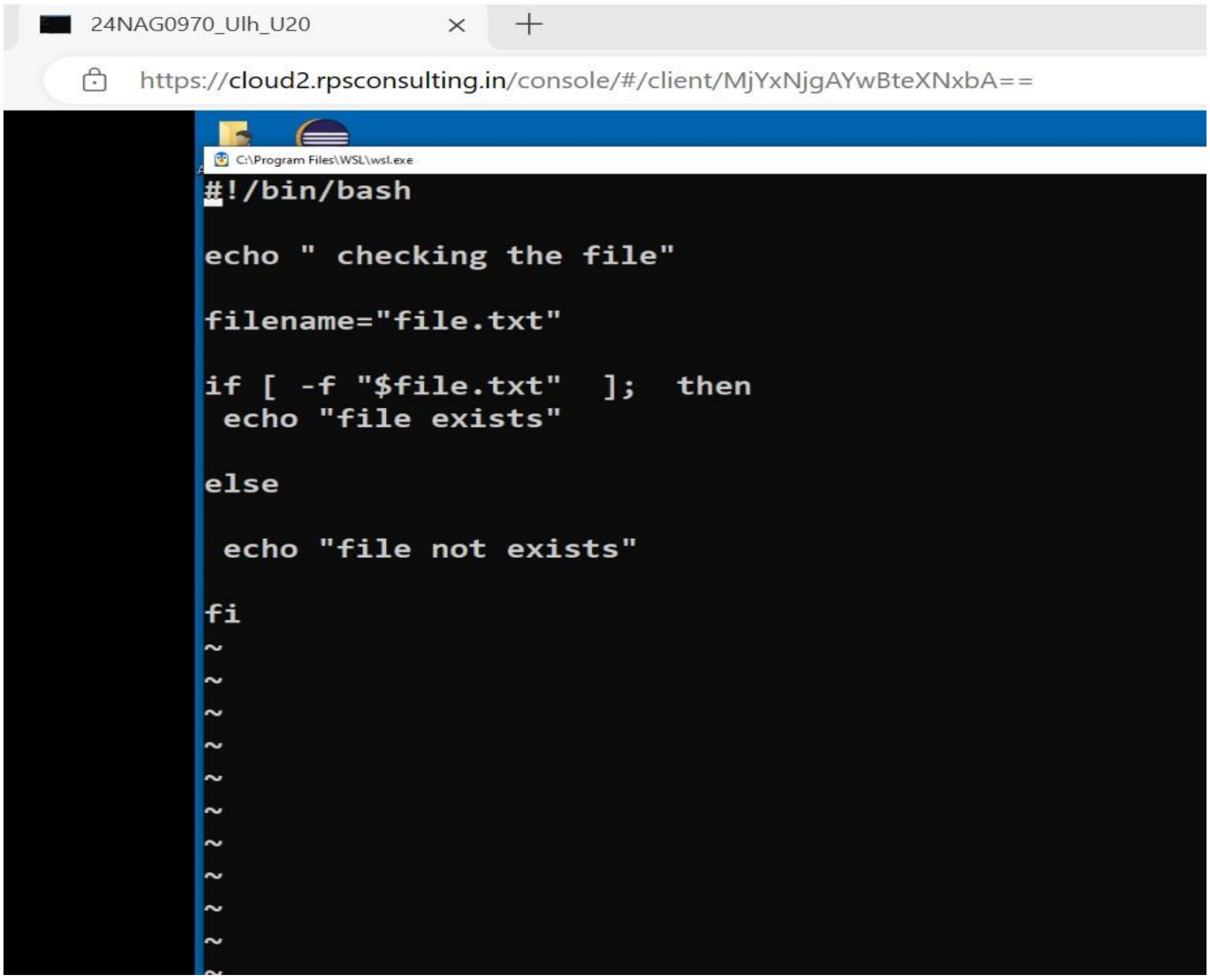
K DEEPAK **ASSIGNMENT(DAY-7)**

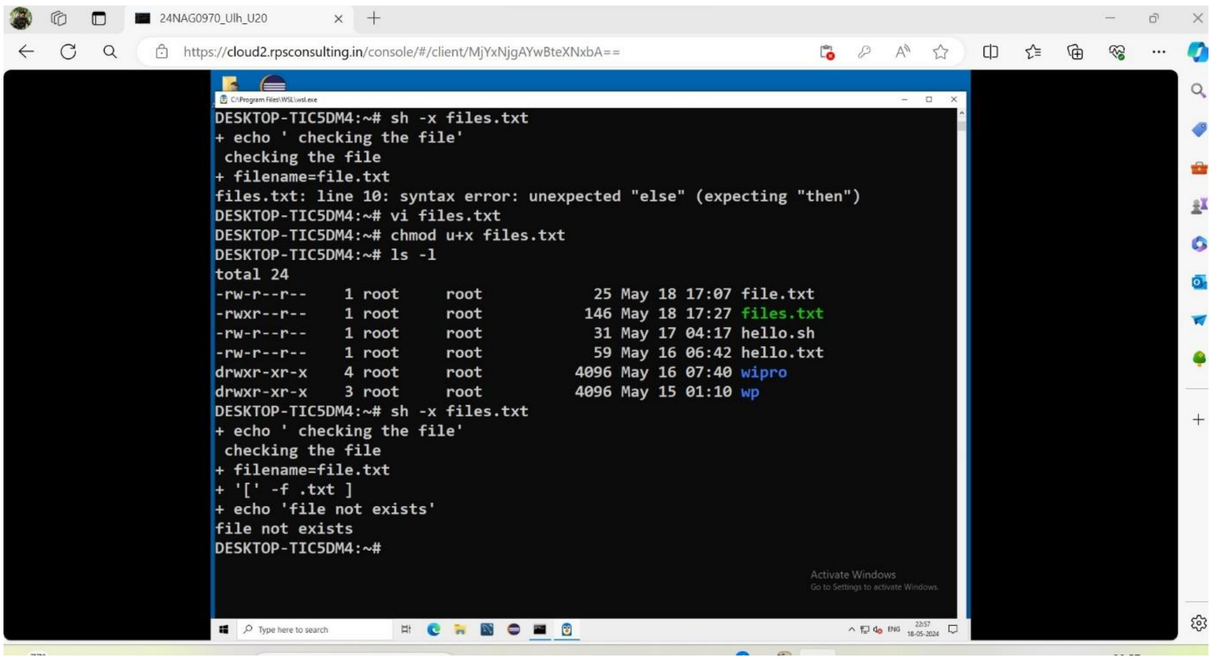
1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

- First we create a file or if we already have an exist file which is used check the file is exist or not exists in directory.
- In this I created a file with file.txt using vi editor and saved.
- Creating the main program to see the file exist or not exist and the named as files.txt in the cloud of vi editor
- Then we write script using file name and storing the name of the file in variable filename .
- Implementing the IF ELSE to check the file is exist or not exists and adding the image of the execution


24NAG0970_Ulh_U20
https://cloud2.rpsconsulting.in/console/#/client/MjYxNjgAYwBteXNxbA==
C:\Program Files\WSL\wsl.exe

```
#!/bin/bash

echo " checking the file"

filename="file.txt"

if [ -f "$file.txt"  ];  then
 echo "file exists"

else

 echo "file not exists"

fi
~
~
~
~
~
~
~
~
~
~
```

- Here we wrote the script in files.txt

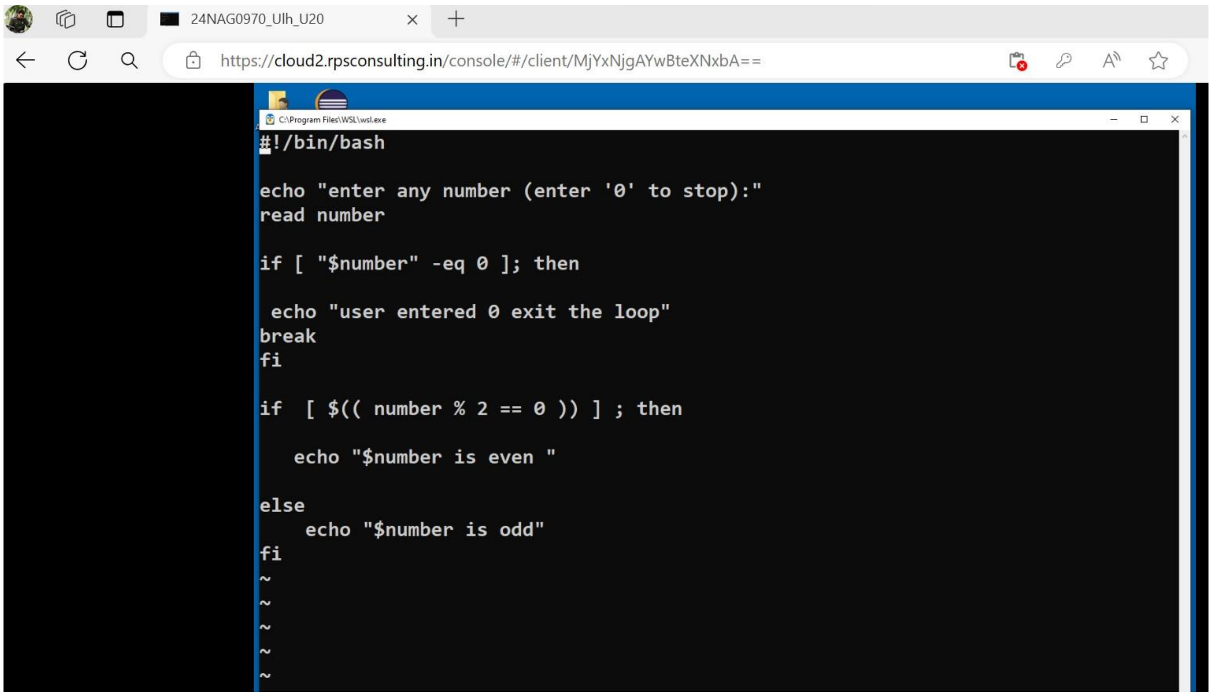

24NAG0970_Ulh_U20
https://cloud2.rpsconsulting.in/console/#/client/MjYxNjgAYwBteXNxbA==

```
DESKTOP-TIC5DM4:~# sh -x files.txt
+ echo ' checking the file'
 checking the file
+ filename=file.txt
files.txt: line 10: syntax error: unexpected "else" (expecting "then")
DESKTOP-TIC5DM4:~# vi files.txt
DESKTOP-TIC5DM4:~# chmod u+x files.txt
DESKTOP-TIC5DM4:~# ls -l
total 24
-rw-r--r--    1 root     root            25 May 18 17:07 file.txt
-rwxr--r--    1 root     root           146 May 18 17:27 files.txt
-rw-r--r--    1 root     root            31 May 17 04:17 hello.sh
-rw-r--r--    1 root     root            59 May 16 06:42 hello.txt
drwxr-xr-x    4 root     root          4096 May 16 07:40 wipro
drwxr-xr-x    3 root     root          4096 May 15 01:10 wp
DESKTOP-TIC5DM4:~# sh -x files.txt
+ echo ' checking the file'
 checking the file
+ filename=file.txt
+ '[' -f .txt ]
+ echo 'file not exists'
file not exists
DESKTOP-TIC5DM4:~#
```

- After saving the files.txt changing which is used execute the script.
- Ls -l is used check the file is stored in the directory.
- Sh -x filename used execute command of the filename and the above script is successful.

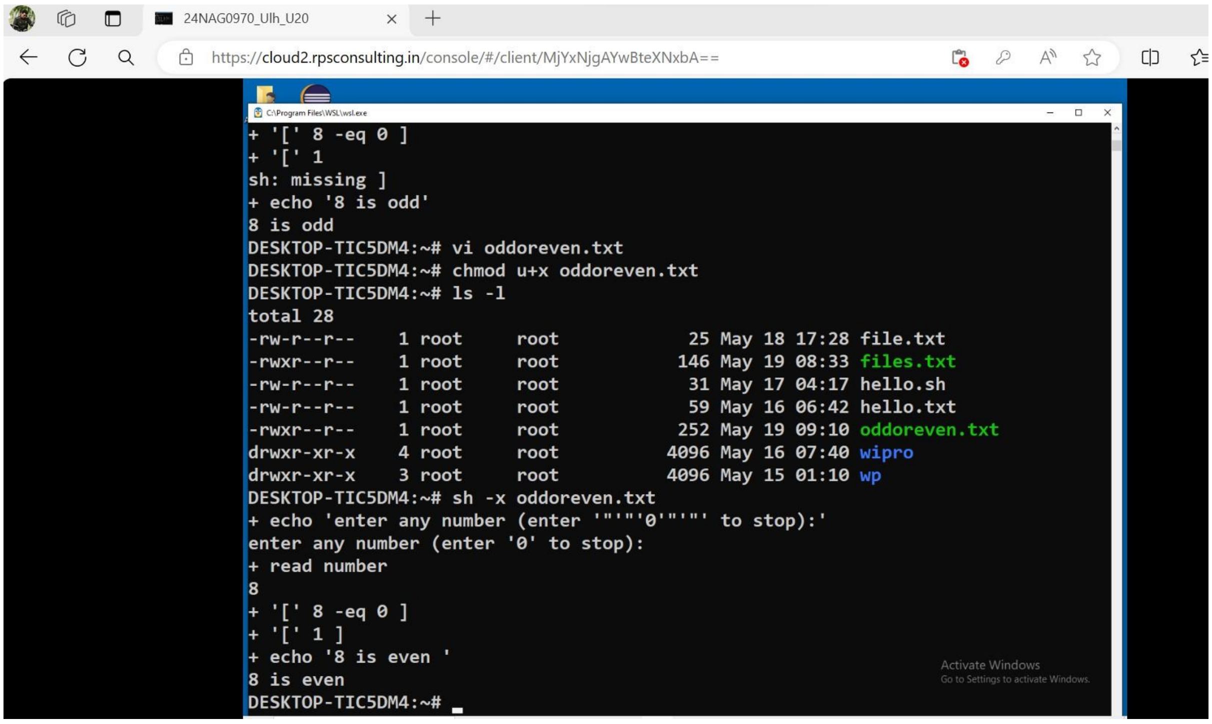
2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even

- Creating a file in vi editor named as oddoreven.txt
- Printing using echo “enter any number” and read the number variable
- After reading the input from the user taking an if statement of the validation of the value given by the user is 0 or greater than 0 ,if its 0 then the loop exit here by the command of break.
- Taking an another looping fuction of if else to check the user value id even or odd.
- The condition on the if is “if [$ ((number % 2 == 0))] then the condition is given value of 8 it is even and it will execute or odd number give executes


24NAG0970_Ulh_U20
https://cloud2.rpsconsulting.in/console/#/client/MjYxNjgAYwBteXNxbA==
C:\Program Files\WSL\wsl.exe

```
#!/bin/bash

echo "enter any number (enter '0' to stop):"
read number

if [ "$number" -eq 0 ]; then

 echo "user entered 0 exit the loop"
break
fi

if  [ $(( number % 2 == 0 )) ] ; then

   echo "$number is even "

else
    echo "$number is odd"
fi
~
~
~
~
~
```

- After the scripting will we execute using chmod u+x filename.txt and Ls -l is used check the file is stored in the directory.


24NAG0970_Ulh_U20
https://cloud2.rpsconsulting.in/console/#/client/MjYxNjgAYwBteXNxbA==
C:\Program Files\WSL\wsl.exe
Activate Windows
Go to Settings to activate Windows.

```
+ '[' 8 -eq 0 ]
+ '[' 1
sh: missing ]
+ echo '8 is odd'
8 is odd
DESKTOP-TIC5DM4:~# vi oddoreven.txt
DESKTOP-TIC5DM4:~# chmod u+x oddoreven.txt
DESKTOP-TIC5DM4:~# ls -l
total 28
-rw-r--r--    1 root     root            25 May 18 17:28 file.txt
-rwxr--r--    1 root     root           146 May 19 08:33 files.txt
-rw-r--r--    1 root     root            31 May 17 04:17 hello.sh
-rw-r--r--    1 root     root            59 May 16 06:42 hello.txt
-rwxr--r--    1 root     root           252 May 19 09:10 oddoreven.txt
drwxr-xr-x    4 root     root          4096 May 16 07:40 wipro
drwxr-xr-x    3 root     root          4096 May 15 01:10 wp
DESKTOP-TIC5DM4:~# sh -x oddoreven.txt
+ echo 'enter any number (enter '"'"'0'"'"' to stop):'
enter any number (enter '0' to stop):
+ read number
8
+ '[' 8 -eq 0 ]
+ '[' 1 ]
+ echo '8 is even '
8 is even
DESKTOP-TIC5DM4:~# _
```

- Sh -x filename used execute command of the filename and the above script is successful

3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

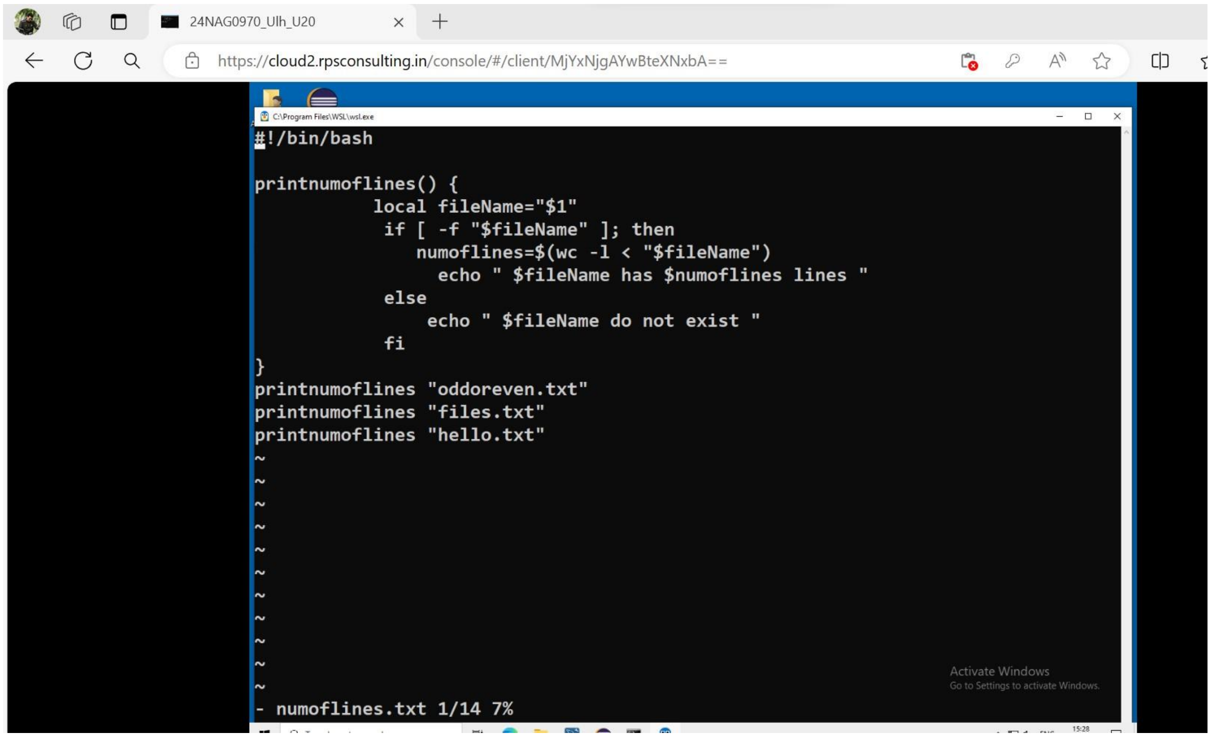
- we use #!/bin/bash for the script should run on the bash shell

- we create a function named as printnumoflines() and also we declare the local variable “fileName" and we assign the value of the first arugment to be passed to the function.

- If loop initializing on the step by checking the file by fileName exists on directory condition “ if [-f “$fileName”]; then.

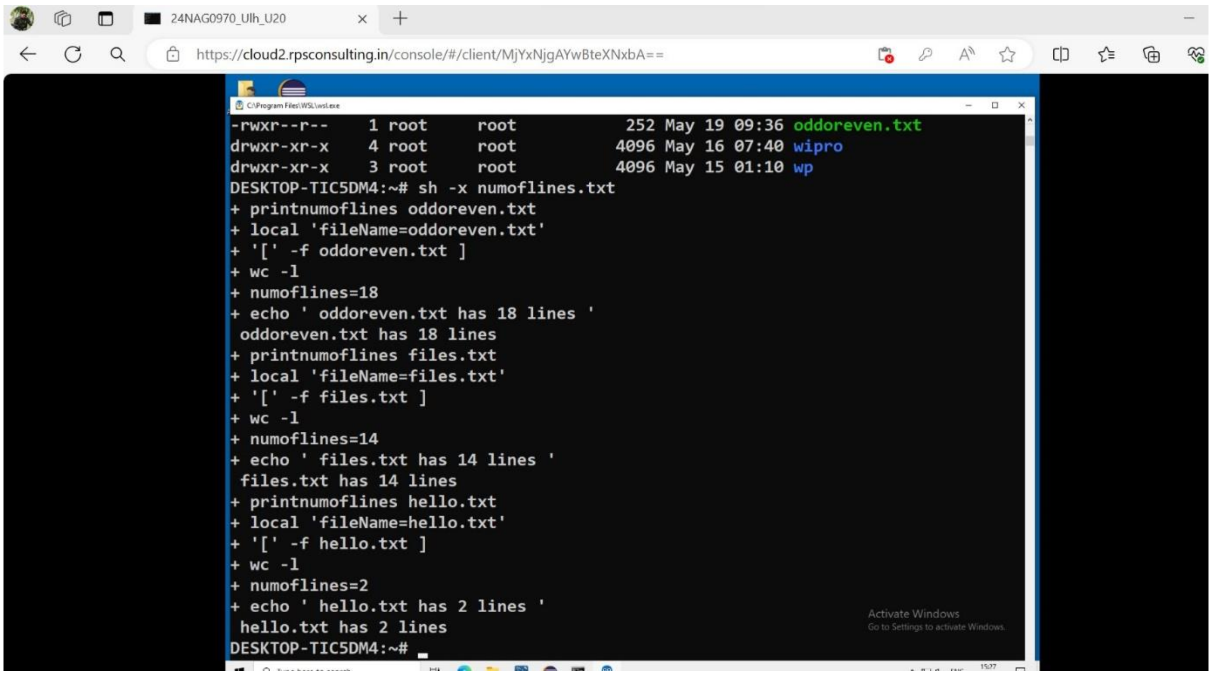
- Numoflines=$(wc -l < “$fileName”) here wc -l to count the number of lines in the file and assigns this value to numoflines.

- If file present then echo the lines or shows do not exist.



```
#!/bin/bash

printnumoflines() {
           local fileName="$1"
            if [ -f "$fileName" ]; then
               numoflines=$(wc -l < "$fileName")
                 echo " $fileName has $numoflines lines "
            else
                echo " $fileName do not exist "
            fi
}
printnumoflines "oddoreven.txt"
printnumoflines "files.txt"
printnumoflines "hello.txt"
~
~
~
~
~
~
~
~
~
~
~
- numoflines.txt 1/14 7%
```

- At final stage code executes and prints and it is attached in below image.

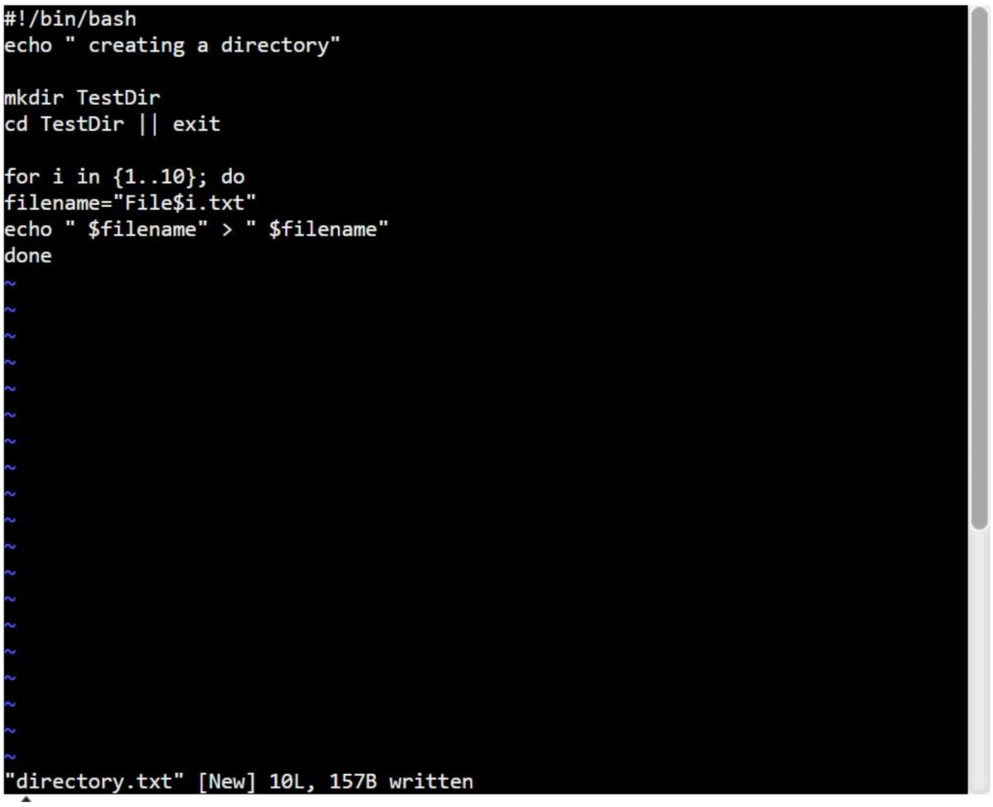

24NAG0970_Ulh_U20
https://cloud2.rpsconsulting.in/console/#/client/MjYxNjgAYwBteXNxbA==
C:\Program Files\WSL\wsl.exe
Activate Windows
Go to Settings to activate Windows.

```
-rwxr--r--    1 root     root           252 May 19 09:36 oddoreven.txt
drwxr-xr-x    4 root     root          4096 May 16 07:40 wipro
drwxr-xr-x    3 root     root          4096 May 15 01:10 wp
DESKTOP-TIC5DM4:~# sh -x numoflines.txt
+ printnumoflines oddoreven.txt
+ local 'fileName=oddoreven.txt'
+ '[' -f oddoreven.txt ]
+ wc -l
+ numoflines=18
+ echo ' oddoreven.txt has 18 lines '
 oddoreven.txt has 18 lines
+ printnumoflines files.txt
+ local 'fileName=files.txt'
+ '[' -f files.txt ]
+ wc -l
+ numoflines=14
+ echo ' files.txt has 14 lines '
 files.txt has 14 lines
+ printnumoflines hello.txt
+ local 'fileName=hello.txt'
+ '[' -f hello.txt ]
+ wc -l
+ numoflines=2
+ echo ' hello.txt has 2 lines '
 hello.txt has 2 lines
DESKTOP-TIC5DM4:~# _
```

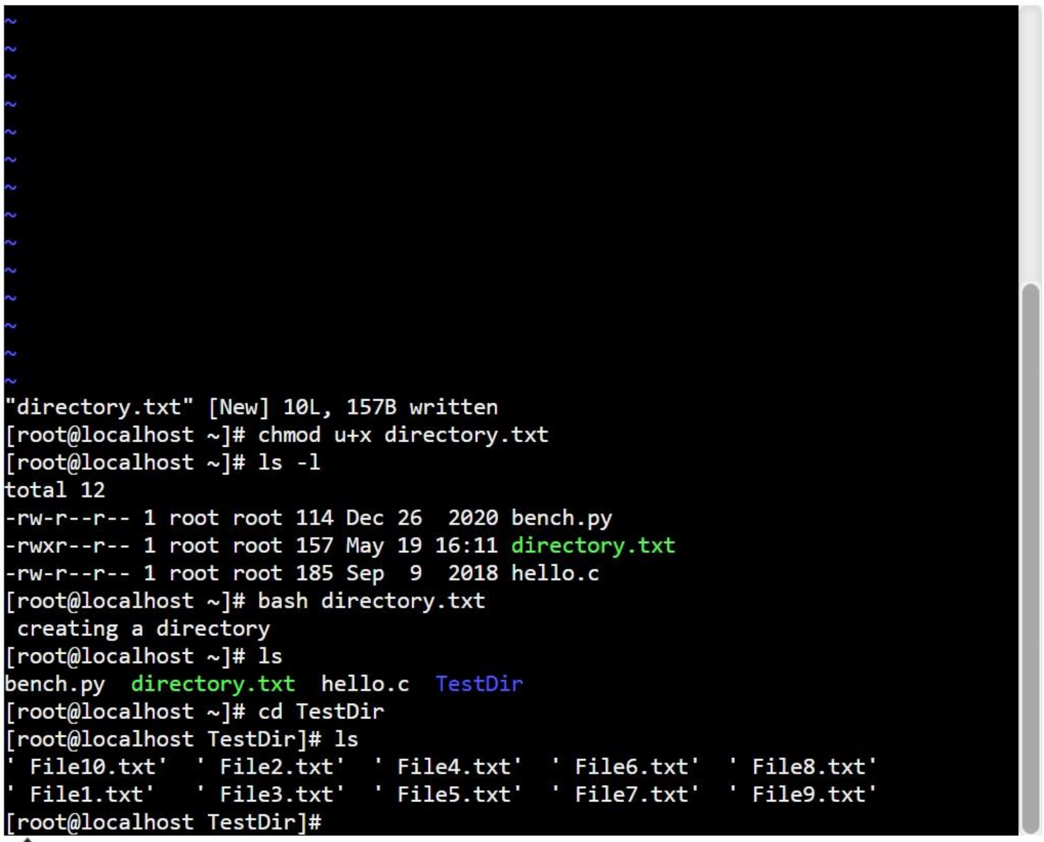
4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

- we use #!/bin/bash for the script should run on the bash shell
- Prints a message indicating that a directory is being created.
- Creates a directory named “TestDir”.
- After that Changes the current directory to “TestDir”. If the directory change fails,the script exits.
- We initialize the for loop Uses a loop to create ten files (“file1.txt to file10.txt) in the TestDir.
- Each file contains the filename and its content.
- After we run or execute the script using the chod and ls -l by checking the status .use bash directory
- We use ls for the content of directories and files
- In the next step we use cd TestDir for change the Directory and again we use the ls which the files1.txt to file10.txt is created or not

- Here the execution of this shell script



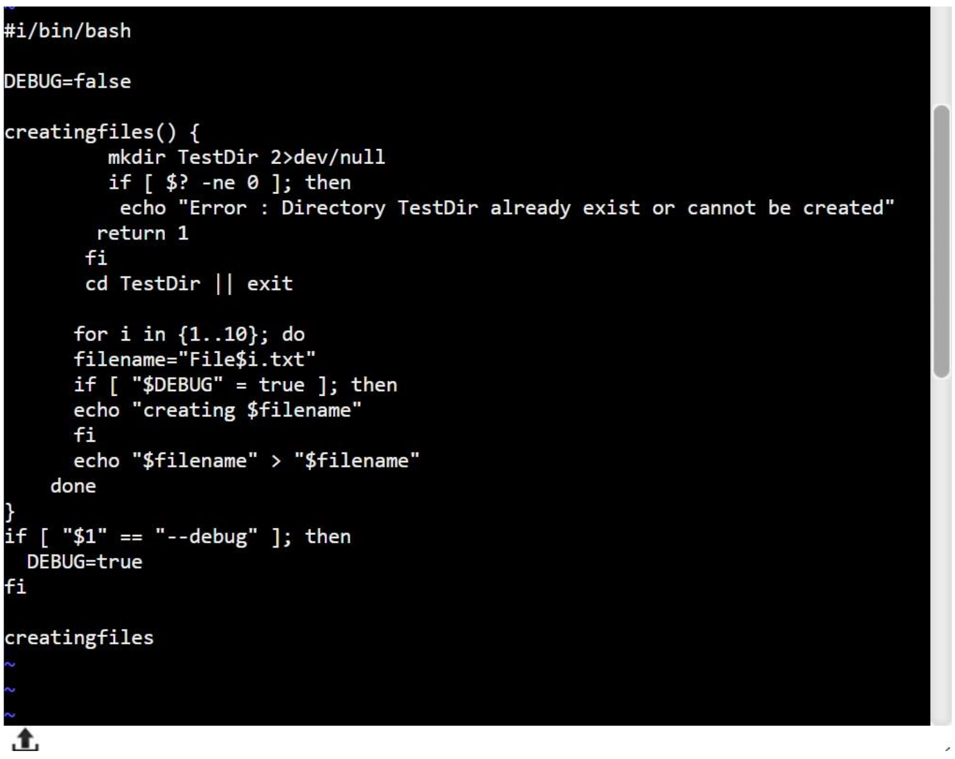
```
#!/bin/bash
echo " creating a directory"

mkdir TestDir
cd TestDir || exit

for i in {1..10}; do
filename="File$i.txt"
echo " $filename" > " $filename"
done
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"directory.txt" [New] 10L, 157B written
```



```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"directory.txt" [New] 10L, 157B written
[root@localhost ~]# chmod u+x directory.txt
[root@localhost ~]# ls -l
total 12
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rwxr--r-- 1 root root 157 May 19 16:11 directory.txt
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# bash directory.txt
 creating a directory
[root@localhost ~]# ls
bench.py  directory.txt  hello.c  TestDir
[root@localhost ~]# cd TestDir
[root@localhost TestDir]# ls
' File10.txt'  ' File2.txt'  ' File4.txt'  ' File6.txt'  ' File8.txt'
' File1.txt'   ' File3.txt'  ' File5.txt'  ' File7.txt'  ' File9.txt'
[root@localhost TestDir]#
```

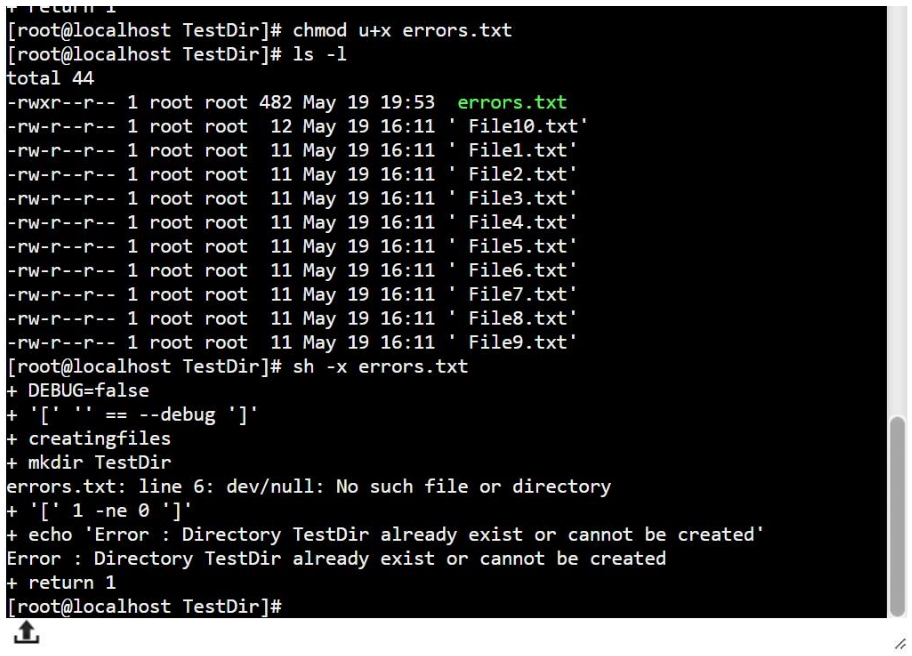
5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.
Add a debugging mode that prints additional information when enabled.

- we use #!/bin/bash for the script should run on the bash shell
- initializing a variable DEBUG with a value of “false”
- creating a directory named as “TestDir” and the “2>/dev/null” part redirects any error messages to dev/null silencing them.
- We check the exit status of “mkdir” command using “$?”,if the directory already exists or cannot be created.it prints an error message and returns 1.also changing the “TestDir” directory and “|| exit” part exits script line if cd command fails.
- In the for loop the iteration over 1 to 10 and it creates each filename and if debug indicate true then creation of file is success.
- Echo creates a file with the current filename and writes content.
- If loop checks the first argument passed to script “—debug” and calls the “creatingfiles” function to execute



```
#i/bin/bash

DEBUG=false

creatingfiles() {
         mkdir TestDir 2>dev/null
         if [ $? -ne 0 ]; then
          echo "Error : Directory TestDir already exist or cannot be created"
        return 1
       fi
       cd TestDir || exit

      for i in {1..10}; do
      filename="File$i.txt"
      if [ "$DEBUG" = true ]; then
      echo "creating $filename"
      fi
      echo "$filename" > "$filename"
    done
}
if [ "$1" == "--debug" ]; then
  DEBUG=true
fi

creatingfiles
~
~
~
```

- Here the execution of the script



```
[root@localhost TestDir]# chmod u+x errors.txt
[root@localhost TestDir]# ls -l
total 44
-rwxr--r-- 1 root root 482 May 19 19:53  errors.txt
-rw-r--r-- 1 root root  12 May 19 16:11 ' File10.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File1.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File2.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File3.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File4.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File5.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File6.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File7.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File8.txt'
-rw-r--r-- 1 root root  11 May 19 16:11 ' File9.txt'
[root@localhost TestDir]# sh -x errors.txt
+ DEBUG=false
+ '[' '' == --debug ']'
+ creatingfiles
+ mkdir TestDir
errors.txt: line 6: dev/null: No such file or directory
+ '[' 1 -ne 0 ']'
+ echo 'Error : Directory TestDir already exist or cannot be created'
Error : Directory TestDir already exist or cannot be created
+ return 1
[root@localhost TestDir]#
```

6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

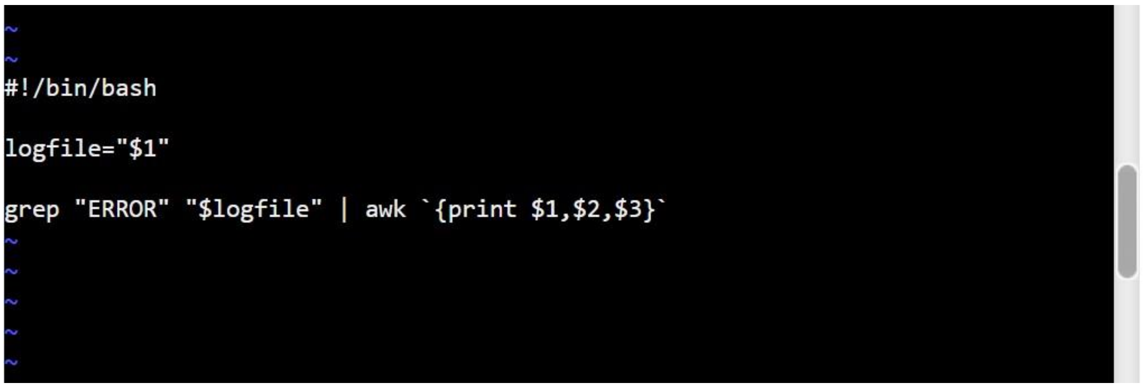
- we use #!/bin/bash for the script should run on the bash shell

- logfile assigns the first argument passed to the script (‘$1’) to variable to the logfile

- the script expects the log file name to be provided as a command-line argument when the script is run.

- Grep “ERROR” “$logfile” filters the contents of the log file, outputting only lines that contain the string "ERROR".

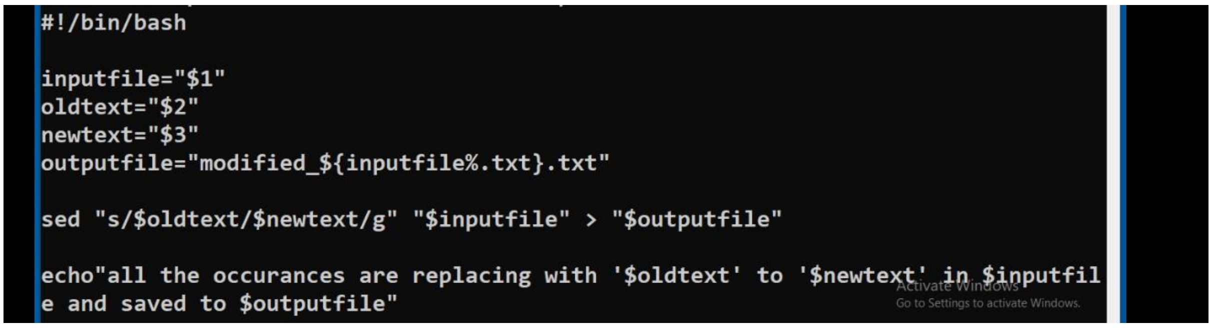
- The output of grep command is piped (“|”) to awk. Awk 1{ print $1,$2,$3} processes each line of the input, printing the first three fields. These corresponds to date,time,string “error”.



```
~
~
#!/bin/bash

logfile="$1"

grep "ERROR" "$logfile" | awk `{print $1,$2,$3}`
~
~
~
~
~
```

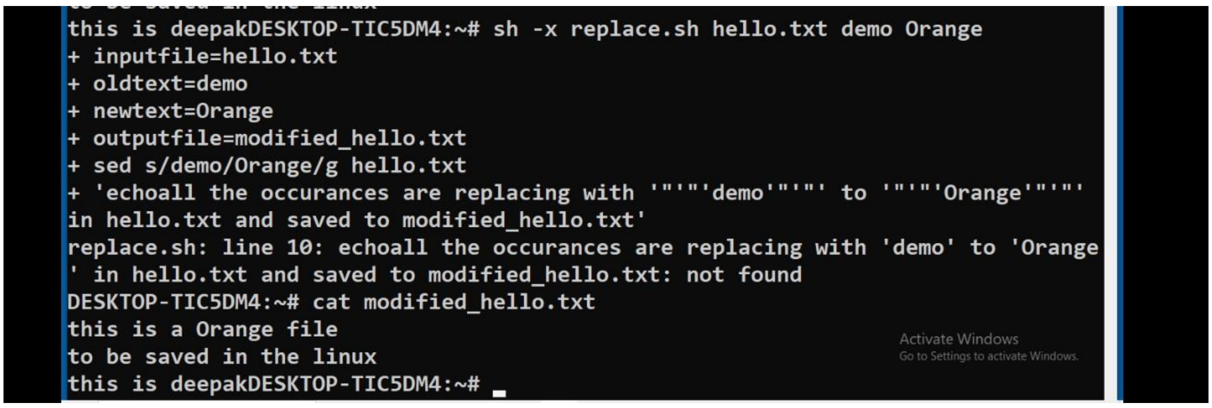
7: Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

- we use #!/bin/bash for the script should run on the bash shell
- “input file” is assigned the first argument provided to the script.
  “old text “ is assigned the second argument.
  “new text” is assigned the third argument.
- “outputfile” is constructed by prefixing the input file name (without the ‘.txt’ extension) with “modified_” and adding “.txt” extension.
- **`sed "s/$oldtext/$newtext/g"`** replaces all occurrences of with “oldtxt” to “newtxt” in inputfile.
- This prints a message indicating that the replacement was successful and specifying the input and output file names.
- We excute the replace.sh and sh -x replace.sh hello.txt demo orange after that we enter the cat modified_hello.txt and execute.



```
#!/bin/bash

inputfile="$1"
oldtext="$2"
newtext="$3"
outputfile="modified_${inputfile%.txt}.txt"

sed "s/$oldtext/$newtext/g" "$inputfile" > "$outputfile"

echo"all the occurances are replacing with '$oldtext' to '$newtext' in $inputfil
e and saved to $outputfile"
```

- Execution on the next image



```
this is deepakDESKTOP-TIC5DM4:~# sh -x replace.sh hello.txt demo Orange
+ inputfile=hello.txt
+ oldtext=demo
+ newtext=Orange
+ outputfile=modified_hello.txt
+ sed s/demo/Orange/g hello.txt
+ 'echoall the occurances are replacing with '"'"'demo'"'"' to '"'"'Orange'"'"'
in hello.txt and saved to modified_hello.txt'
replace.sh: line 10: echoall the occurances are replacing with 'demo' to 'Orange
' in hello.txt and saved to modified_hello.txt: not found
DESKTOP-TIC5DM4:~# cat modified_hello.txt
this is a Orange file
to be saved in the linux
this is deepakDESKTOP-TIC5DM4:~# _
```