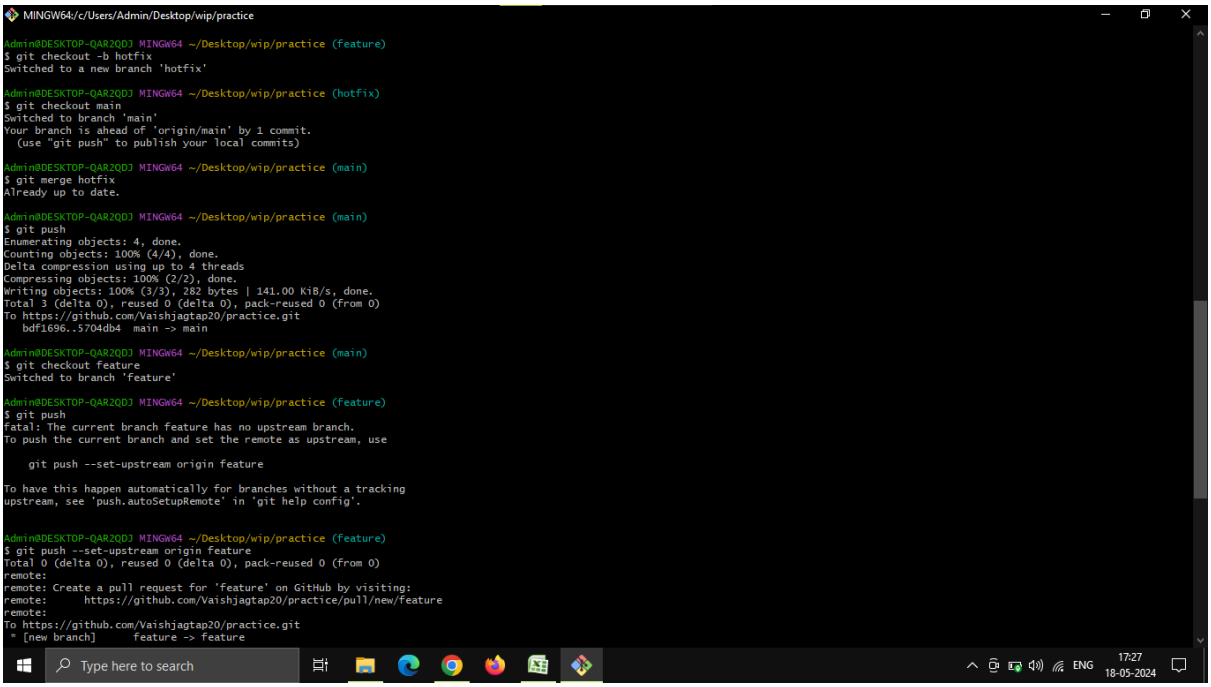
Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

```
MINGW64:/c/Users/Admin/Desktop/wip/practice
$ git init
Reinitialized existing Git repository in C:/Users/Admin/Desktop/wip/.git/

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ config --global user.name "Vaishjagtap20"
bash: config: command not found

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ config --global user.email "Vaishuj203@gmail.com"
bash: config: command not found

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ config --global user.email Vaishuj203@gmail.com
bash: config: command not found

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ git config --global user.name "Vaishjagtap20"

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ git config --global user.email Vaishuj203@gmail.com

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ cd practice
bash: cd: practice: No such file or directory

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ git clone https://github.com/Vaishjagtap20/practice.git
Cloning into 'practice'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip (master)
$ cd practice

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ touch myfile.txt

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git add .

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git commit -m "Initial commit"
[main 5704db4] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 myfile.txt

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$
```

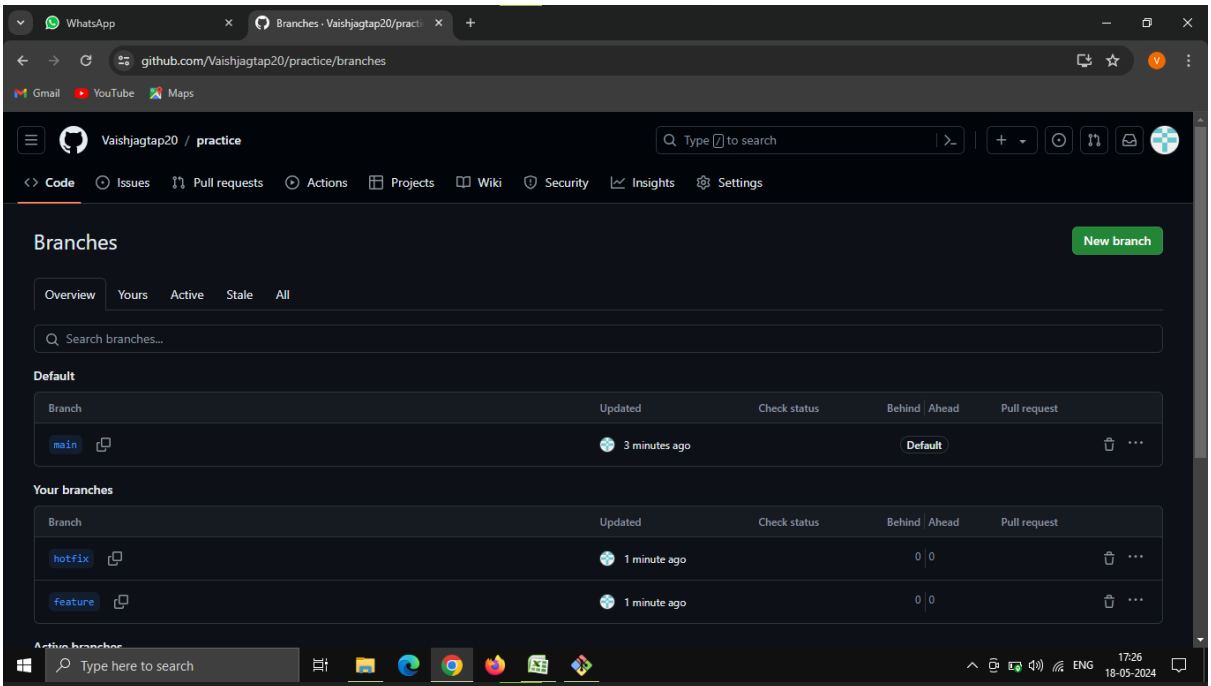
Assignment 2: Branch Creation and Switching Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

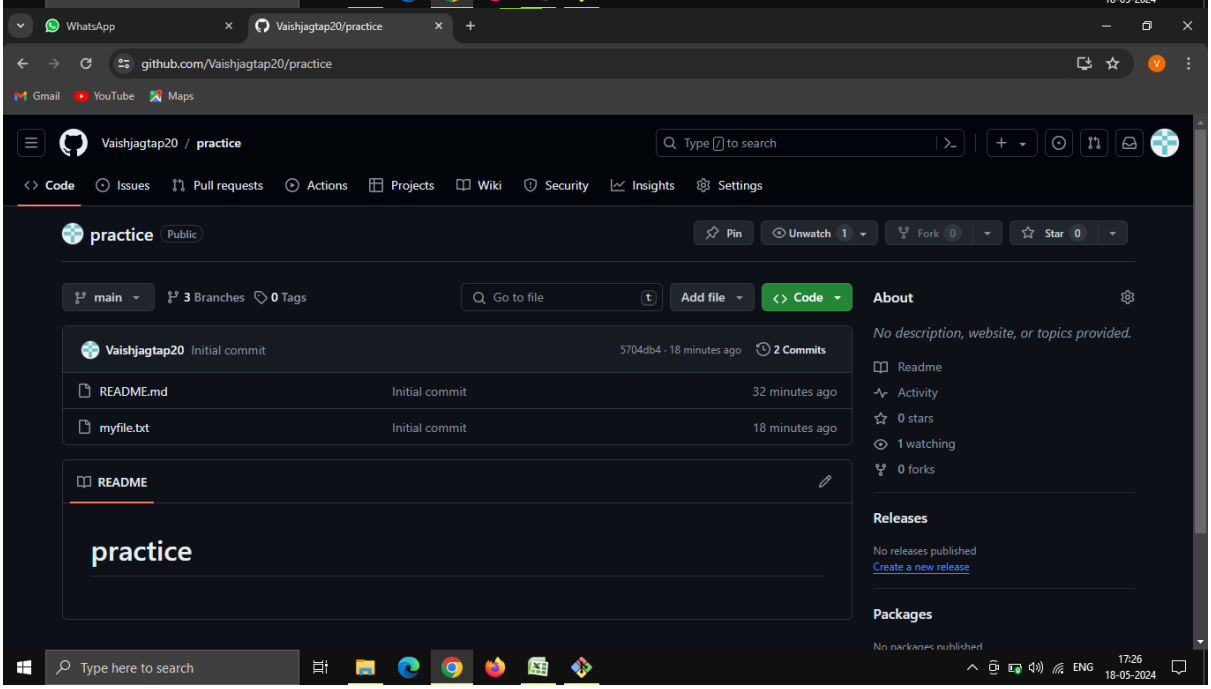
```
Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git checkout -b feature
Switched to a new branch 'feature'

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git status
On branch feature
nothing to commit, working tree clean

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git add .

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git commit -m "branch create"
On branch feature
nothing to commit, working tree clean

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$
```

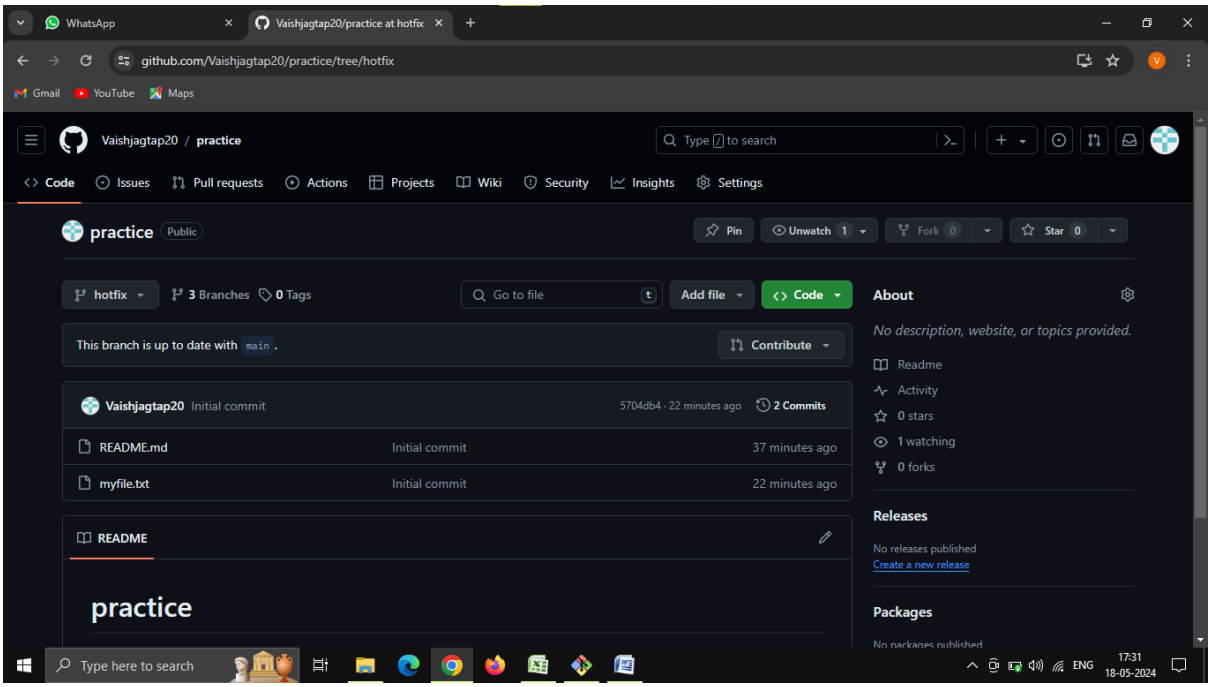
Assignment 3: Feature Branches and Hotfixes Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.



```
MINGW64:/c/Users/Admin/Desktop/wip/practice

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git checkout -b hotfix
Switched to a new branch 'hotfix'

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (hotfix)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git merge hotfix
Already up to date.

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 282 bytes | 141.00 Ki8/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vaishjagtap20/practice.git
   bdf1696..5704db4  main -> main

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (main)
$ git checkout feature
Switched to branch 'feature'

Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git push
fatal: The current branch feature has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.


Admin@DESKTOP-QAR2QDJ MINGW64 ~/Desktop/wip/practice (feature)
$ git push --set-upstream origin feature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:      https://github.com/Vaishjagtap20/practice/pull/new/feature
remote:
To https://github.com/Vaishjagtap20/practice.git
 * [new branch]      feature -> feature
```

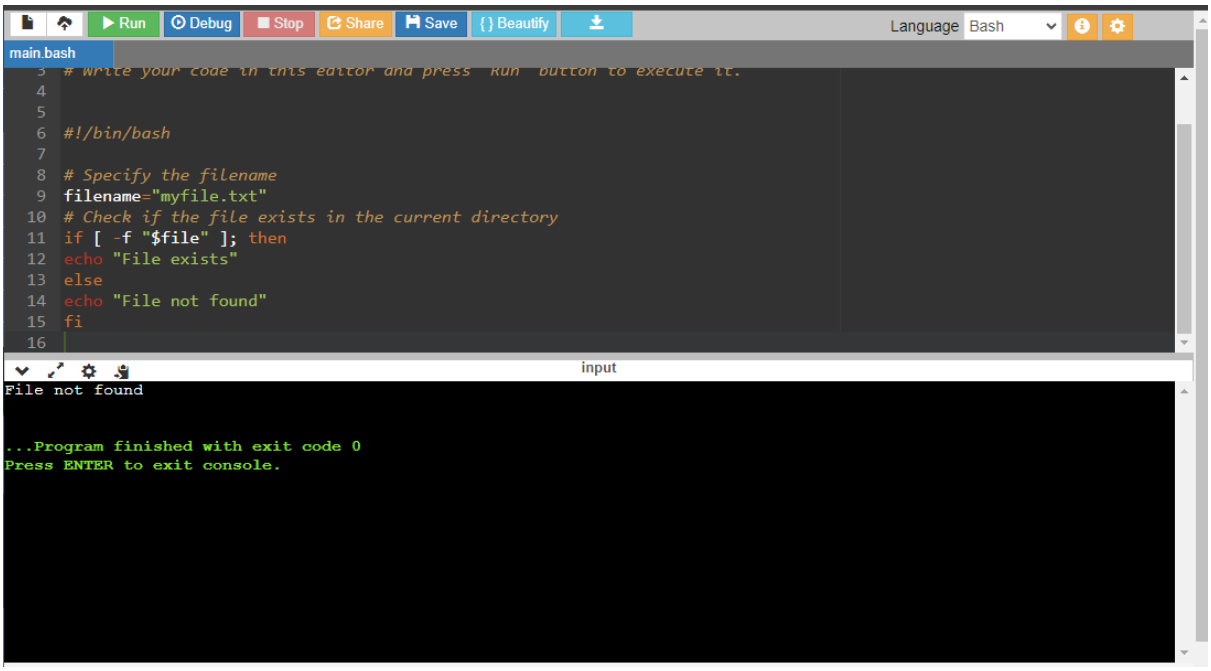

WhatsApp
Branches · Vaishjagtap20/practi
github.com/Vaishjagtap20/practice/branches
Gmail
YouTube
Maps
Vaishjagtap20 / practice
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Branches
New branch
Overview
Yours
Active
Stale
All
Search branches...
Default
Branch
Updated
Check status
Behind | Ahead
Pull request
main
3 minutes ago
Default
Your branches
hotfix
1 minute ago
0 | 0
feature
1 minute ago
0 | 0
Type here to search
ENG
17:26
18-05-2024


WhatsApp
Vaishjagtap20/practice
github.com/Vaishjagtap20/practice
Gmail
YouTube
Maps
Vaishjagtap20 / practice
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
practice
Public
Pin
Unwatch 1
Fork 0
Star 0
main
3 Branches
0 Tags
Go to file
t
Add file
Code
Vaishjagtap20 Initial commit
5704db4 · 18 minutes ago
2 Commits
README.md
Initial commit
32 minutes ago
myfile.txt
Initial commit
18 minutes ago
README
practice
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
Type here to search
ENG
17:26
18-05-2024



Assignment 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"

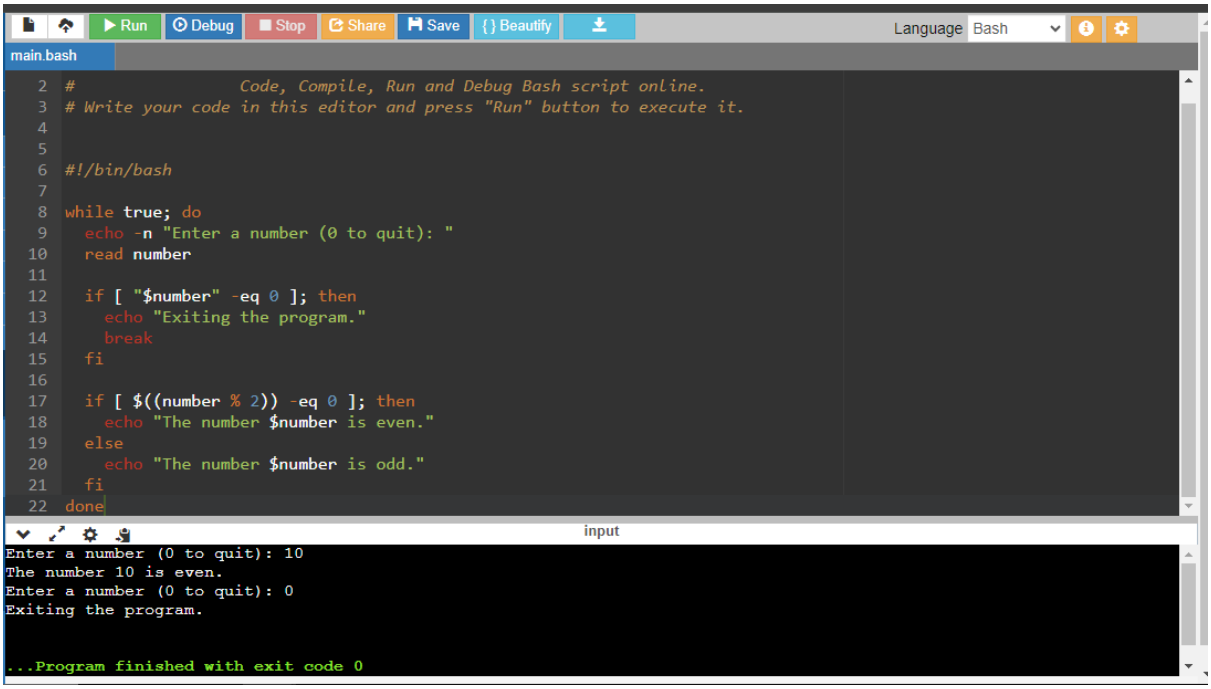
```
#!/bin/bash

if [ -f "myfile.txt" ]; then

    echo "File exists"

else

    echo "File not found"

fi
```



Assignment 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.

```
#!/bin/bash


while true; do
   read -p "Enter a number (0 to quit): " number
   if [ "$number" -eq 0 ]; then
      break
   elif [ $((number % 2)) -eq 0 ]; then
      echo "The number $number is even"
   else
      echo "The number $number is odd"
   fi
```

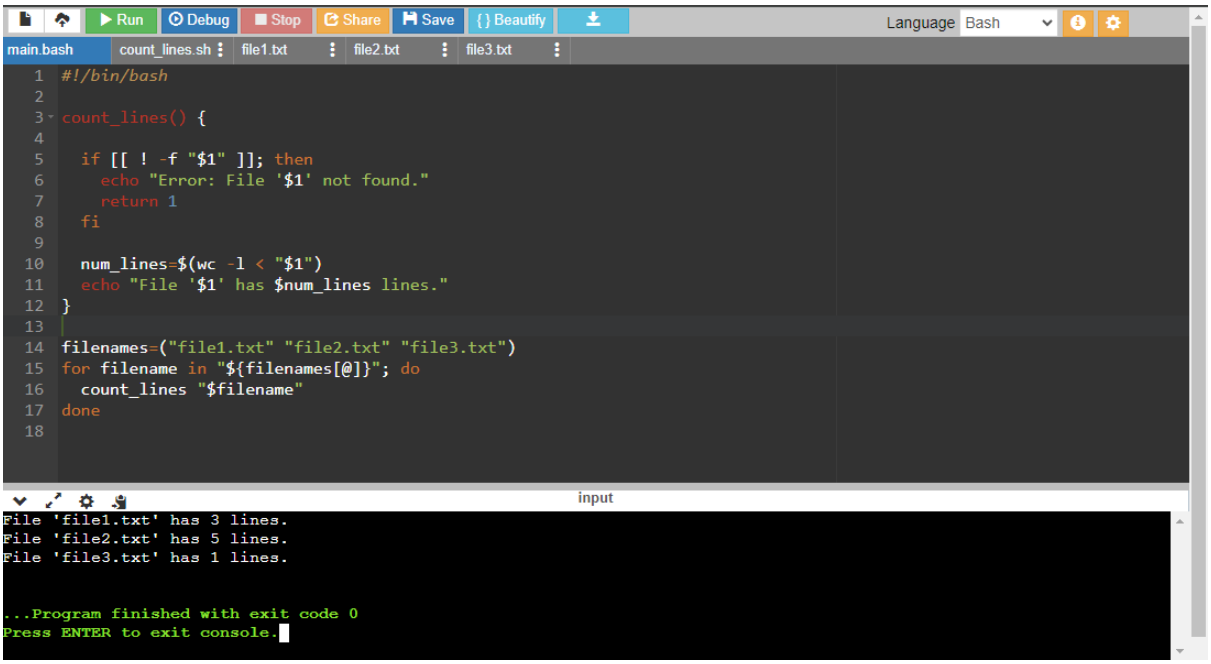
done



Assignment 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.

```
#!/bin/bash

count_lines() {

  if [[ ! -f "$1" ]]; then

    echo "Error: File '$1' not found."

    return 1

  fi

  num_lines=$(wc -l < "$1")

  echo "File '$1' has $num_lines lines."
```

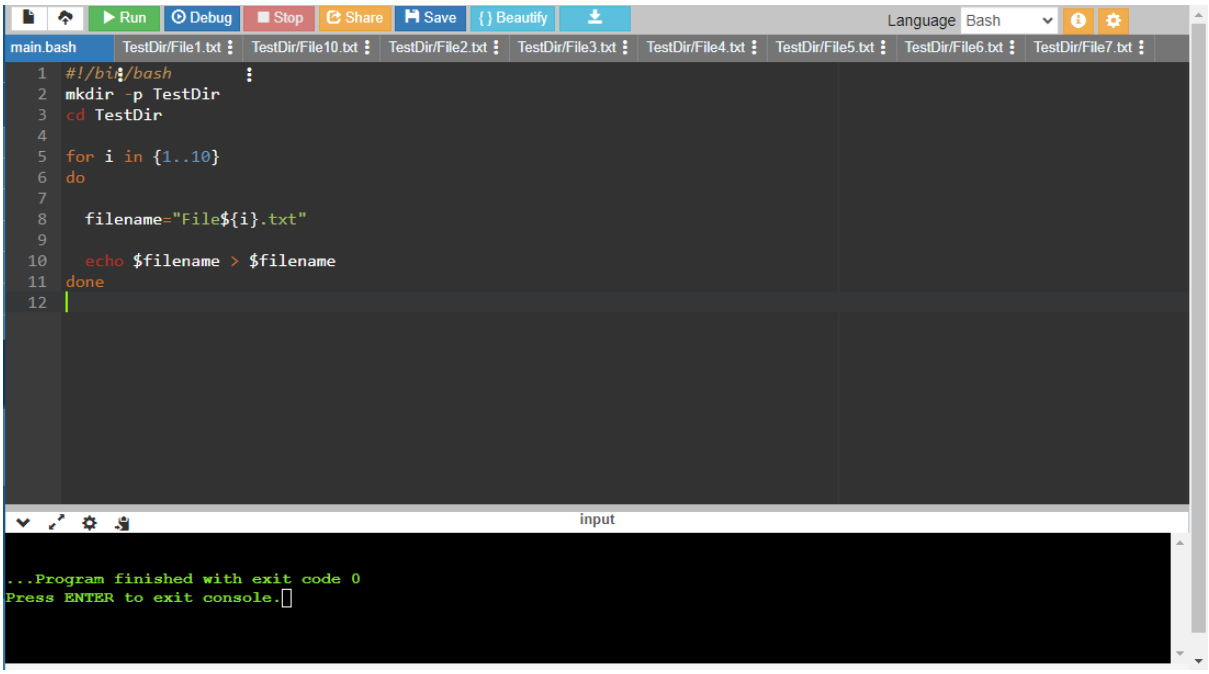
```
}

filenames=("file1.txt" "file2.txt" "file3.txt")

for filename in "${filenames[@]}"; do

  count_lines "$filename"

done
```



Assignment 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").

```
#!/bin/bash

mkdir -p TestDir

cd TestDir
```

```
for i in {1..10}
do
  filename="File${i}.txt"
  echo $filename > $filename
done
```

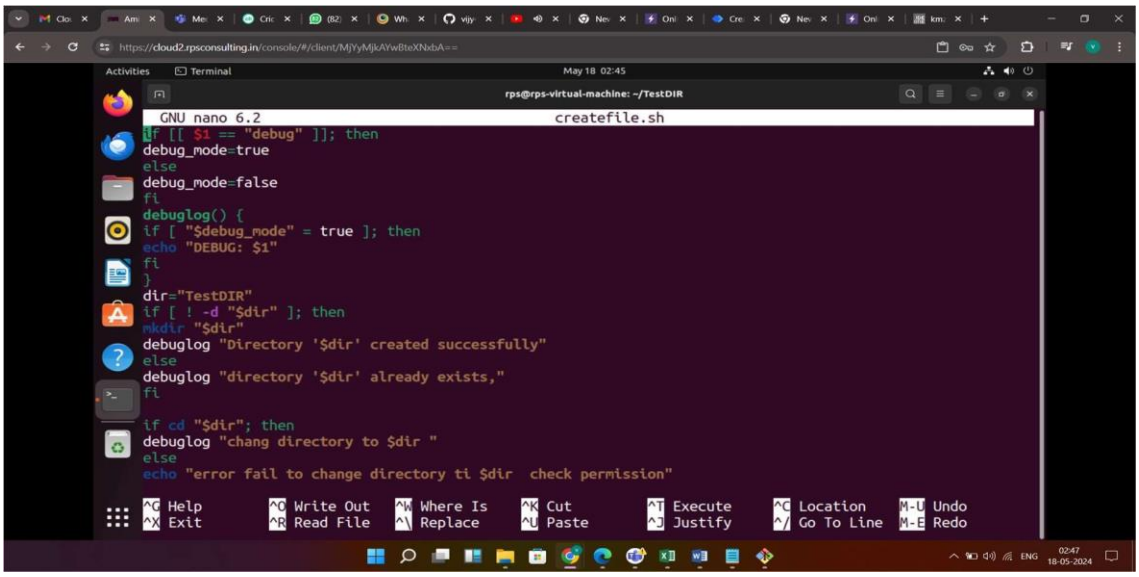


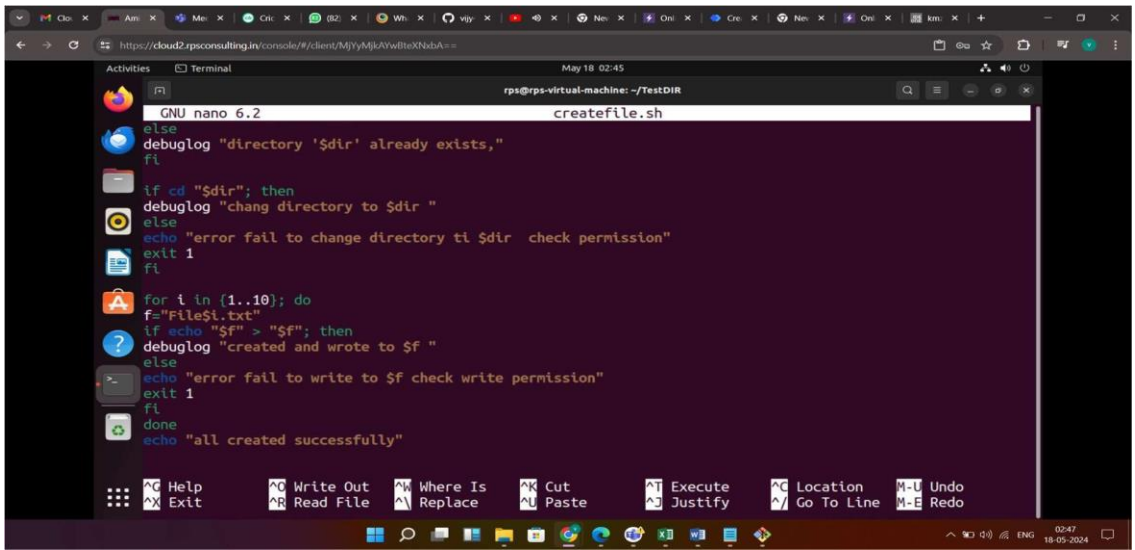
Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files. Add a debugging mode that prints additional information when enabled

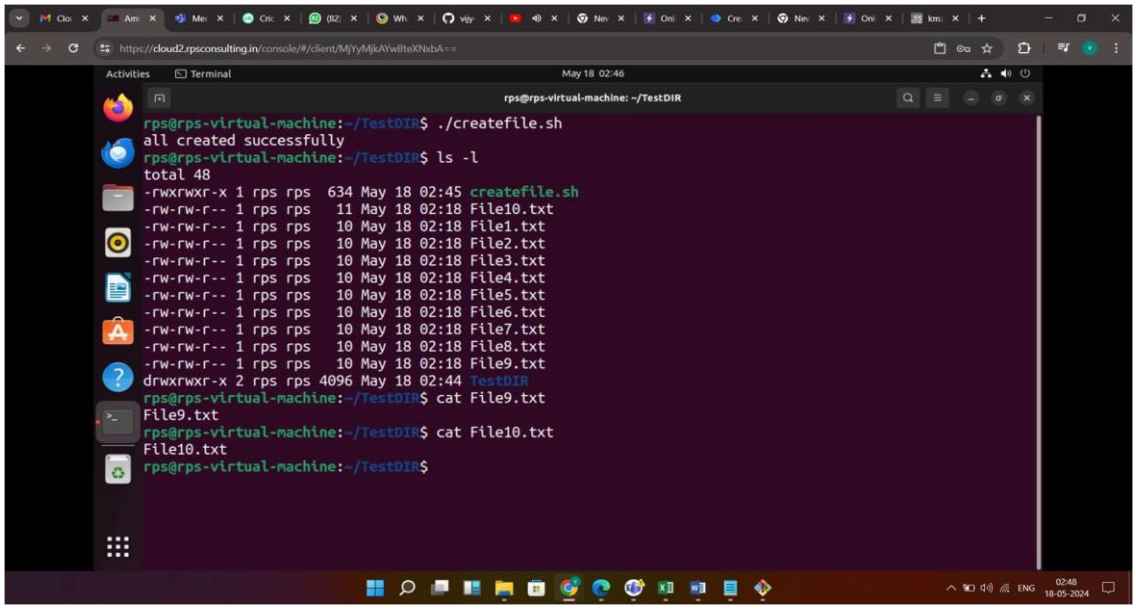
```
#!/bin/bash
if [[ $1 == "debug" ]]; then
   debug_mode=true
else
```

```
    debug_mode=false
fi
debuglog()
{
    if [ "$debug_mode" = true ]; then
        echo "DEBUG: $1"
    fi
}
dir ="TestDir"
if [ ! -d "$dir " ]; then
    mkdir "$dir "
    debuglog "Directory '$dir created successfully."
else
    debuglog "Directory '$dir  already exists, skipping creation."
fi
if cd "$dir "; then
    debuglog "Changed directory to $dir."
else
    echo "Error: Failed to change directory to $dir . Check
permissions."
    exit 1
fi
for i in {1..10}; do
    f="File$i.txt"
    if echo "$f" > "$f"; then
```

```
    debuglog "Created and wrote to $f."
else
    echo "Error: Failed to write to
    $f." exit 1
```



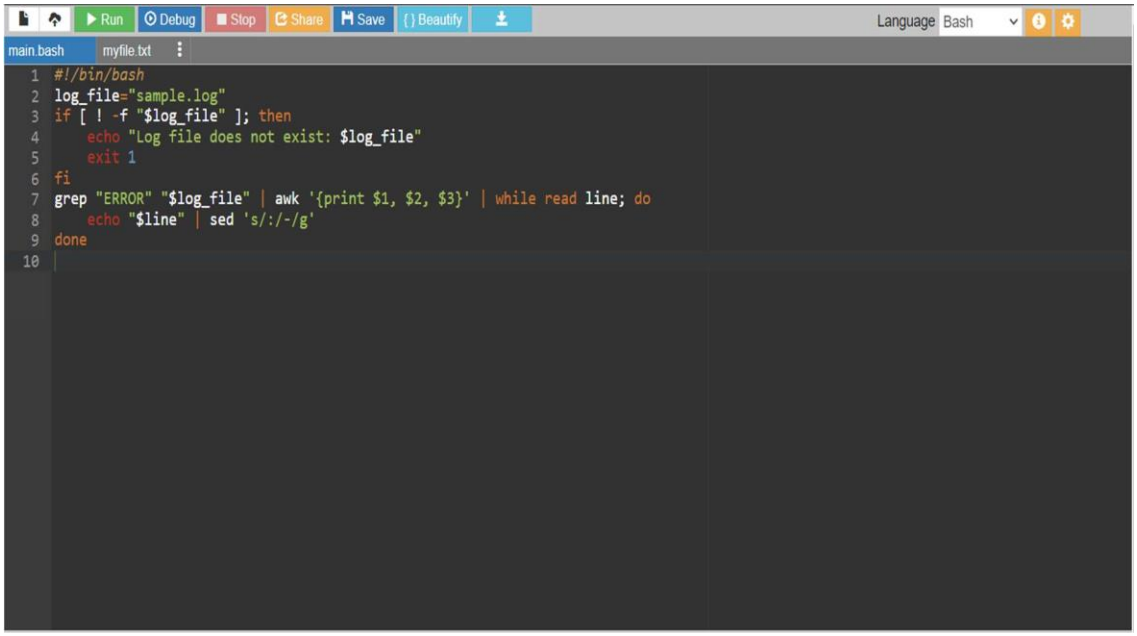




Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line Data Processing with sed

```
#!/bin/bash
log_file="sample.log"
if [ ! -f "$log_file" ]; then
echo "Log file does not exist: $log_file"
exit 1
fi
grep "ERROR" "$log_file" | awk '{print $1, $2, $3}' | while read line; do
```

```
    echo "$line" | sed 's/:/-/g'
done
```



Assignment 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

```
#!/bin/bash
if [ $# -ne 3 ]; then
  echo "Usage: $0 input_file old_text
  new_text" exit 1
fi
input_file="
$1"
old_text="$
```

2"

new_text="

$3"

output_file="${input_file%.txt}_modifie

d.txt" if [! -f "$input_file"]; then

echo "Error: File does not exist -

$input_file" exit 1

fi

sed "s/$old_text/$new_text/g" "$input_file" >

"$output_file" echo "Operation completed. Modified file is

$output_file"