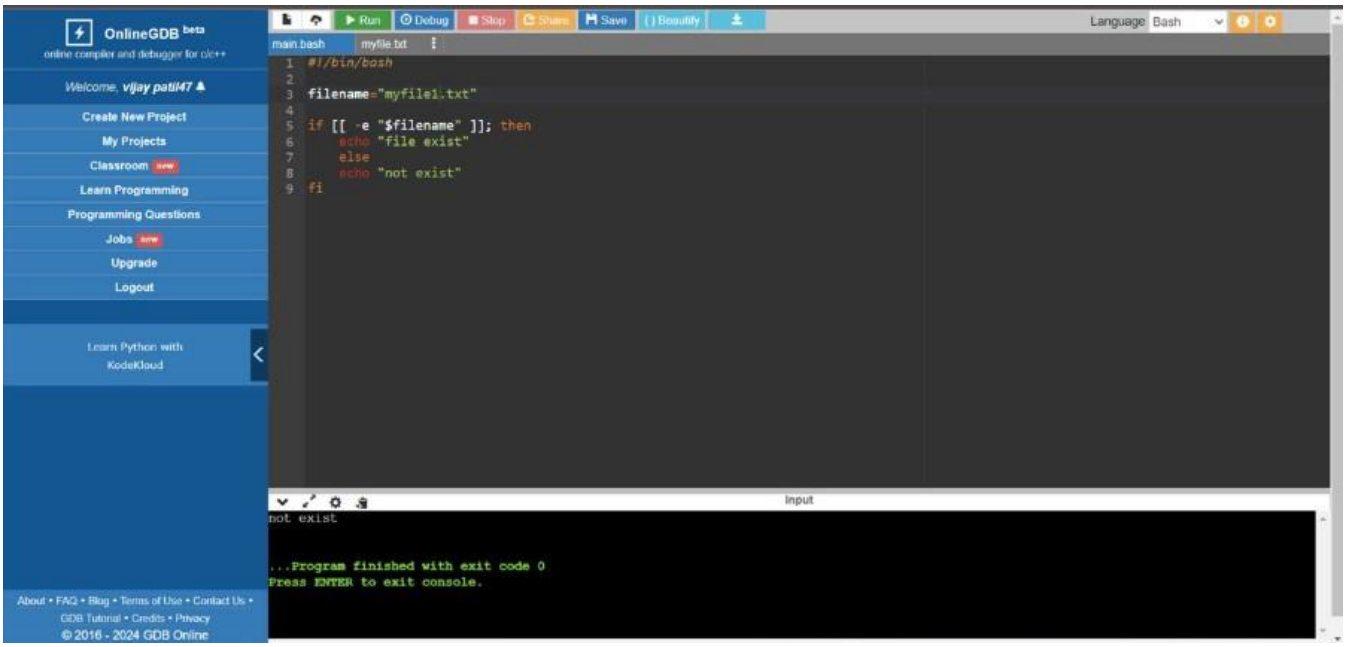
Assignment 1

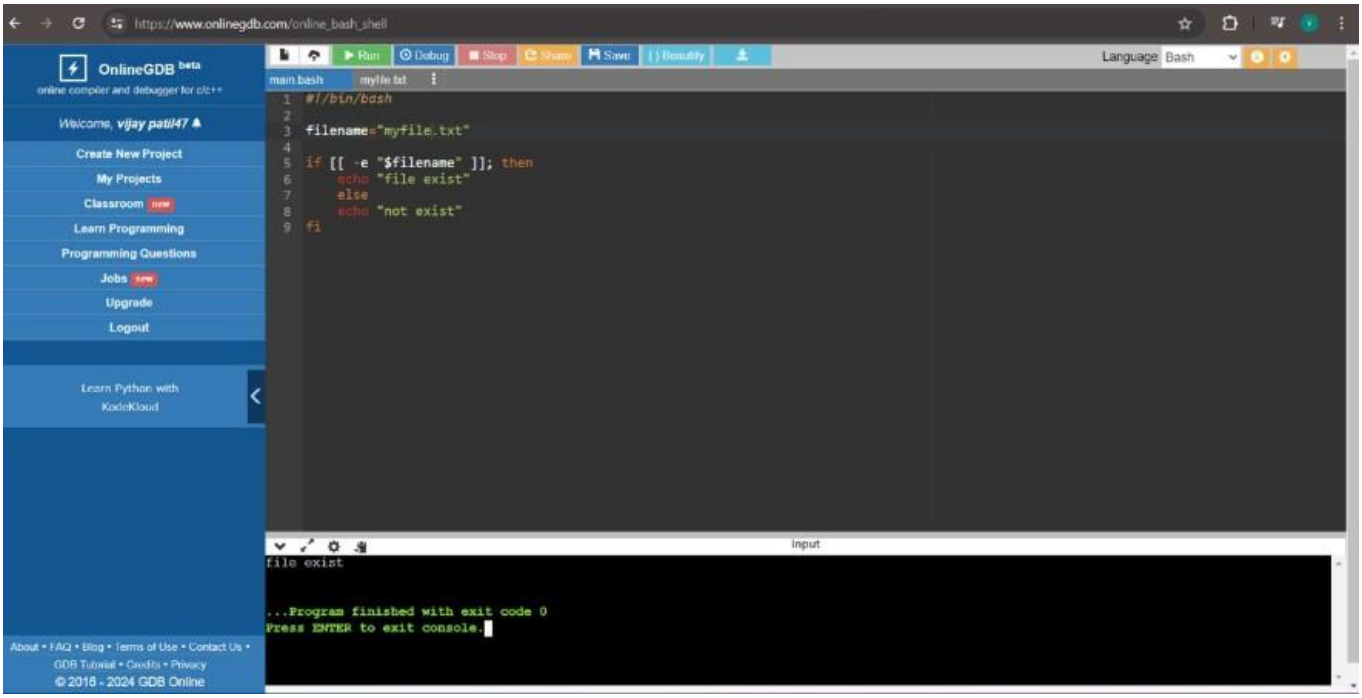
1] Ensure the script checks if a specific file (e.g.., myfile.txt) exists in the current directory. If exists, print “file exists”, otherwise print “File not found”.

Input and Output Snapshot:-



OUTPUT:

File not exist



OUTPUT:

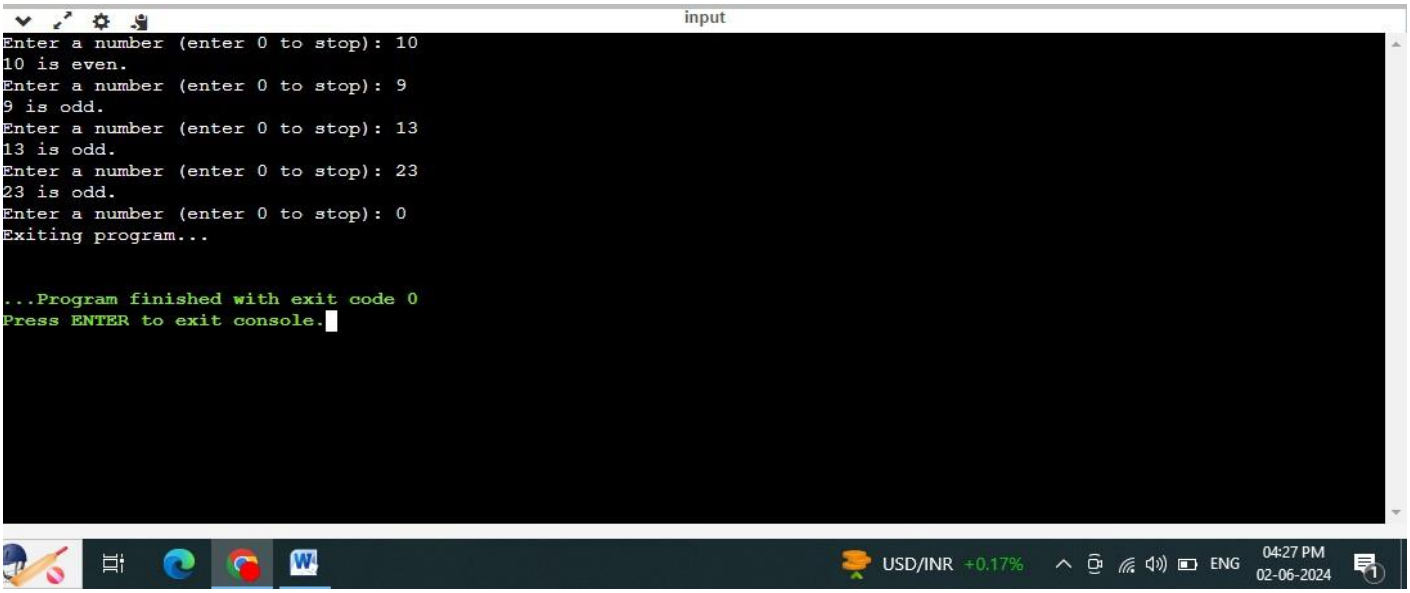
File exist

2] Write a script that reads number 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 (enter 0 to quit): " num

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

 echo "Exiting program..."

break

fi

 if (( num % 2 == 0 )); then
```

```
echo "$num is even."
 else echo "$num is odd."
fi
done
```

OUTPUT:



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

Nano count_line.sh

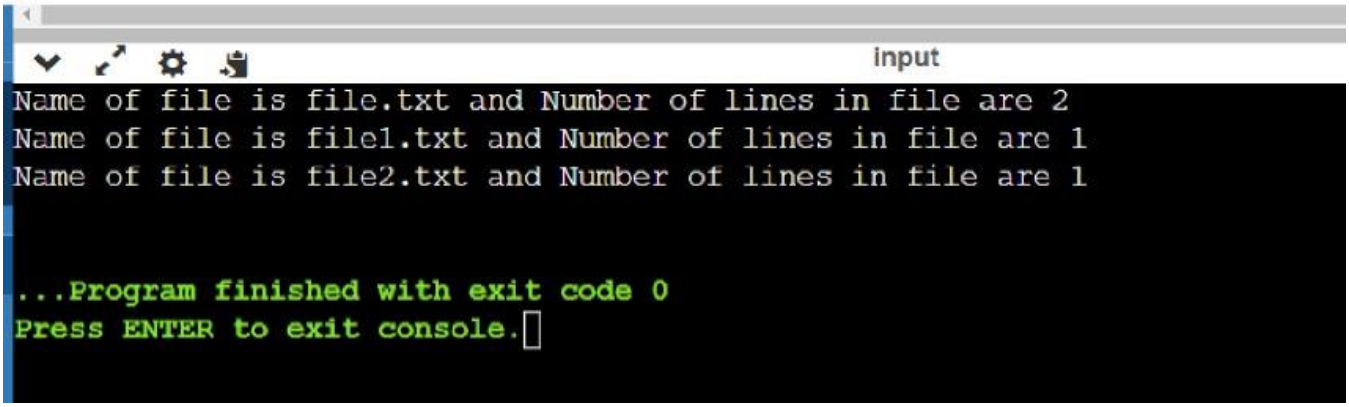
This command is use for the creating script file.

Script:

```
#!/bin/bash
```

```
count_lines() {
    local filename=$1
if [-f “$filename”]; then
    local line_count= $(wc -| <”$filename”)
   echo “Name of file is ”$filename and Number of lines in file are $lines”
else
   echo “Error: The file ‘$filename’ does not exist.”
fi
}
count_lines “file1.txt”
count_lines “file2.txt”
count_lines “file3.txt”
```

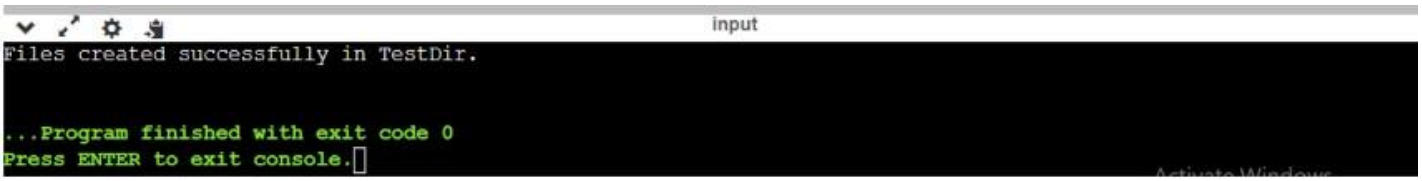
OUTPUT:



4] Write a script that creates a directory named TestDir and inside it, create ten files named File1.txt, File2.txt,…. File 10.txt. Each file should contain irs filename as its content(e.g., File1.txt contains “File1.txt”).

Script:

```
For i in (1…10); do

   F= “File$i.txt”

  Echo $f> “$f”

done

echo “file created.”
```




```
dir_name= “TestDir”

mkdir –p “$dir_name”


for i in {1…10}; do

  filename= “File$i.txt”

  filepath= “$dir_name/$filename”

echo “$filename” > “$filepath”

done

echo “Directory ‘$dir_name’ and files created successfully!”
```

5] Modify the script to handle errors, such as the directory already existing or lacking permission to create files.

```
#!/bin/bash


handle_error() {

   echo "Error: $1"

   exit 1

}

if [ -d "TestDir" ]; then

   handle_error "Directory 'TestDir' already exists."

fi

mkdir TestDir || handle_error "Failed to create directory 'TestDir'."

cd TestDir || handle_error "Failed to change into directory 'TestDir'."


for ((i=1; i<=10; i++)); do

   filename="File${i}.txt"

   echo "$filename" > "$filename" || handle_error "Failed to create file
'$filename'."

done


# Display message

echo "Files created successfully in TestDir."
```

```
directory and files created(or already exists).

...Program finished with exit code 0
Press ENTER to exit console.
```

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.

Run Debug Stop Share Save { } Beautify Language Bash

main.bash myfile.txt

```bash
#!/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
```

Script:

```
#!/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
```

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}_modified.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"
```

main.bash myfile.txt

```bash
#!/bin/bash

# Check for proper usage
if [ $# -ne 3 ]; then
    echo "Usage: $0 input_file old_text new_text"
    exit 1
fi

# Assign script arguments to variables
input_file="$1"
old_text="$2"
new_text="$3"
output_file="${input_file%.txt}_modified.txt"  # Appends '_modified' to the original filename

# Check if the input file exists
if [ ! -f "$input_file" ]; then
    echo "Error: File does not exist - $input_file"
    exit 1
fi

# Use sed to replace all occurrences of old_text with new_text
sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"

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

28°C