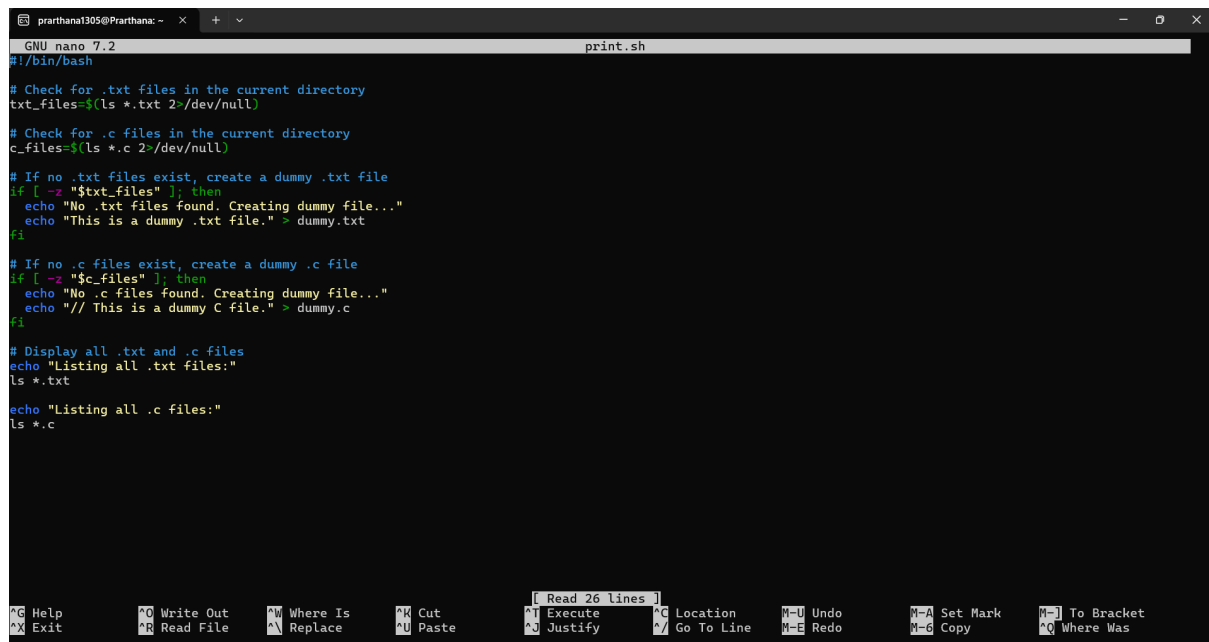


Assignment 3: Shell scripts

Question 1: Script to print All .txt and .c files in the current directory

Shell script to print All .txt and .c files in the current directory:



```
GNU nano 7.2 print.sh
#!/bin/bash

# Check for .txt files in the current directory
txt_files=$(ls *.txt 2>/dev/null)

# Check for .c files in the current directory
c_files=$(ls *.c 2>/dev/null)

# If no .txt files exist, create a dummy .txt file
if [ -z "$txt_files" ]; then
    echo "No .txt files found. Creating dummy file..."
    echo "This is a dummy .txt file." > dummy.txt
fi

# If no .c files exist, create a dummy .c file
if [ -z "$c_files" ]; then
    echo "No .c files found. Creating dummy file..."
    echo "// This is a dummy C file." > dummy.c
fi

# Display all .txt and .c files
echo "Listing all .txt files:"
ls *.txt

echo "Listing all .c files:"
ls *.c
```

Output:



```
prarthana1305@Prarthana:~$ nano print.sh
prarthana1305@Prarthana:~$ chmod +x print.sh
prarthana1305@Prarthana:~$ ./print.sh
No .c files found. Creating dummy file...
Listing all .txt files:
greetings.txt js1.txt js3.txt marksheet.txt numbers.txt
Listing all .c files:
dummy.c
```

Summary of the approach:

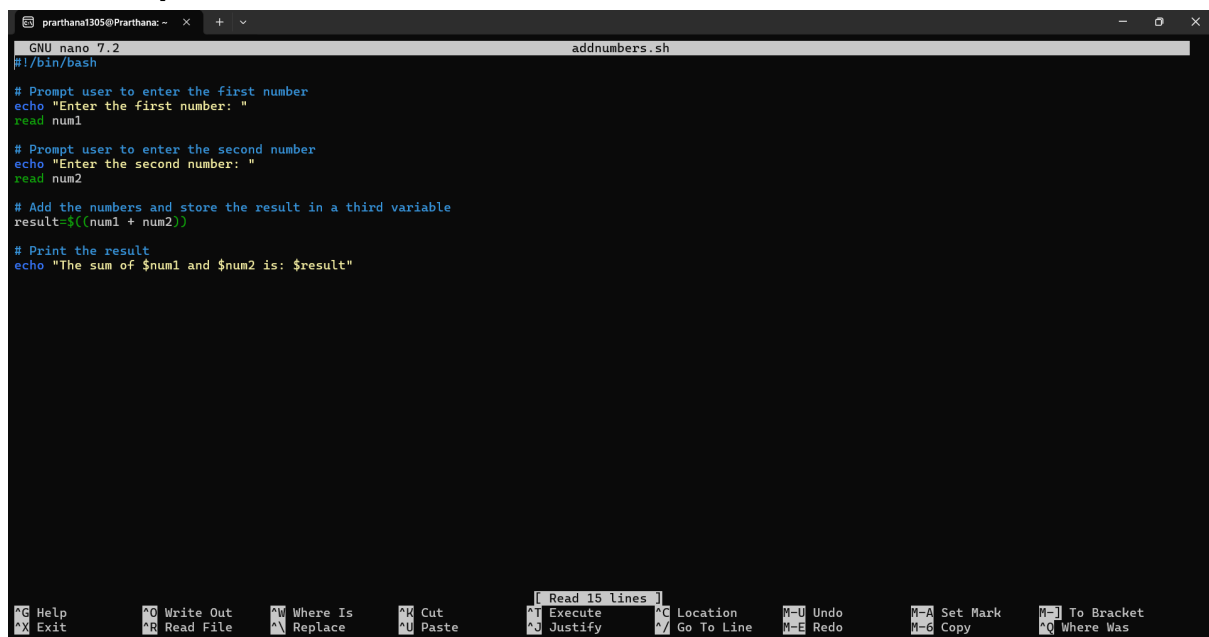
- Checks for .txt and .c files using ls *.txt and ls *.c.
- Creates dummy files if none exist.
- Lists all .txt and .c files in the directory.
-

Keyword Usage:

- ls → Lists files in the directory.
- 2>/dev/null → Suppresses error messages.
- if [-z "\$variable"] → Checks if a variable is empty.
- > → Redirects output to a file (creates or overwrites).
- echo → Writes text to the terminal or files.

Question 2: Script to Add Two Numbers

Shell script to add two numbers:



```
prarthana1305@Prarthana: ~  
GNU nano 7.2 addnumbers.sh  
#!/bin/bash  
  
# Prompt user to enter the first number  
echo "Enter the first number: "  
read num1  
  
# Prompt user to enter the second number  
echo "Enter the second number: "  
read num2  
  
# Add the numbers and store the result in a third variable  
result=$((num1 + num2))  
  
# Print the result  
echo "The sum of $num1 and $num2 is: $result"
```

Output:



```
prarthana1305@Prarthana:~$ nano addnumbers.sh  
prarthana1305@Prarthana:~$ chmod +x addnumbers.sh  
prarthana1305@Prarthana:~$ ./addnumbers.sh  
Enter the first number:  
452  
Enter the second number:  
6589  
The sum of 452 and 6589 is: 7041
```

Summary of the approach to add two numbers

- Reads user input using read for two numbers.
- Performs addition using `$((num1 + num2))`.
- Displays the result using echo.

Keyword Usage:

- `#!/bin/bash` → Specifies the script should be executed with Bash.
- `echo` → Prints messages to the terminal.
- `read` → Captures user input.
- `$((expression))` → Performs arithmetic operations.

Output:

- Checks for .txt and .c files using `ls *.txt` and `ls *.c`.
- Creates dummy files if none exist.
- Lists all .txt and .c files in the directory