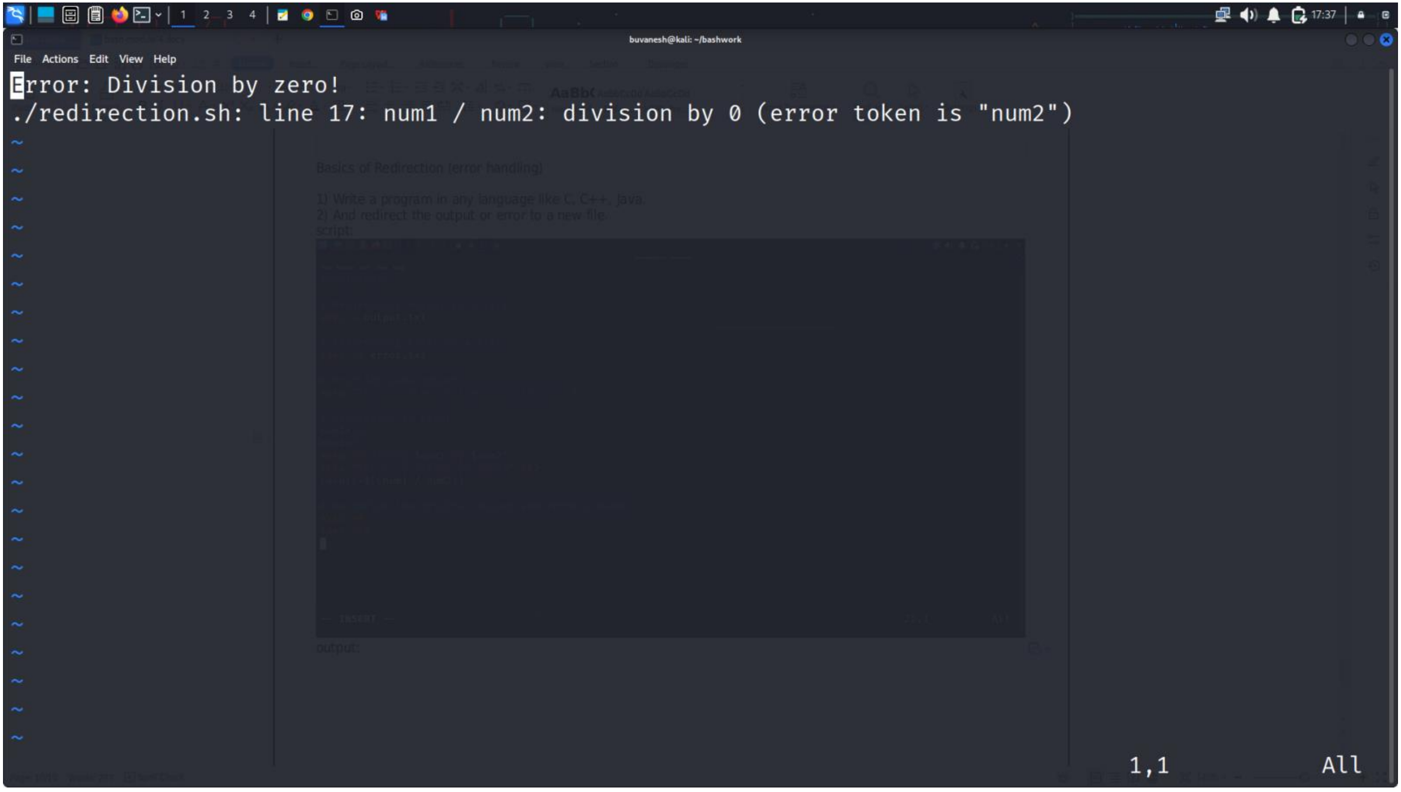


```
#!/bin/bash

# Redirecting output to a file
exec > output.txt

# Redirecting error to a file
exec 2> error.txt

# Printing some output
echo "This will be written to output.txt"

# Generating an error
num1=10
num2=0
echo "Dividing $num1 by $num2"
echo "Error: Division by zero!" >&2
result=$((num1 / num2))

# Restoring the original output and error streams
exec >&-
exec 2>&-

~
~
~
~
~
-- INSERT --                                         22,1          All
```

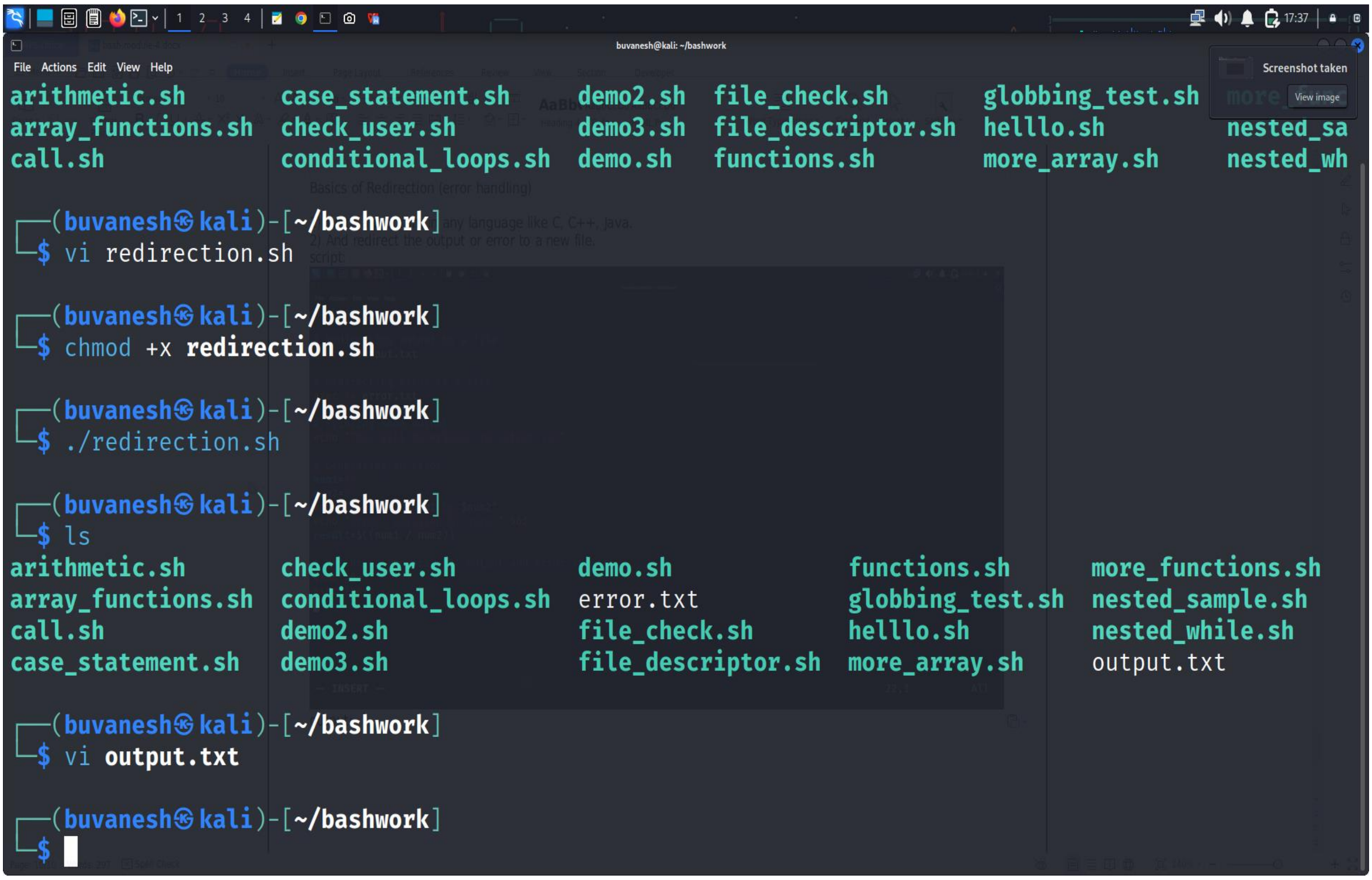
output:



```
This will be written to output.txt
Dividing 10 by 0
~
~
~
                                                     1,1           All
```



```
Error: Division by zero!
./redirection.sh: line 17: num1 / num2: division by 0 (error token is "num2")
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                                       1,1          All
```

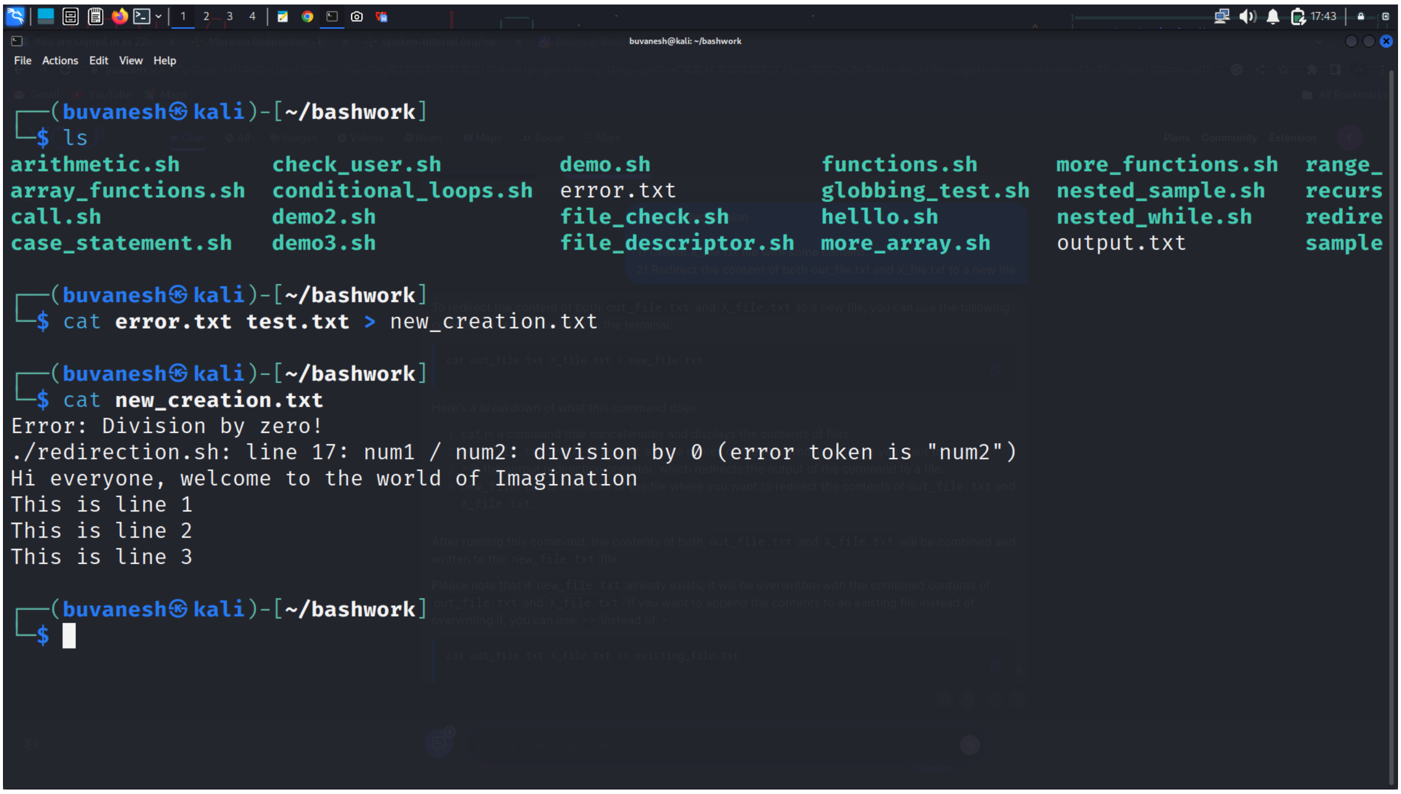


```
arithmetic.sh       case_statement.sh    demo2.sh  file_check.sh       globbing_test.sh
array_functions.sh  check_user.sh        demo3.sh  file_descriptor.sh  helllo.sh         nested_sa
call.sh             conditional_loops.sh demo.sh   functions.sh        more_array.sh     nested_wh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi redirection.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ chmod +x redirection.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ./redirection.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ls
arithmetic.sh       check_user.sh         demo.sh             functions.sh      more_functions.sh
array_functions.sh  conditional_loops.sh  error.txt           globbing_test.sh  nested_sample.sh
call.sh             demo2.sh              file_check.sh       helllo.sh         nested_while.sh
case_statement.sh   demo3.sh              file_descriptor.sh  more_array.sh     output.txt

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi output.txt

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ 
```

More on Redirection

1) Create X_file.txt file with some content.
2) Redirect the content of both out_file.txt and X_file.txt to a new file

output:

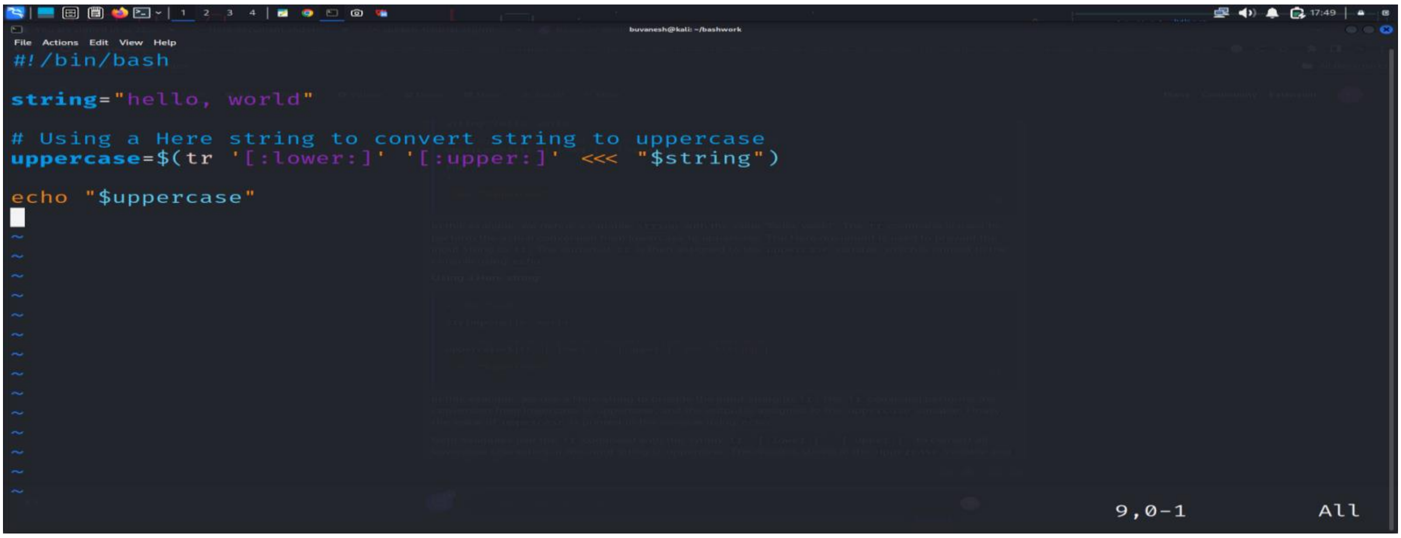


```
┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ls
arithmetic.sh        check_user.sh         demo.sh              functions.sh        more_functions.sh   range_
array_functions.sh   conditional_loops.sh  error.txt            globbing_test.sh    nested_sample.sh    recurs
call.sh              demo2.sh              file_check.sh        helllo.sh           nested_while.sh     redire
case_statement.sh    demo3.sh              file_descriptor.sh   more_array.sh       output.txt          sample

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ cat error.txt test.txt > new_creation.txt

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ cat new_creation.txt
Error: Division by zero!
./redirection.sh: line 17: num1 / num2: division by 0 (error token is "num2")
Hi everyone, welcome to the world of Imagination
This is line 1
This is line 2
This is line 3

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ 
```

Here document and Here string

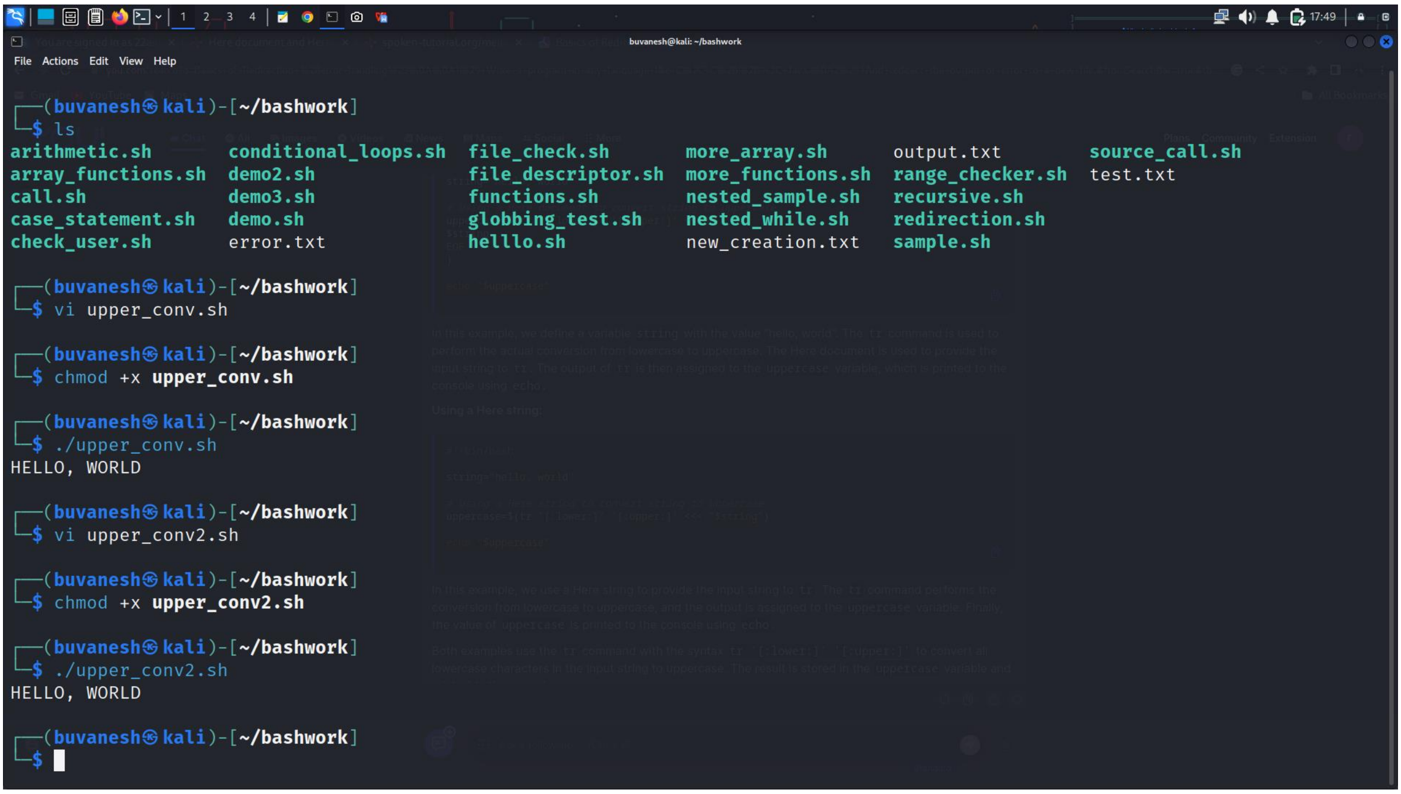
1) Convert a string to uppercase using:
   a) Here document
   b) Here string
   Hint: tr a-z A-Z

script:



```
#!/bin/bash

string="hello, world"

# Using a Here string to convert string to uppercase
uppercase=$(tr '[:lower:]' '[:upper:]' <<< "$string")

echo "$uppercase"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                         9,0-1         All
```

output:



```
┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ls
arithmetic.sh       conditional_loops.sh  file_check.sh       more_array.sh      output.txt        source_call.sh
array_functions.sh  demo2.sh              file_descriptor.sh  more_functions.sh  range_checker.sh  test.txt
call.sh             demo3.sh              functions.sh        nested_sample.sh   recursive.sh
case_statement.sh   demo.sh               globbing_test.sh    nested_while.sh    redirection.sh
check_user.sh       error.txt             helllo.sh           new_creation.txt   sample.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi upper_conv.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ chmod +x upper_conv.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ./upper_conv.sh
HELLO, WORLD

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ vi upper_conv2.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ chmod +x upper_conv2.sh

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ ./upper_conv2.sh
HELLO, WORLD

┌──(buvanesh㉿kali)-[~/bashwork]
└─$ 
```