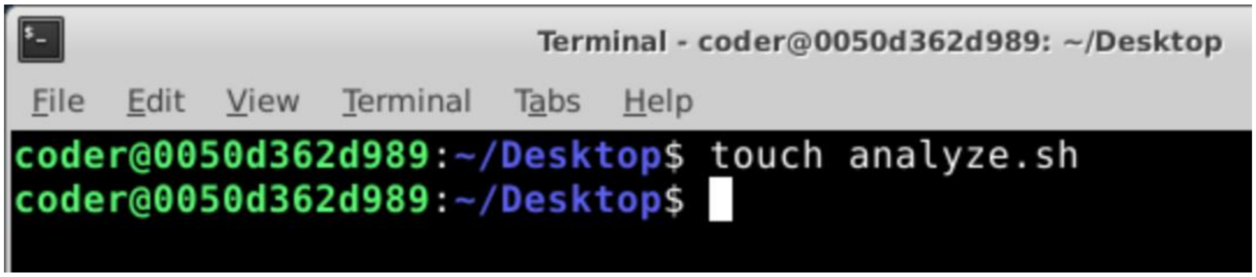
Question 1 [6 Points]

Create a shell script named analyze.sh that accepts exactly ONE command-line argument.
• If the argument is a file: – Display the number of lines, words, and characters in the file.
• If the argument is a directory: – Display the total number of files present. – Display the number of .txt files in the directory.
• If the argument count is invalid or the path does not exist: – Display an appropriate error message.

Answer:

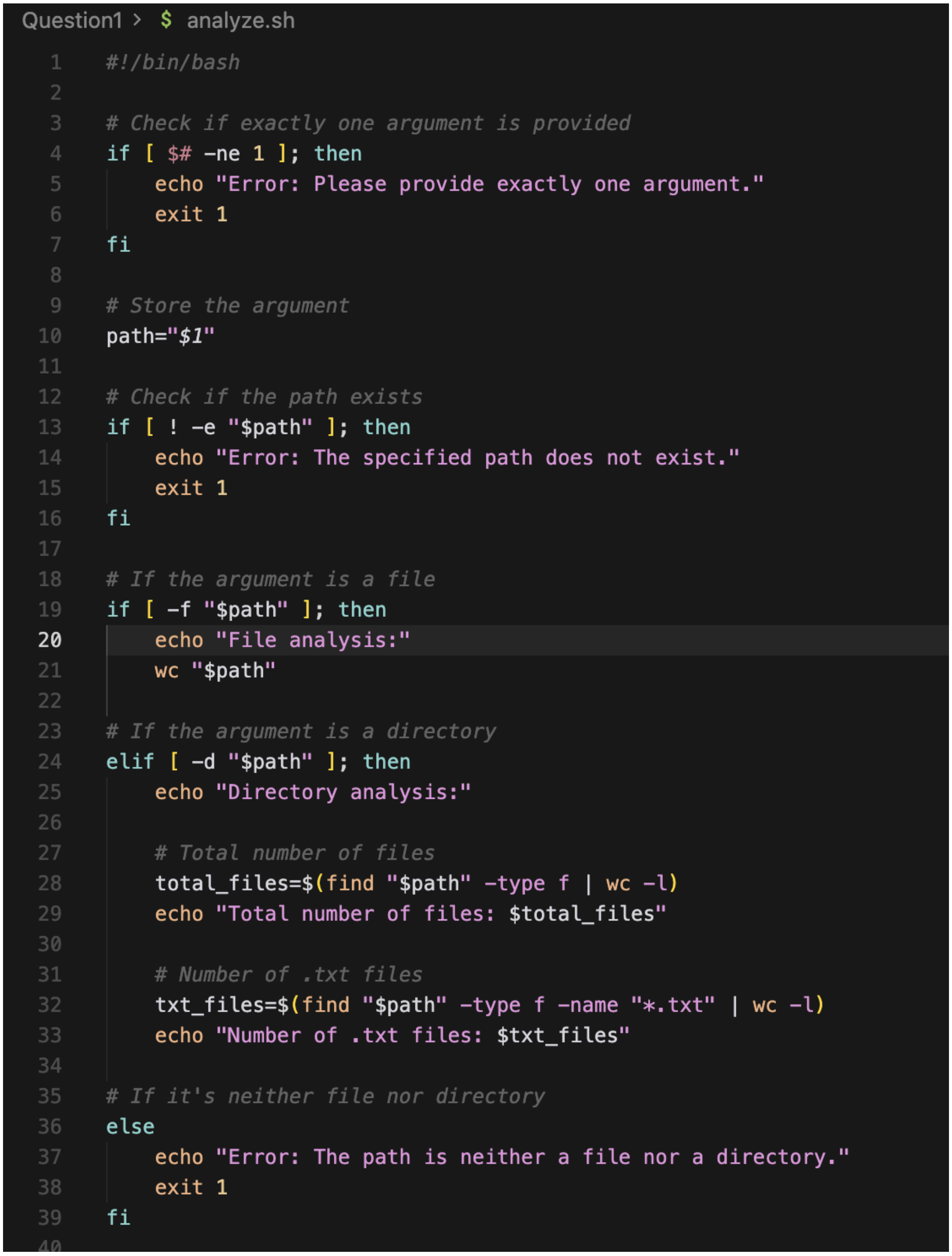
1) Creating a shell scipt file named " analyze.sh " using the touch cmd.


Terminal - coder@0050d362d989: ~/Desktop
File Edit View Terminal Tabs Help

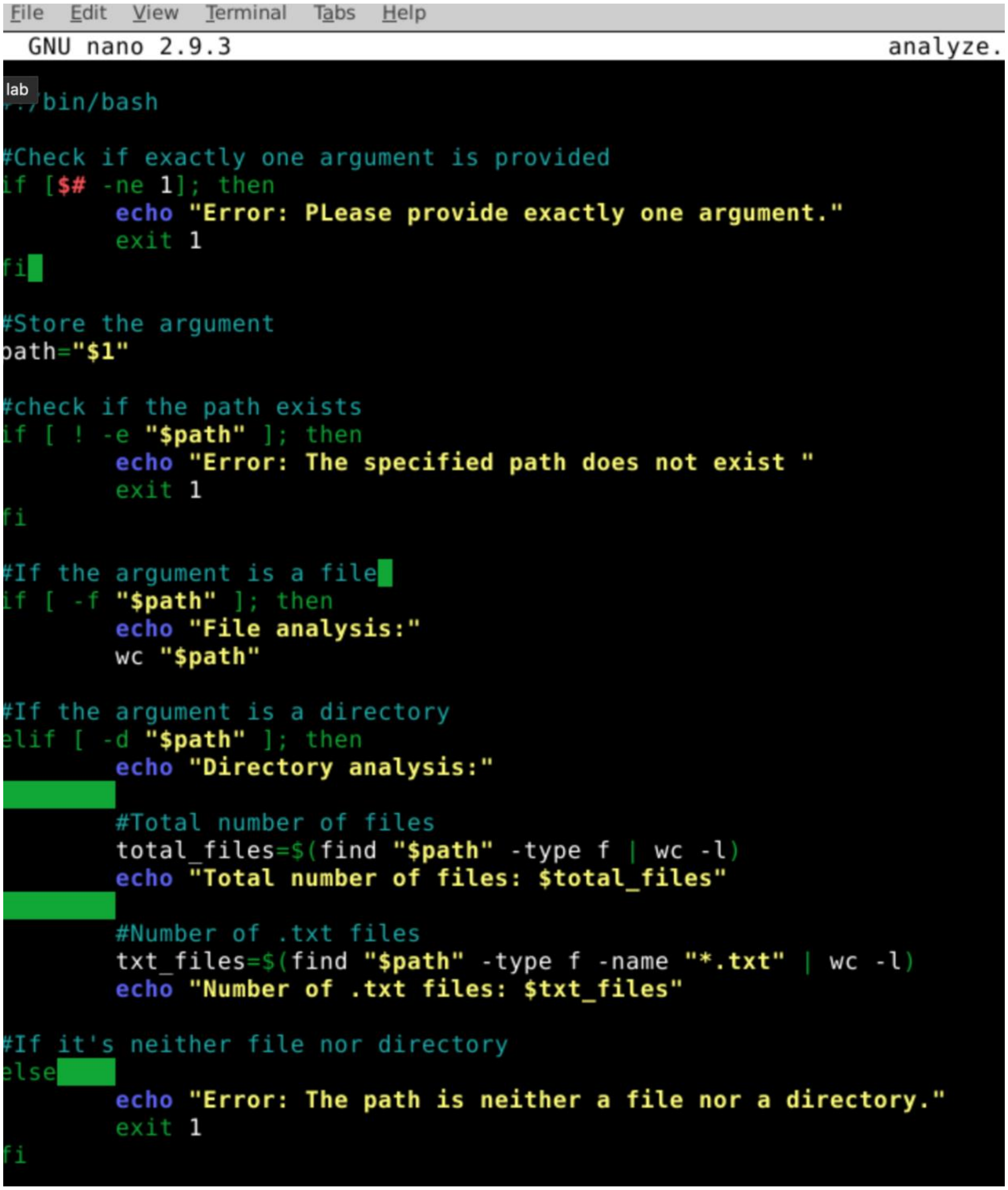
```
coder@0050d362d989:~/Desktop$ touch analyze.sh
coder@0050d362d989:~/Desktop$ 
```

2) I used "nano analyze.sh " to open the file in editor.

3)



```bash
#!/bin/bash

# Check if exactly one argument is provided
if [ $# -ne 1 ]; then
    echo "Error: Please provide exactly one argument."
    exit 1
fi

# Store the argument
path="$1"

# Check if the path exists
if [ ! -e "$path" ]; then
    echo "Error: The specified path does not exist."
    exit 1
fi

# If the argument is a file
if [ -f "$path" ]; then
    echo "File analysis:"
    wc "$path"

# If the argument is a directory
elif [ -d "$path" ]; then
    echo "Directory analysis:"

    # Total number of files
    total_files=$(find "$path" -type f | wc -l)
    echo "Total number of files: $total_files"

    # Number of .txt files
    txt_files=$(find "$path" -type f -name "*.txt" | wc -l)
    echo "Number of .txt files: $txt_files"

# If it's neither file nor directory
else
    echo "Error: The path is neither a file nor a directory."
    exit 1
fi
```



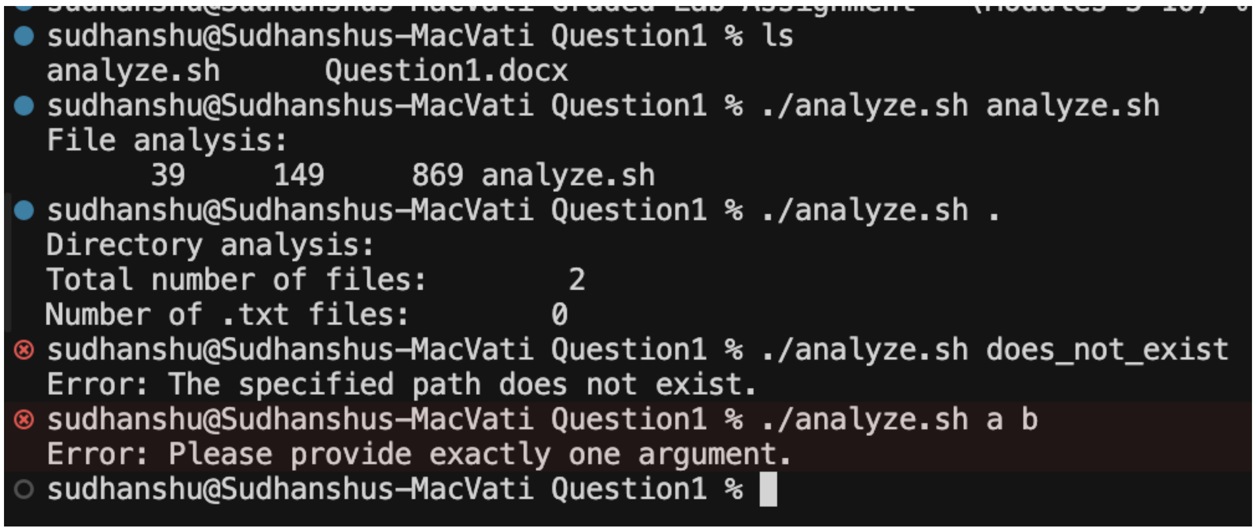
File Edit View Terminal Tabs Help

GNU nano 2.9.3 analyze.

lab

```
/bin/bash

#Check if exactly one argument is provided
if [$# -ne 1]; then
        echo "Error: PLease provide exactly one argument."
        exit 1
fi

#Store the argument
path="$1"

#check if the path exists
if [ ! -e "$path" ]; then
        echo "Error: The specified path does not exist "
        exit 1
fi

#If the argument is a file
if [ -f "$path" ]; then
        echo "File analysis:"
        wc "$path"

#If the argument is a directory
elif [ -d "$path" ]; then
        echo "Directory analysis:"

        #Total number of files
        total_files=$(find "$path" -type f | wc -l)
        echo "Total number of files: $total_files"

        #Number of .txt files
        txt_files=$(find "$path" -type f -name "*.txt" | wc -l)
        echo "Number of .txt files: $txt_files"

#If it's neither file nor directory
else
        echo "Error: The path is neither a file nor a directory."
        exit 1
fi
```

Terminal Output:



```
● sudhanshu@Sudhanshus-MacVati Question1 % ls
  analyze.sh      Question1.docx
● sudhanshu@Sudhanshus-MacVati Question1 % ./analyze.sh analyze.sh
  File analysis:
        39     149     869 analyze.sh
● sudhanshu@Sudhanshus-MacVati Question1 % ./analyze.sh .
  Directory analysis:
  Total number of files:        2
  Number of .txt files:        0
⊗ sudhanshu@Sudhanshus-MacVati Question1 % ./analyze.sh does_not_exist
  Error: The specified path does not exist.
⊗ sudhanshu@Sudhanshus-MacVati Question1 % ./analyze.sh a b
  Error: Please provide exactly one argument.
○ sudhanshu@Sudhanshus-MacVati Question1 %
```

Explanation:

#!/bin/bash - This is telling linux, when someone runs this file, use bash to execute it. Without this line Linux may not know how to interpret the script.

# - Used to put out any comments in the scipt for explanation.

if [ $# -ne 1 ]; then : This starts the conditional statement

-ne : "not equal"

-lt: "less than"

-gt : "greater than"

exit 1: Exit the scipt immediately

fi : means end the if block

wc "$path" : It prints the following things

1. Number of lines
2. Number of words
3. Number of characters
4. File name

total_files=$(find "$path" -type f | wc -l)

-type f : Excludes directories, Only regular files

| : Pipe used to send output of left cmd to right cmd

wc -l : Counts the number of lines