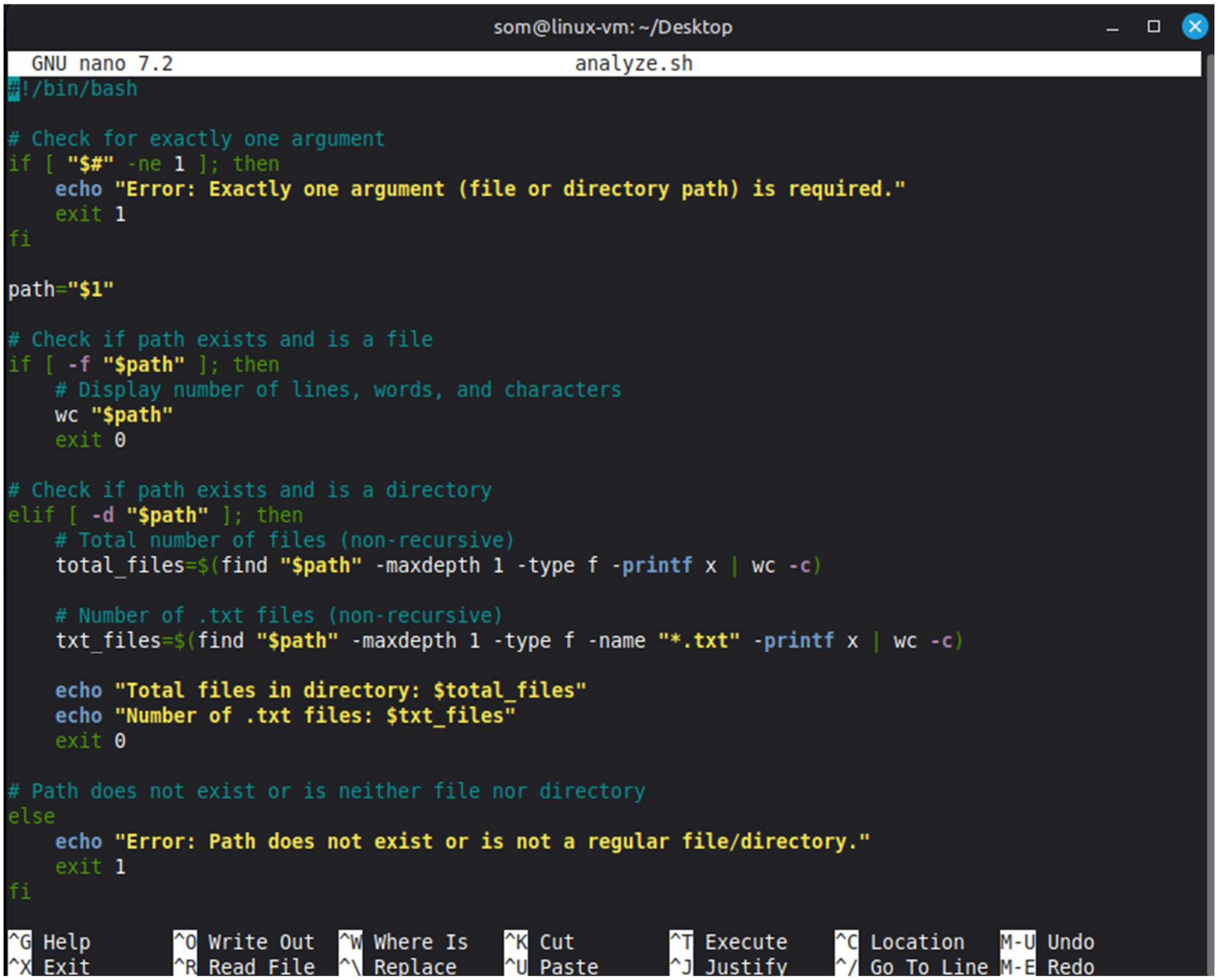
Question 1 (2024eb03003):

Please find screenshot of shell script below and attaching analyze.sh script to GitHub repository:

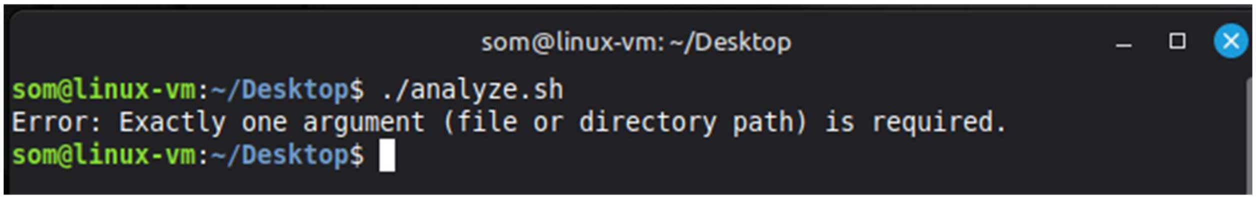


```
#!/bin/bash

# Check for exactly one argument
if [ "$#" -ne 1 ]; then
    echo "Error: Exactly one argument (file or directory path) is required."
    exit 1
fi

path="$1"

# Check if path exists and is a file
if [ -f "$path" ]; then
    # Display number of lines, words, and characters
    wc "$path"
    exit 0

# Check if path exists and is a directory
elif [ -d "$path" ]; then
    # Total number of files (non-recursive)
    total_files=$(find "$path" -maxdepth 1 -type f -printf x | wc -c)

    # Number of .txt files (non-recursive)
    txt_files=$(find "$path" -maxdepth 1 -type f -name "*.txt" -printf x | wc -c)

    echo "Total files in directory: $total_files"
    echo "Number of .txt files: $txt_files"
    exit 0

# Path does not exist or is neither file nor directory
else
    echo "Error: Path does not exist or is not a regular file/directory."
    exit 1
fi
```

Testing the analyze.sh Script

Test 1: Invalid number of arguments

./analyze.sh

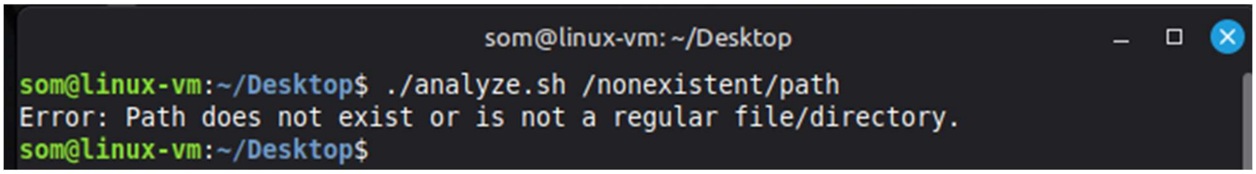


```
som@linux-vm:~/Desktop$ ./analyze.sh
Error: Exactly one argument (file or directory path) is required.
som@linux-vm:~/Desktop$ 
```

Script displays "Error: Exactly one argument (file or directory path) is required." and exits with code 1. This validates the argument count check works correctly.

Test 2: Non-existent path

1. *./analyze.sh /nonexistent/path*

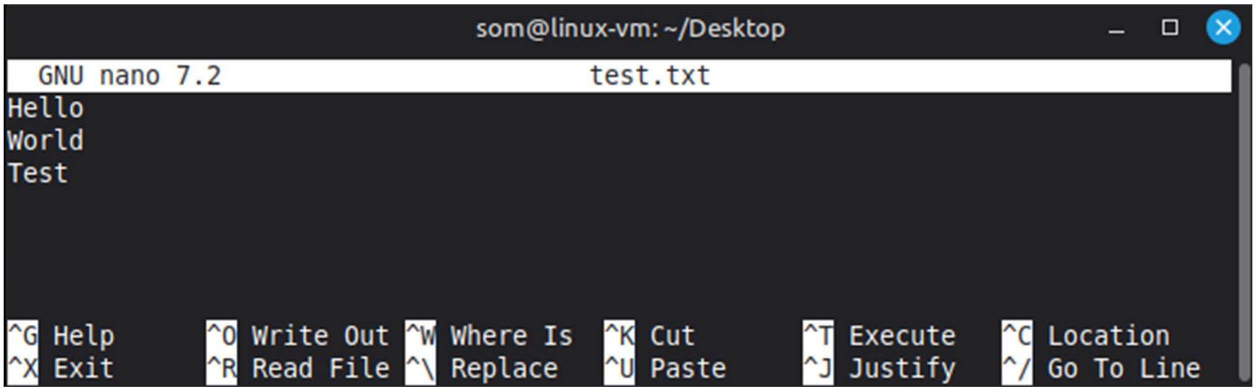


```
som@linux-vm:~/Desktop$ ./analyze.sh /nonexistent/path
Error: Path does not exist or is not a regular file/directory.
som@linux-vm:~/Desktop$
```

Script outputs "Error: Path does not exist or is not a regular file/directory." and exits with code 1. Proper error handling for invalid paths confirmed.

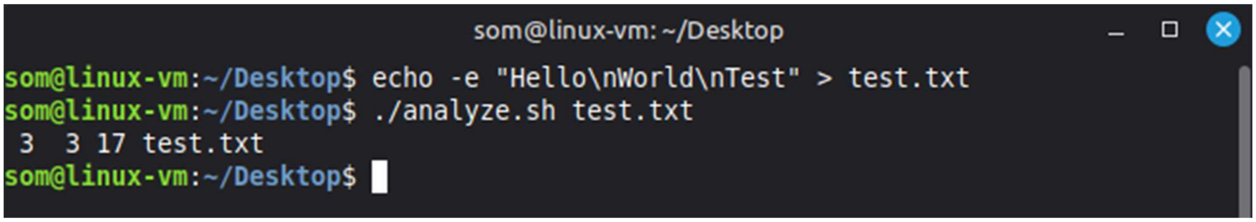
Test 3: Regular text file

echo -e "Hello\nWorld\nTest" > test.txt



```
  GNU nano 7.2                            test.txt
Hello
World
Test
```

./analyze.sh test.txt



```
som@linux-vm:~/Desktop$ echo -e "Hello\nWorld\nTest" > test.txt
som@linux-vm:~/Desktop$ ./analyze.sh test.txt
 3  3 17 test.txt
som@linux-vm:~/Desktop$ 
```

Breakdown:

- 3 lines: "Hello", "World", "Test" (each terminated by \n)
- 3 words: One word per line (no multi-word lines)
- 17 characters: H-e-l-l-o + \n + W-o-r-l-d + \n + T-e-s-t + \n = 17 bytes total

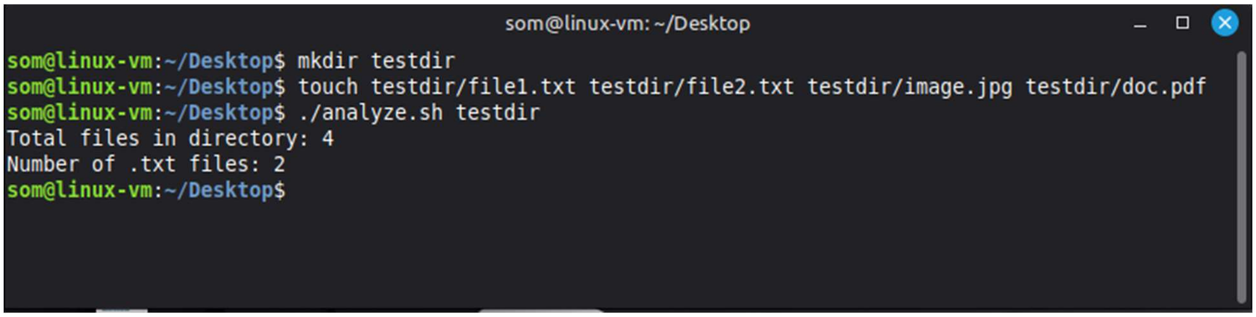
The wc output 3 3 17 test.txt is perfectly correct for this file content. The script analyze.sh works exactly as designed - it passes the file directly to wc, which produces standard output format matching the requirements

Test 4: Directory with mixed files

mkdir testdir

touch testdir/file1.txt testdir/file2.txt testdir/image.jpg testdir/doc.pdf

./analyze.sh testdir



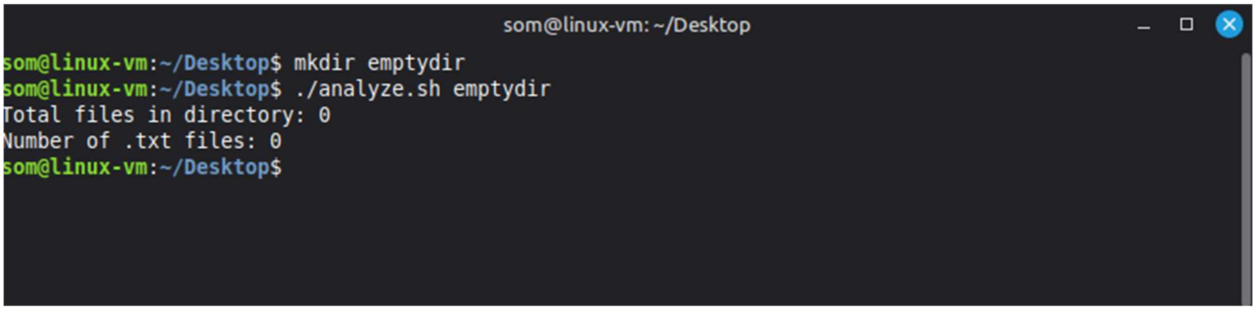
```
som@linux-vm:~/Desktop$ mkdir testdir
som@linux-vm:~/Desktop$ touch testdir/file1.txt testdir/file2.txt testdir/image.jpg testdir/doc.pdf
som@linux-vm:~/Desktop$ ./analyze.sh testdir
Total files in directory: 4
Number of .txt files: 2
som@linux-vm:~/Desktop$
```

Shows "Total files in directory: 4" and "Number of .txt files: 2" accurately. Directory counting logic performs correctly (non-recursive).

Test 5: Empty directory

mkdir emptydir

./analyze.sh emptydir



```
som@linux-vm:~/Desktop$ mkdir emptydir
som@linux-vm:~/Desktop$ ./analyze.sh emptydir
Total files in directory: 0
Number of .txt files: 0
som@linux-vm:~/Desktop$
```

Reports "Total files in directory: 0" and "Number of .txt files: 0". Handles edge case properly.