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
filename="myfile.txt"
if [ -e "$filename" ]; then
   echo "File exists"
else
   echo "File not found"
fi
```

Save this script in a file,

for example, `check_file.sh`, and make it executable using the command

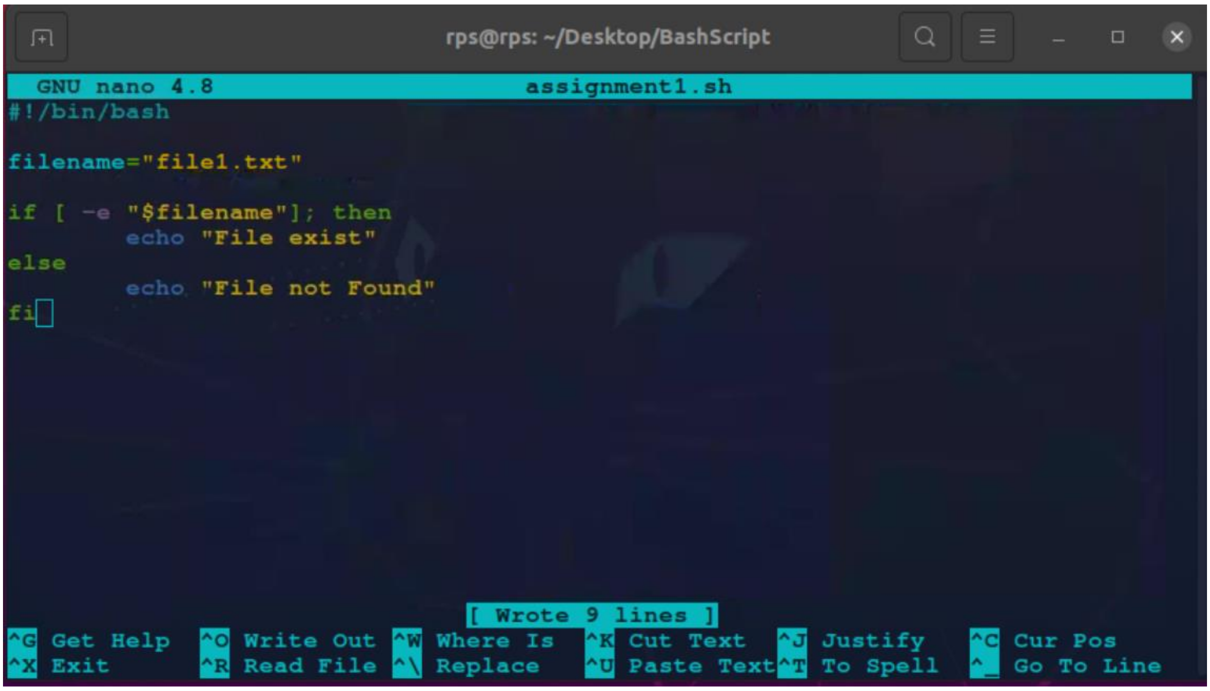
```
chmod +x check_file.sh.
```

You can run this script in the terminal by navigating to the directory containing the script and then executing it:

```
./check_file.sh
```

If `myfile.txt` exists in the same directory as the script, it will print "File exists"; otherwise, it will print "File not found".

Code:



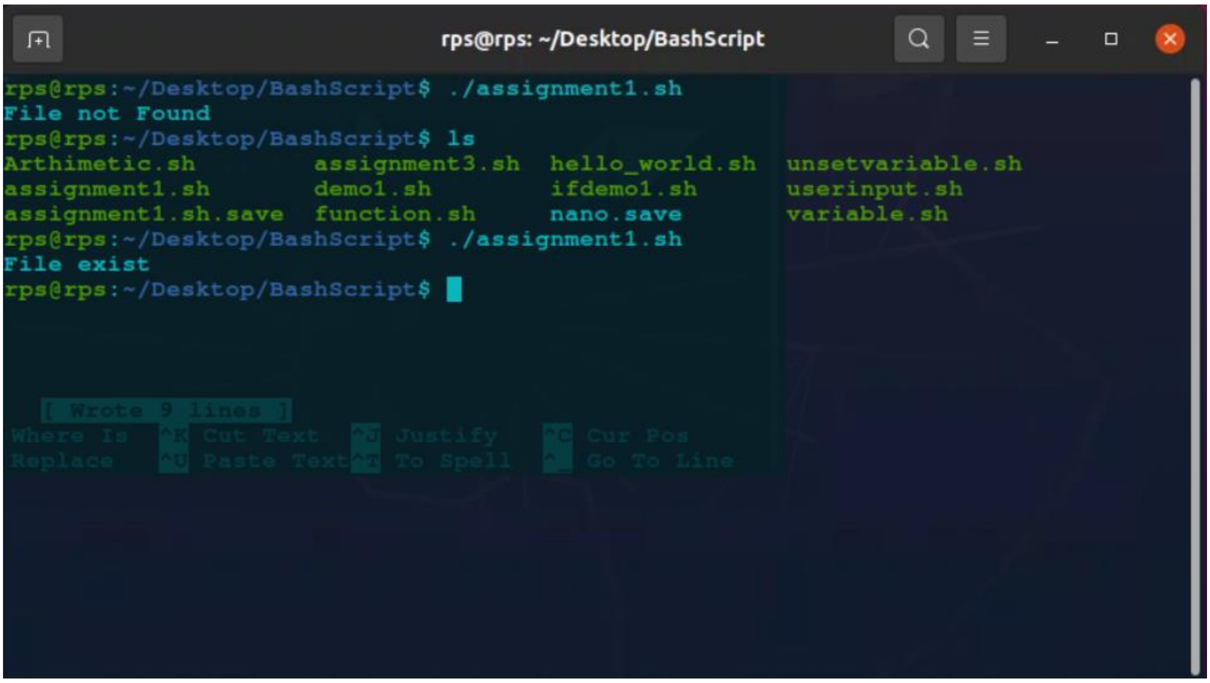
```
  GNU nano 4.8                    assignment1.sh
#!/bin/bash

filename="file1.txt"

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

```
                              [ Wrote 9 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text^T To Spell    ^_ Go To Line
```

Output:



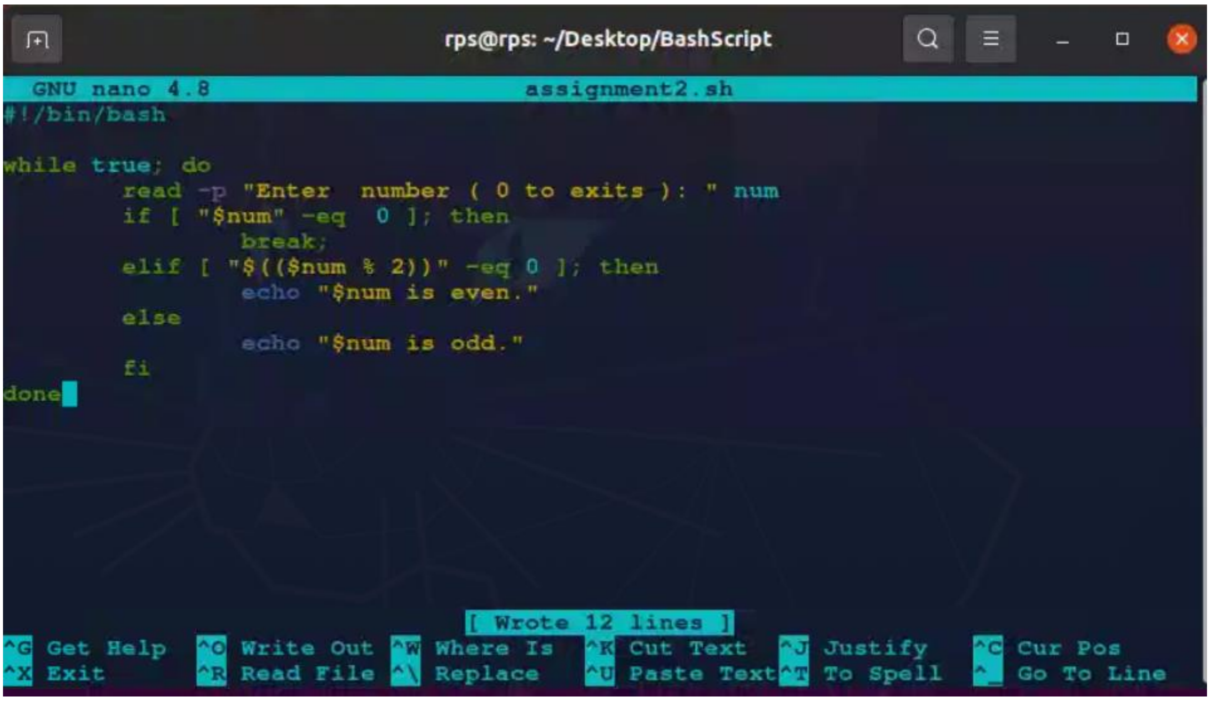
```
rps@rps:~/Desktop/BashScript$ ./assignment1.sh
File not Found
rps@rps:~/Desktop/BashScript$ ls
Arthimetic.sh          assignment3.sh  hello_world.sh  unsetvariable.sh
assignment1.sh         demo1.sh        ifdemo1.sh      userinput.sh
assignment1.sh.save    function.sh     nano.save       variable.sh
rps@rps:~/Desktop/BashScript$ ./assignment1.sh
File exist
rps@rps:~/Desktop/BashScript$ 
```

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 exit): " num
    if [ "$num" -eq 0 ]; then
        break
    elif [ "$(($num % 2))" -eq 0 ]; then
        echo "$num is even."
    else
        echo "$num is odd."
    fi
done
```

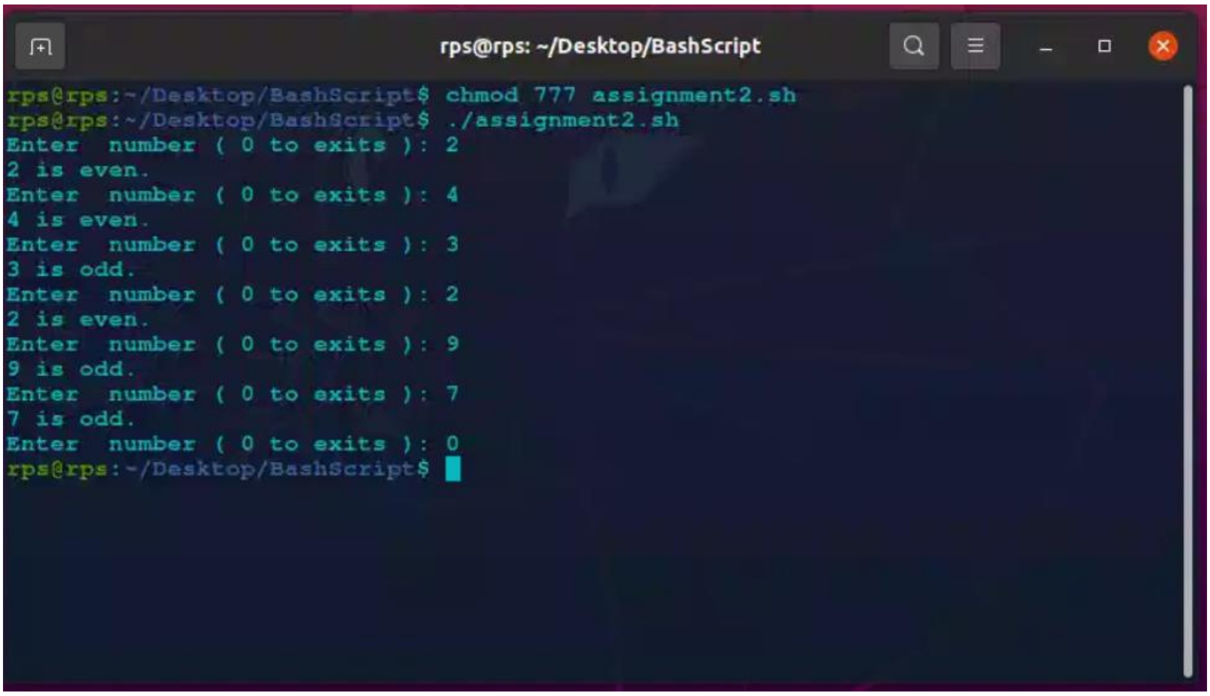
Code:



```
#!/bin/bash

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

Output:



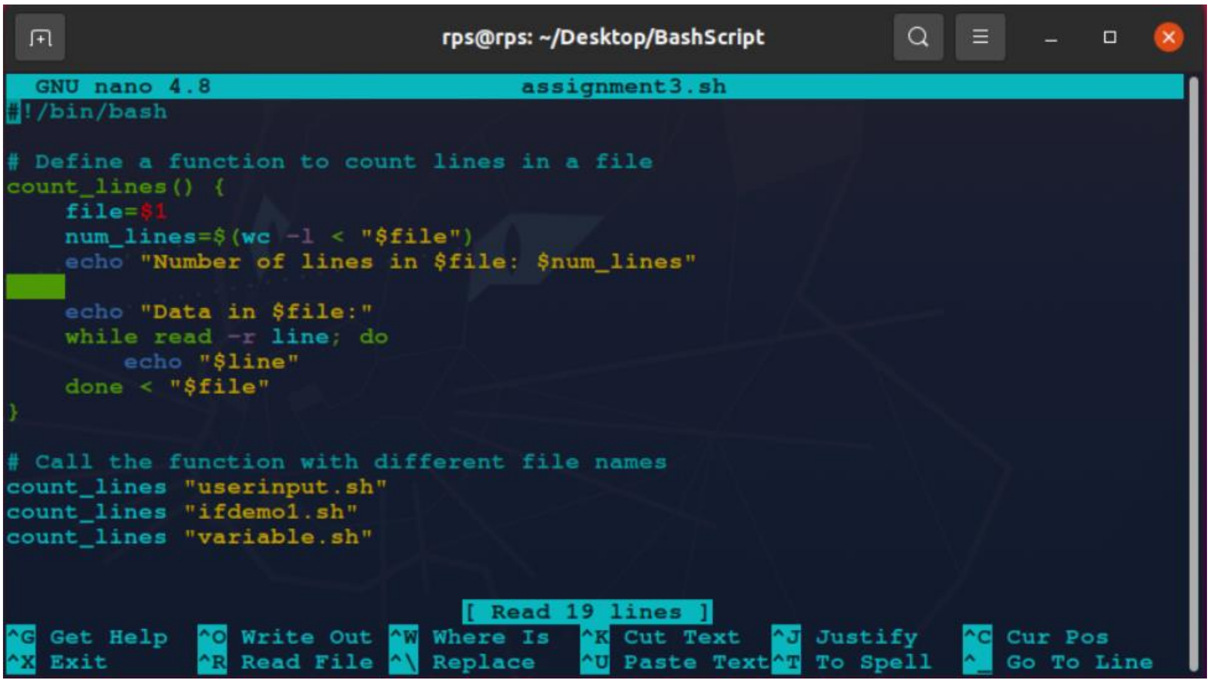
```
rps@rps:~/Desktop/BashScript$ chmod 777 assignment2.sh
rps@rps:~/Desktop/BashScript$ ./assignment2.sh
Enter  number ( 0 to exits ): 2
2 is even.
Enter  number ( 0 to exits ): 4
4 is even.
Enter  number ( 0 to exits ): 3
3 is odd.
Enter  number ( 0 to exits ): 2
2 is even.
Enter  number ( 0 to exits ): 9
9 is odd.
Enter  number ( 0 to exits ): 7
7 is odd.
Enter  number ( 0 to exits ): 0
rps@rps:~/Desktop/BashScript$
```

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() {
    filename="$1"
    if [ -f "$filename" ]; then
        lines=$(wc -l < "$filename")
        echo "Number of lines in $filename: $lines"
    else
        echo "File $filename not found."
    fi
}
# Call the function with different filenames
count_lines "file1.txt"
count_lines "file2.txt"
```

Code:



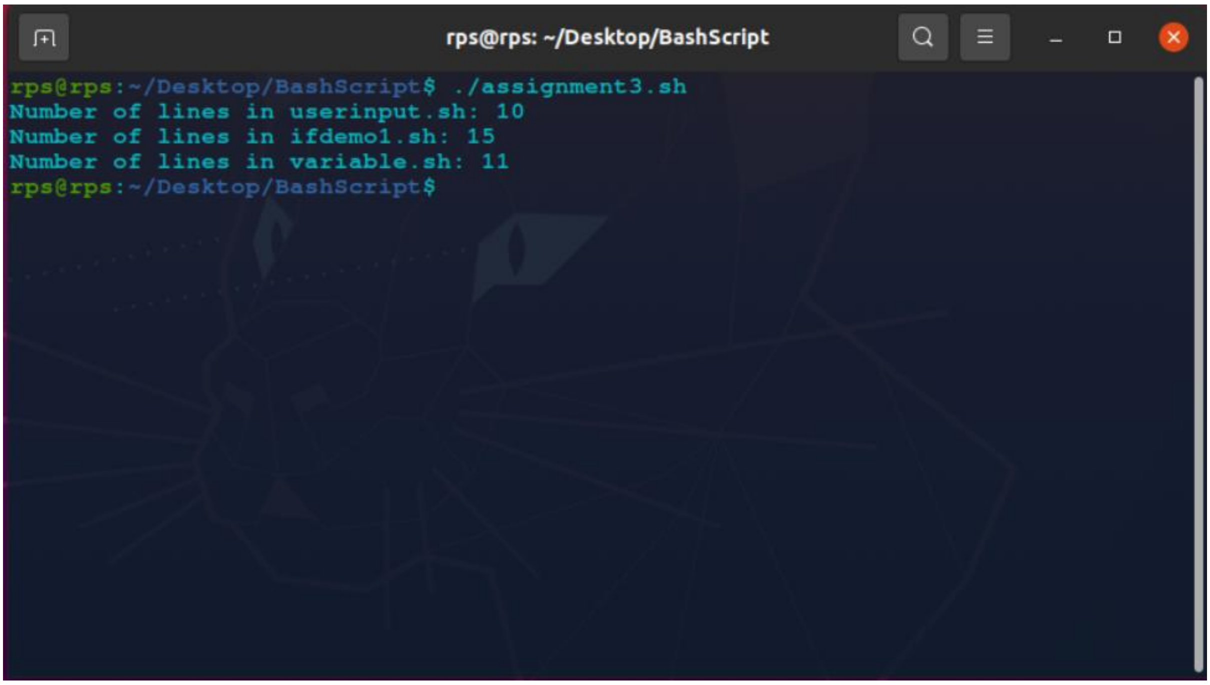
```
#!/bin/bash

# Define a function to count lines in a file
count_lines() {
    file=$1
    num_lines=$(wc -l < "$file")
    echo "Number of lines in $file: $num_lines"

    echo "Data in $file:"
    while read -r line; do
        echo "$line"
    done < "$file"
}

# Call the function with different file names
count_lines "userinput.sh"
count_lines "ifdemo1.sh"
count_lines "variable.sh"
```

Output:



```
rps@rps:~/Desktop/BashScript$ ./assignment3.sh
Number of lines in userinput.sh: 10
Number of lines in ifdemo1.sh: 15
Number of lines in variable.sh: 11
rps@rps:~/Desktop/BashScript$
```