Assignment no 5

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".

Solution:-

```bash
#!/bin/bash
touch file2.txt
filename="file2.txt"

if [ -e "$filename" ];
then
    echo "File Exists"
else
    echo "Does not Exists"
fi
```

Output:-

```
File Exists


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

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.

Solution:-

Run Debug Stop Share Save { } Beautify

main.bash file2.txt

```bash
#!/bin/bash

while true;
do

 echo "Enter numbers "
 read num

 if [ "$num" -eq 0 ]
 then
    echo "zero is not allowed"
    exit 1
 fi

 if [ $((num % 2)) == 0 ]
 then
    echo "$num is even"
 else
    echo "$num is odd"
 fi
done
```

Output:-

```
Enter numbers
52
52 is even
Enter numbers
33
33 is odd
Enter numbers
0
zero is not allowed


...Program finished with exit code 1
Press ENTER to exit console.
```

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.

Solution:-

Run Debug Stop Share Save { } Beautify Language Bash

main.bash file2.txt file.txt file1.txt

```bash
#!/bin/bash
#touch file.txt file1.txt file2.txt
line_count(){
filename=$1
lines=$(wc -l < "$filename")
echo "Name of file is $filename and Number of lines in file are $lines"
}
line_count file.txt
line_count file1.txt
line_count file2.txt
```

Output:-

```
input
Name of file is file.txt and Number of lines in file are 2
Name of file is file1.txt and Number of lines in file are 1
Name of file is file2.txt and Number of lines in file are 1


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

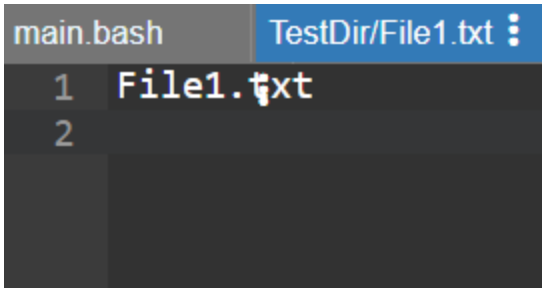
4] Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

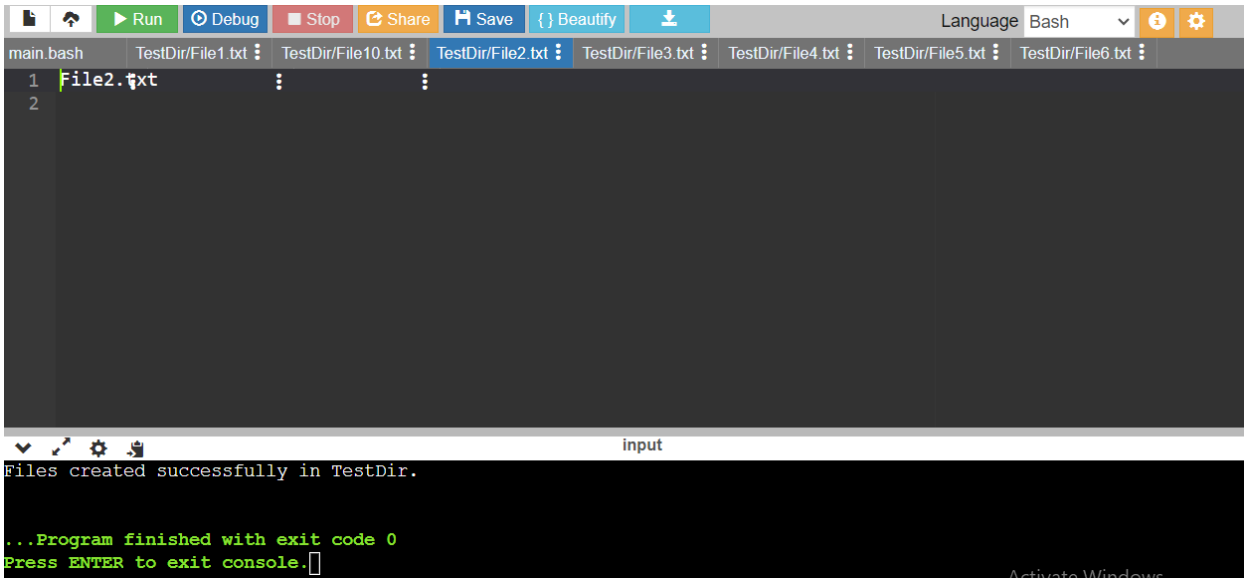
Solution:-

main.bash | TestDir/File1.txt | TestDir/File10.txt | TestDir/File2.txt | TestD

```bash

#!/bin/bash
mkdir TestDir
 cd TestDir || exit

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

    touch "File$i.txt"

    echo "File$i.txt" > "File$i.txt"
done
echo "Files created successfully in TestDir."
```

Output:-





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.

Solution:-

Execute | Beautify | Share Source Code Help

```bash
#!/bin/bash


create_directory() {
  local directory="$1"
  local debug_mode="${2:-false}"

  if [[ $debug_mode == true ]]; then
    echo "Checking if directory '$directory' exists..."
  fi

  if [ ! -d "$directory" ]; then
    mkdir -p "$directory"
    if [[ $? -eq 0 ]]; then
      if [[ $debug_mode == true ]]; then
        echo "Directory '$directory' created successfully."
      fi
    else
      echo "Error creating directory '$directory': Exit code ($?)"
    fi
  else
    if [[ $debug_mode == true ]]; then
      echo "Directory '$directory' already exists."
    fi
  fi
}


debug_mode=true
directory_to_create="my_new_directory"

create_directory "$directory_to_create" "$debug_mode"
```

```bash
}


debug_mode=true
directory_to_create="my_new_directory"

create_directory "$directory_to_create" "$debug_mode"

if [[ $? -eq 0 ]]; then
  echo "Directory '$directory_to_create' created!"
fi

```

Output:-

```
Terminal
Checking if directory 'my_new_directory' exists...
Directory 'my_new_directory' created successfully.
Directory 'my_new_directory' created!
```

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

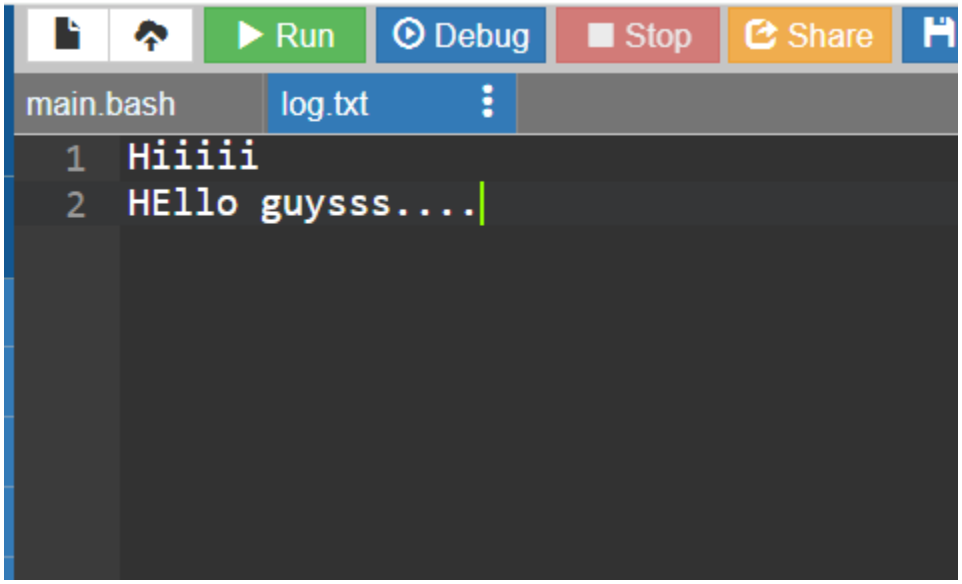
Solution:-

Run Debug Stop Share Save { } Beautify Lang

main.bash log.txt

```bash
#!/bin/bash
touch log.txt
log_file=log.txt
error_lines=$(grep "ERROR" "$log_file")


for line in $error_lines; do

  date_time=$(echo "$line" | awk '{print $1,$2}')
  error_message=$(echo "$line" | awk '{print $3}')


  echo "Date/Time: $date_time - Error: $error_message"
done
```

Log.txt file-



Output:-

Run Debug Stop Share Save { } Beautify

main.bash log.txt

```bash
#!/bin/bash
touch log.txt
log_file=log.txt
error_lines=$(grep "ERROR" "$log_file")


for line in $error_lines; do

  date_time=$(echo "$line" | awk '{print $1,$2}')
  error_message=$(echo "$line" | awk '{print $3}')


  echo "Date/Time: $date_time - Error: $error_message"
done
```

input

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

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.

Solution:-

tutorialspoint | Online Bash Compiler

Execute | Beautify | Share | Source Code | Help

```bash
if [ $# -ne 2 ];
then
    echo "Usage : $0 <input_file> <new_text>"
    exit 1
fi

input_file="$1"
new_text="$2"

fi [ ! -f "$input_file" ];
then
    echo "Error: File '$input_file' does not exist."
fi

output_file="${input_file%.txt}.replaced.txt"

sed -i "s/old_text/$new_text/g" "$input_file"

if [ $? -eq 0 ]
then
    echo "Successfully replaced 'old_text' with '$new_text' in '$input_file'."
    echo "Output saved to: $output_file"
else
    echo "Error: Failed to replace text in '$input_file'."
fi
```

Output:-

Terminal

```
Usage : main.sh <input_file> <new_text>
```