BASH PROGRAMS

BASH CASE Programs:

1.

```
harish129@66c459cb7f2a51a: ~/practise/case-statements
  GNU nano 7.2                                  case1.sh
#!/bin/bash

echo "Do you know Java Programming?"
read -p "Yes/No? :" Answer
case $Answer in
    Yes|yes|y|Y)
        echo "That's amazing."
        echo
        ;;
    No|no|N|n)
        echo "It's easy. Let's start learning from javatpoint."
        ;;
esac

                                   [ Read 14 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

OUTPUT:

```
harish129@66c459cb7f2a51a: ~/practise/case-statements
harish129@66c459cb7f2a51a:~/practise/case-statements$ ./case1.sh
Do you know Java Programming?
Yes/No? :Yes
That's amazing.

harish129@66c459cb7f2a51a:~/practise/case-statements$ sudo nano case1.sh
harish129@66c459cb7f2a51a:~/practise/case-statements$ ./case1.sh
Do you know Java Programming?
Yes/No? :no
It's easy. Let's start learning from javatpoint.
harish129@66c459cb7f2a51a:~/practise/case-statements$
```

2.

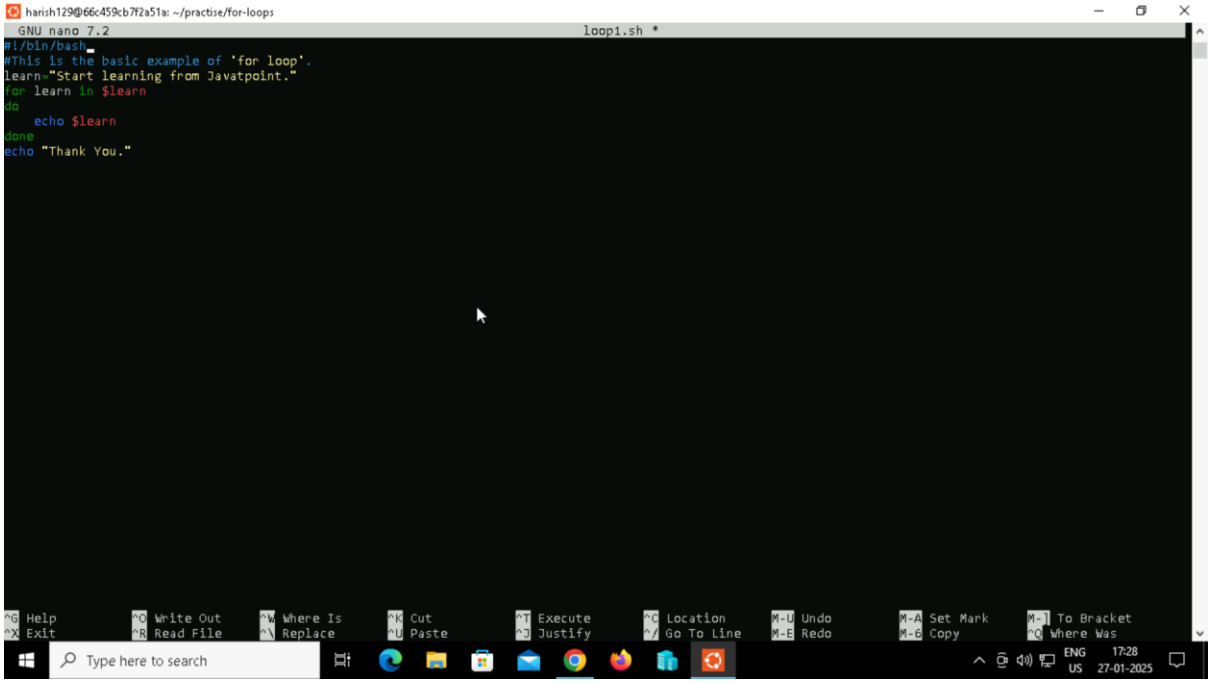
```
harish129@66c459cb7f2a51a: ~/practise/case-statements
  GNU nano 7.2                                  case2.sh *
#!/bin/bash
echo "Which Operating System are you using?"
echo "Windows, Android, Chrome, Linux, Others?"
read -p "Type your OS Name:" OS
case $OS in
    Windows|windows)
        echo "That's common. You should try something new."
        echo
        ;;
    Android|android)
        echo "This is my favorite. It has lots of applications."
        echo
        ;;
    Chrome|chrome)
        echo "Cool!!! It's for pro users. Amazing Choice."
        echo
        ;;
    Linux|linux)
        echo "You might be serious about security!!"
        echo
        ;;
    *)
        echo "Sounds interesting. I will try that.
        echo
        ;;
esac

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was
```

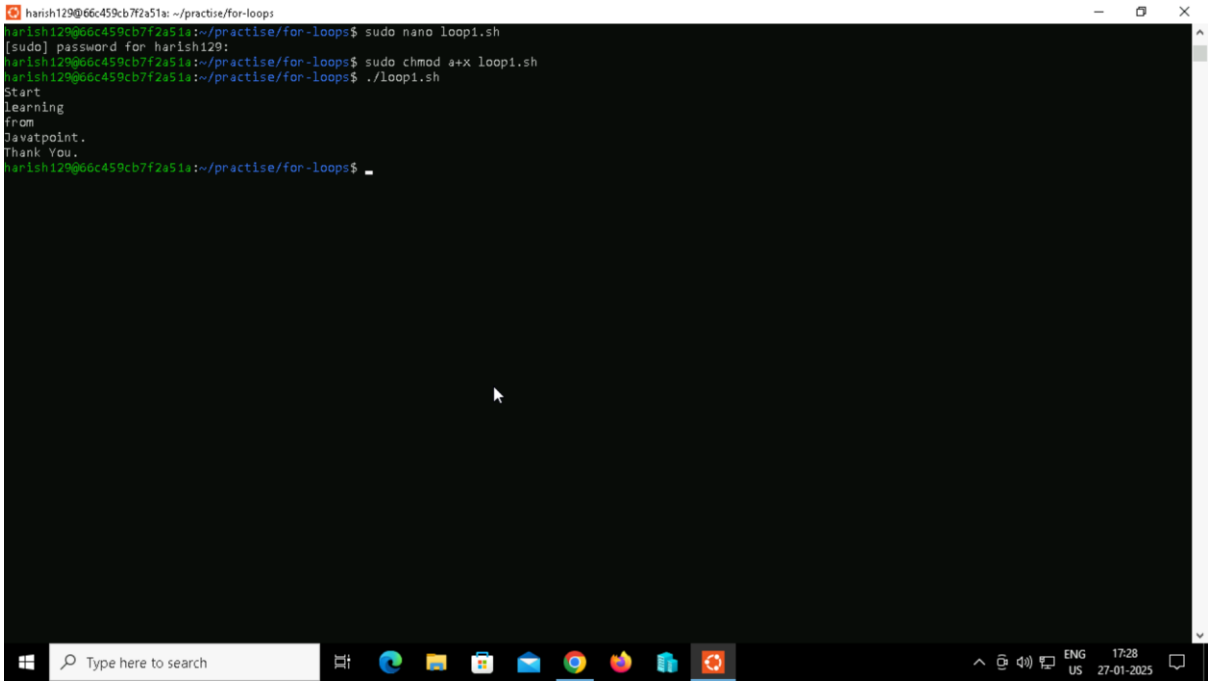
OUTPUT:

```
harish129@66c459cb7f2a51a: ~/practise/case-statements
harish129@66c459cb7f2a51a:~/practise/case-statements$ sudo nano case2.sh
harish129@66c459cb7f2a51a:~/practise/case-statements$ ./case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:Android
This is my favorite. It has lots of applications.

harish129@66c459cb7f2a51a:~/practise/case-statements$ ./case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:Linux
You might be serious about security!!

harish129@66c459cb7f2a51a:~/practise/case-statements$ ./case2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name:Windows
That's common. You should try something new.

harish129@66c459cb7f2a51a:~/practise/case-statements$
```

For Loops:

1.


harish129@66c459cb7f2a51a: ~/practise/for-loops
GNU nano 7.2
loop1.sh *
#!/bin/bash
#This is the basic example of 'for loop'.
learn="Start learning from Javatpoint."
for learn in $learn
do
echo $learn
done
echo "Thank You."
^G Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut
^U Paste
^T Execute
^J Justify
^C Location
^/ Go To Line
M-U Undo
M-E Redo
M-A Set Mark
M-6 Copy
M-] To Bracket
^Q Where Was
Type here to search
ENG
US
17:28
27-01-2025


harish129@66c459cb7f2a51a: ~/practise/for-loops
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo nano loop1.sh
[sudo] password for harish129:
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo chmod a+x loop1.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ ./loop1.sh
Start
learning
from
Javatpoint.
Thank You.
harish129@66c459cb7f2a51a:~/practise/for-loops$
Type here to search
ENG
US
17:28
27-01-2025

2.

```
harish129@66c459cb7f2a51a: ~/practise/for-loops
  GNU nano 7.2                                   loop2.sh *
#!/bin/bash
#This is the basic example to print a series of numbers from 1 to 10.

for num in {1..10}
do
    echo $num
done

echo "Series of numbers from 1 to 10."

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo      M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo      M-6 Copy       ^Q Where Was
```

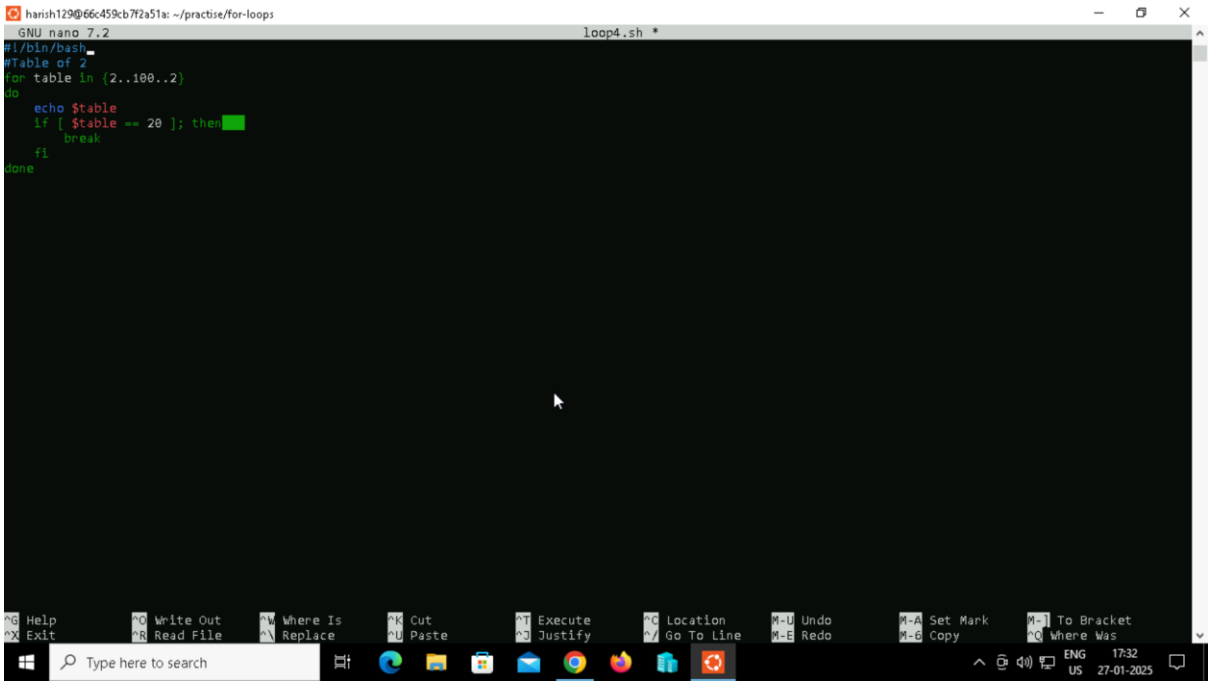
```
harish129@66c459cb7f2a51a: ~/practise/for-loops
harish129@66c459cb7f2a51a:~/practise$ mkdir for-loops
harish129@66c459cb7f2a51a:~/practise$ cd for-loops
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo nano loop1.sh
[sudo] password for harish129:
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo chmod a+x loop1.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ ./loop1.sh
Start
learning
from
Javatpoint.
Thank You.
harish129@66c459cb7f2a51a:~/practise/for-loops$ cp loop1.sh loop2.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo nano loop2.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ ./loop2.sh
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
harish129@66c459cb7f2a51a:~/practise/for-loops$
```

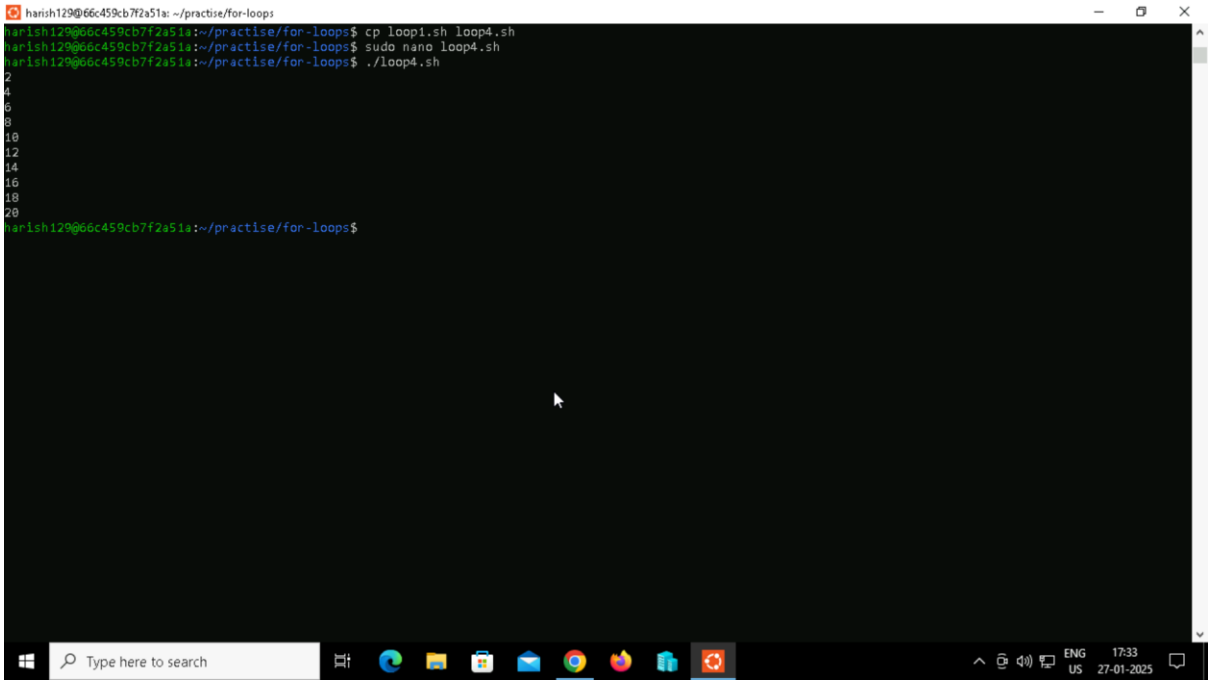
3.

```
harish129@66c459cb7f2a51a: ~/practise/for-loops
  GNU nano 7.2                                   loop3.sh *
#!/bin/bash
#Array Declaration
arr=( "Welcome" "to" "Javatpoint" )
for i in "${arr[@]}"
do
    echo $i
done

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo      M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo      M-6 Copy       ^Q Where Was
```

```
harish129@66c459cb7f2a51a: ~/practise/for-loops
8
9
10
Series of numbers from 1 to 10.
harish129@66c459cb7f2a51a:~/practise/for-loops$ cp loop1.sh loop3.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo nano loop3.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ ./loop3.sh
Welcome
to
Javatpoint
harish129@66c459cb7f2a51a:~/practise/for-loops$
```

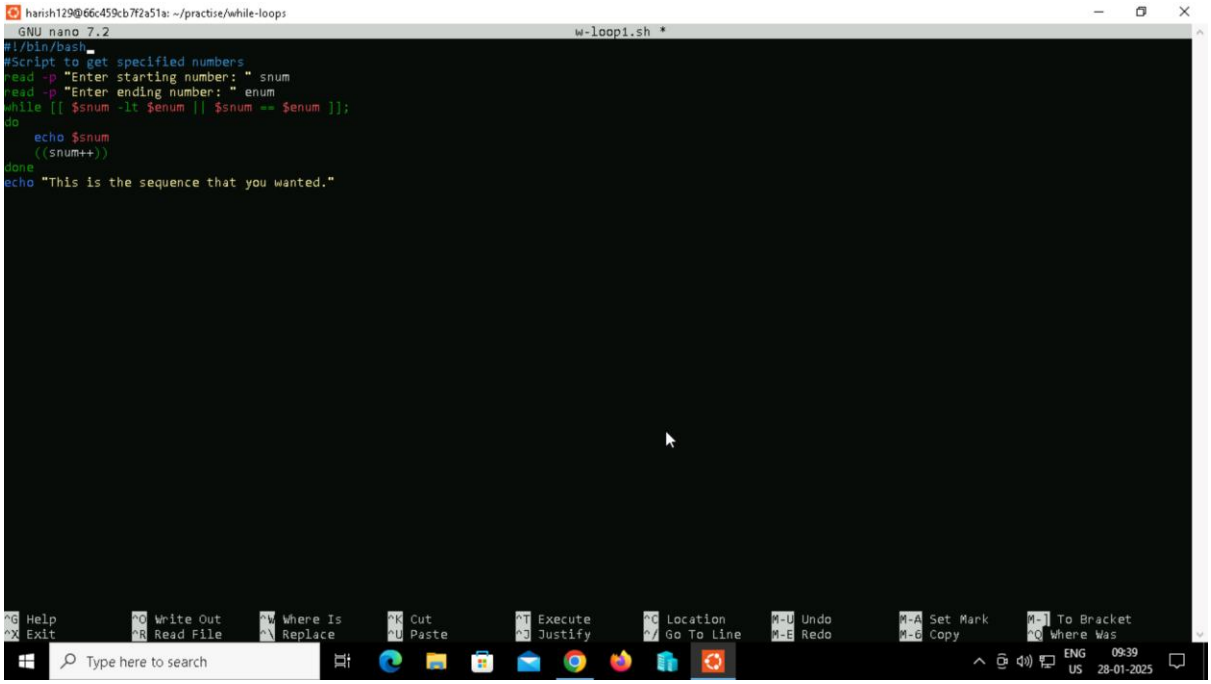
4.


harish129@66c459cb7f2a51a: ~/practise/for-loops
GNU nano 7.2
loop4.sh *
#!/bin/bash
#Table of 2
for table in {2..100..2}
do
echo $table
if [$table == 20]; then
break
fi
done
^G Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut
^U Paste
^T Execute
^J Justify
^C Location
^/ Go To Line
M-U Undo
M-E Redo
M-A Set Mark
M-6 Copy
M-] To Bracket
^Q Where Was
Type here to search
ENG
US
17:32
27-01-2025

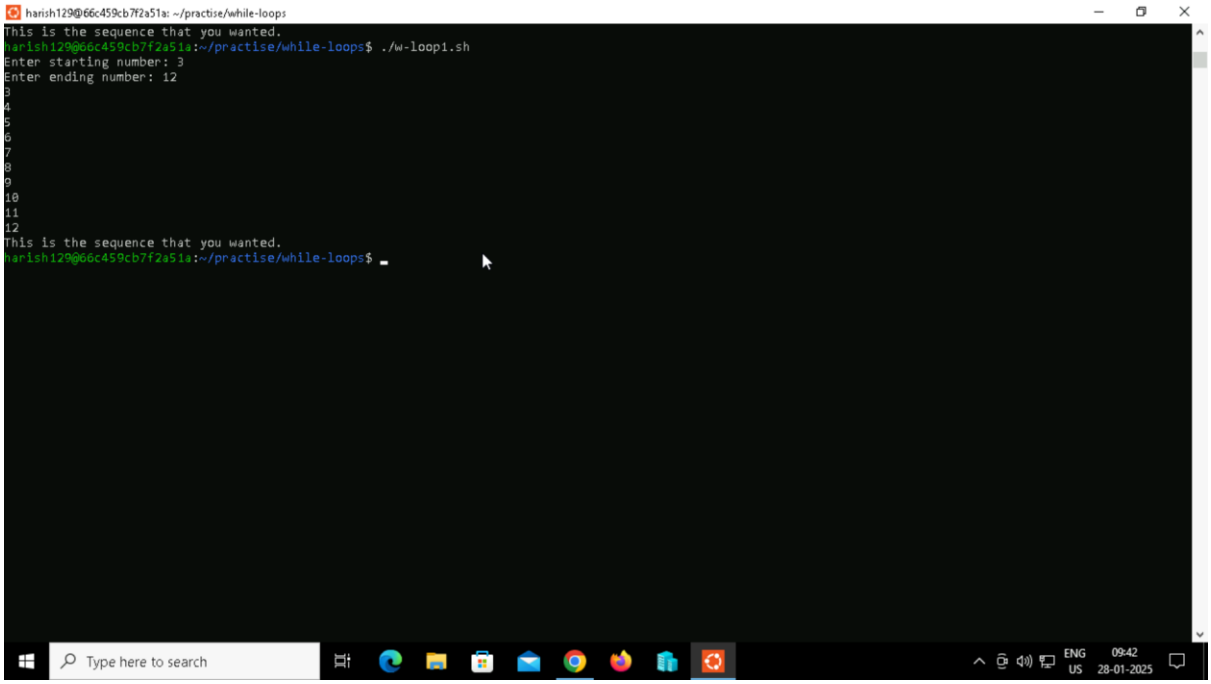

harish129@66c459cb7f2a51a: ~/practise/for-loops
harish129@66c459cb7f2a51a:~/practise/for-loops$ cp loop1.sh loop4.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ sudo nano loop4.sh
harish129@66c459cb7f2a51a:~/practise/for-loops$ ./loop4.sh
2
4
6
8
10
12
14
16
18
20
harish129@66c459cb7f2a51a:~/practise/for-loops$
Type here to search
ENG
US
17:33
27-01-2025

WHILE-LOOPS:

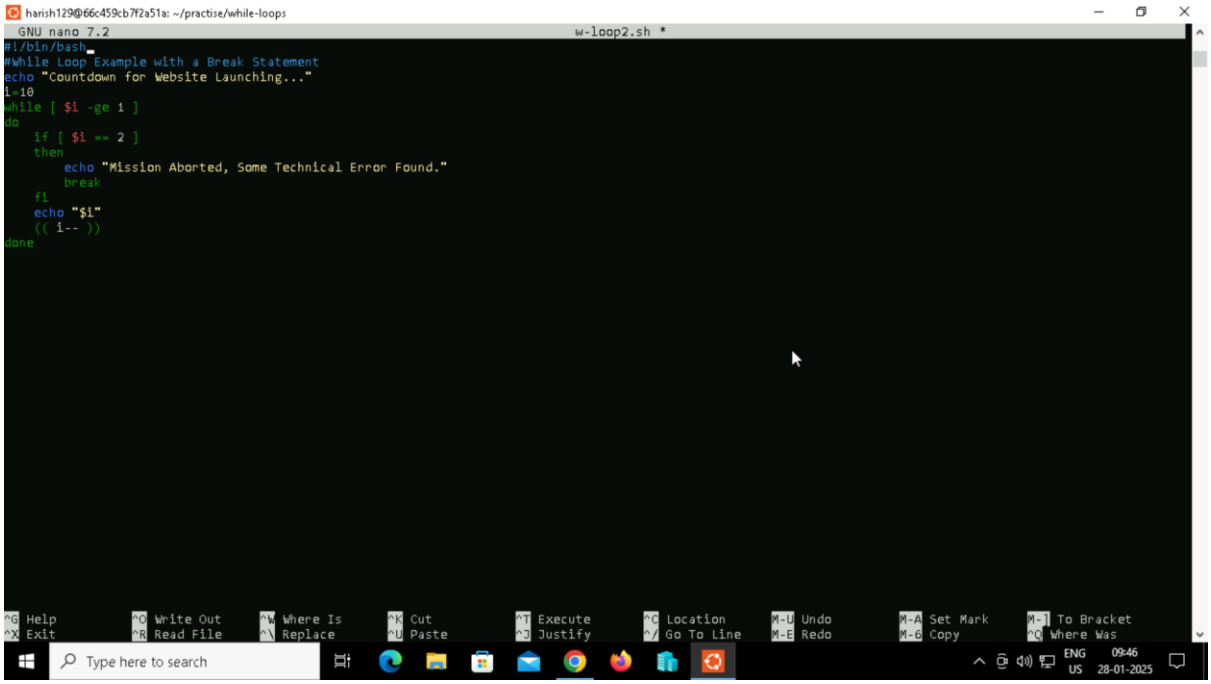
1. While loop with Multiple Conditions.



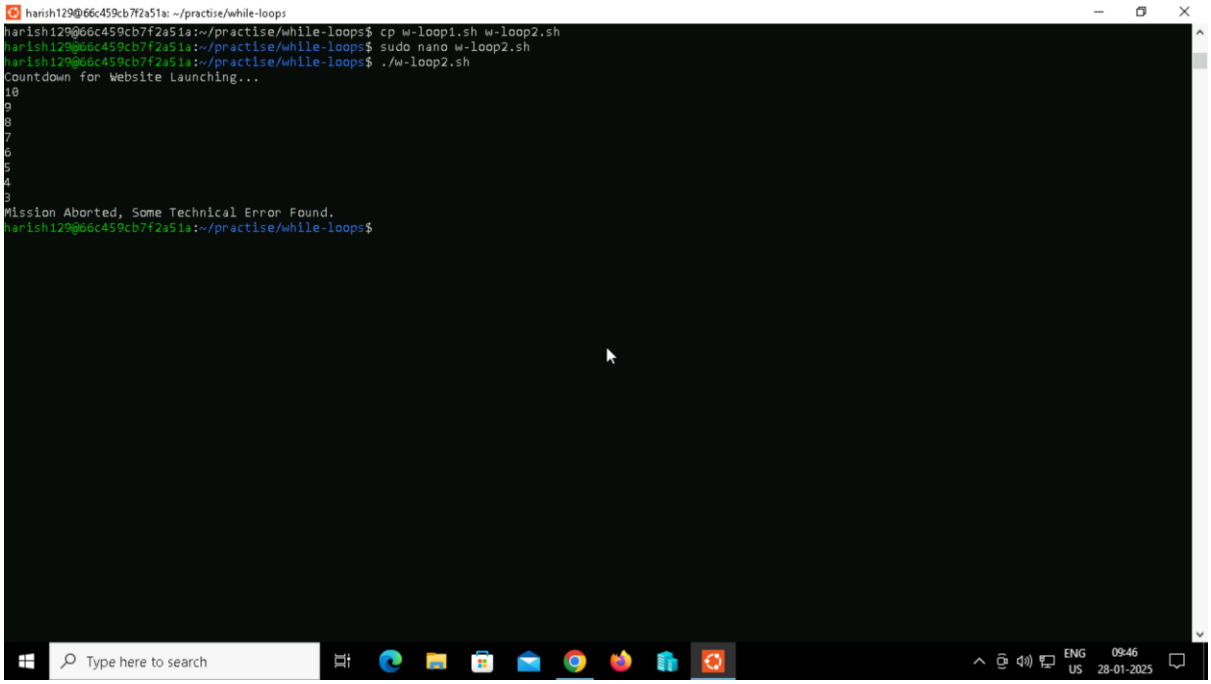
Output:



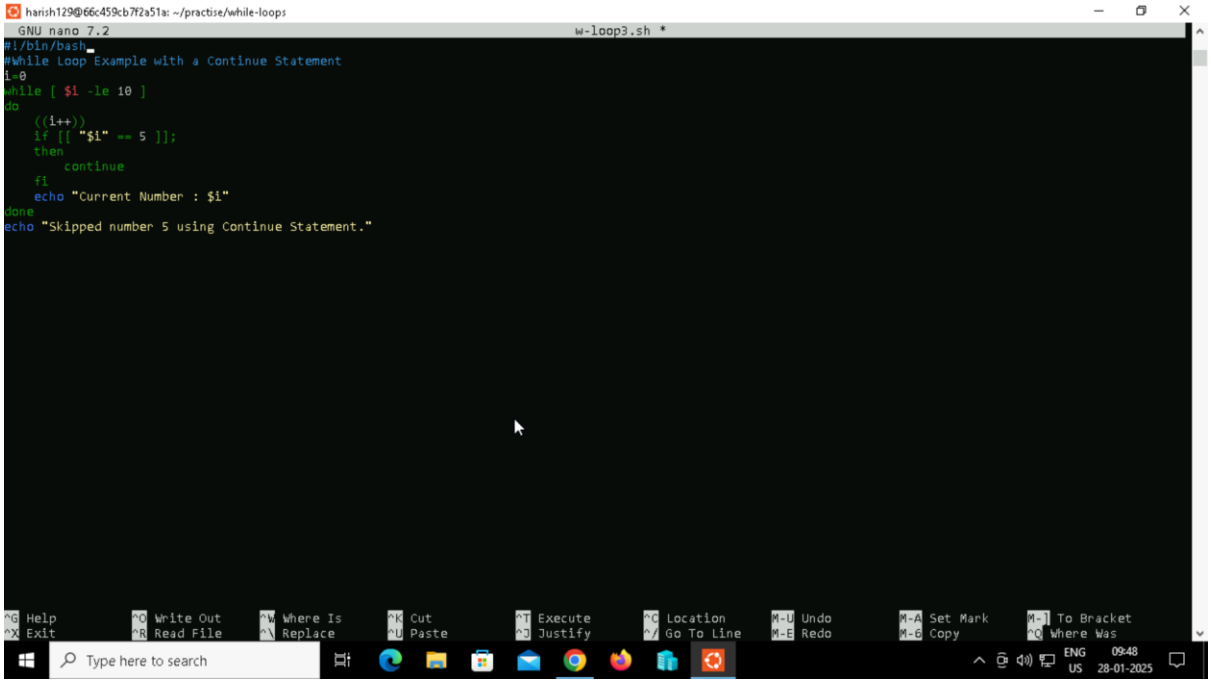
2. While loop with Break Statement



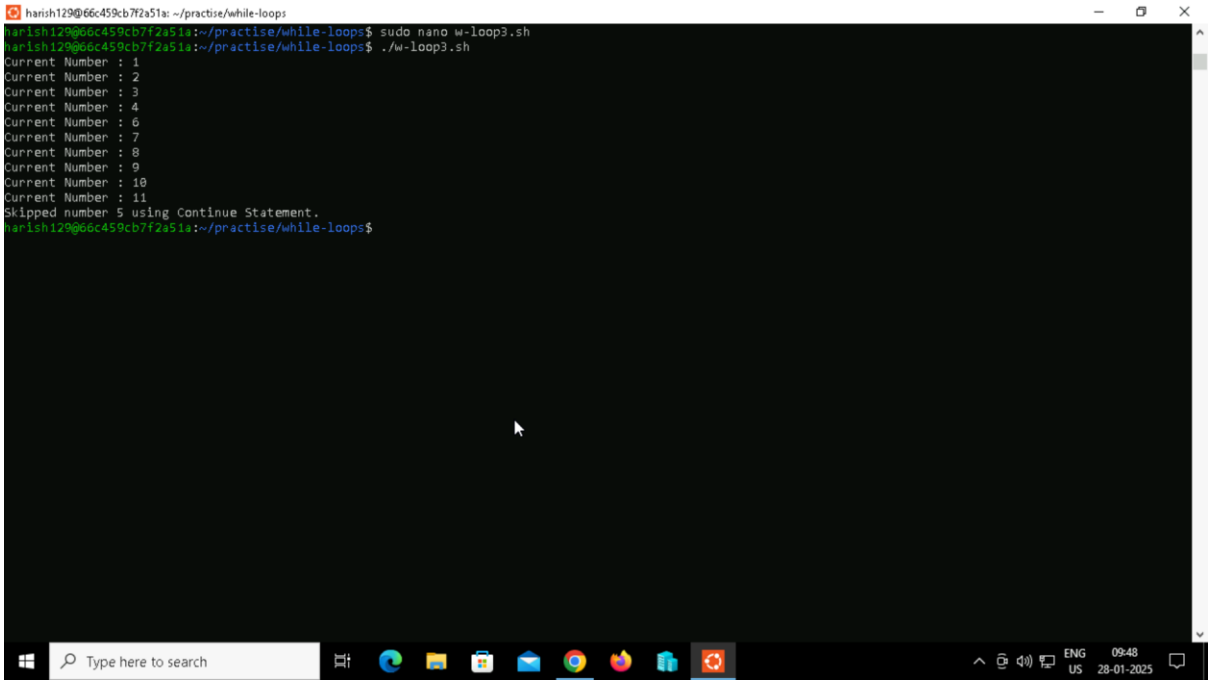
Output:



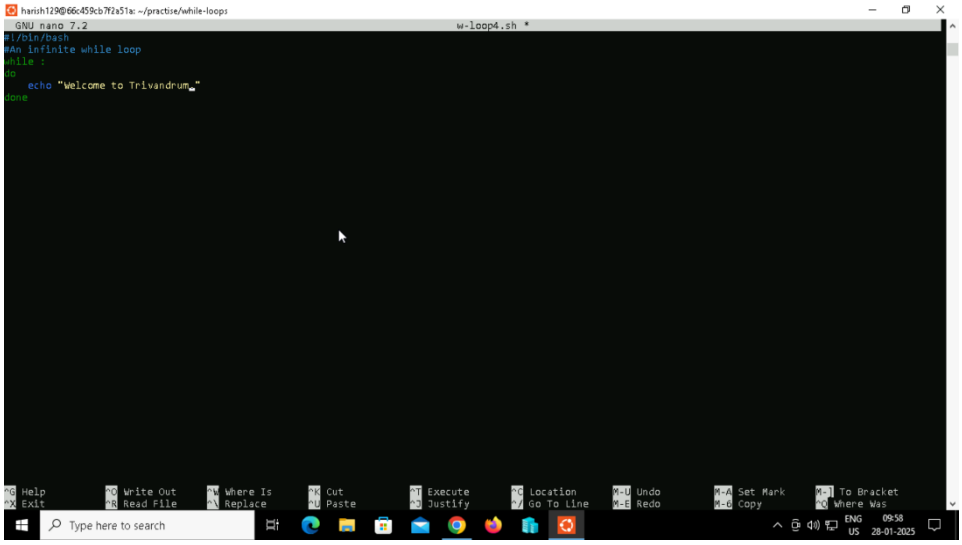
3. While – loop with Continue statement:



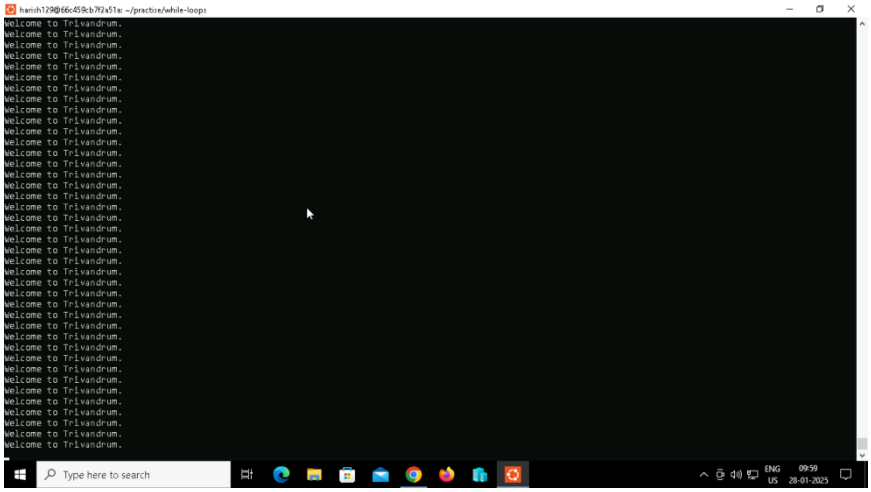
Output:

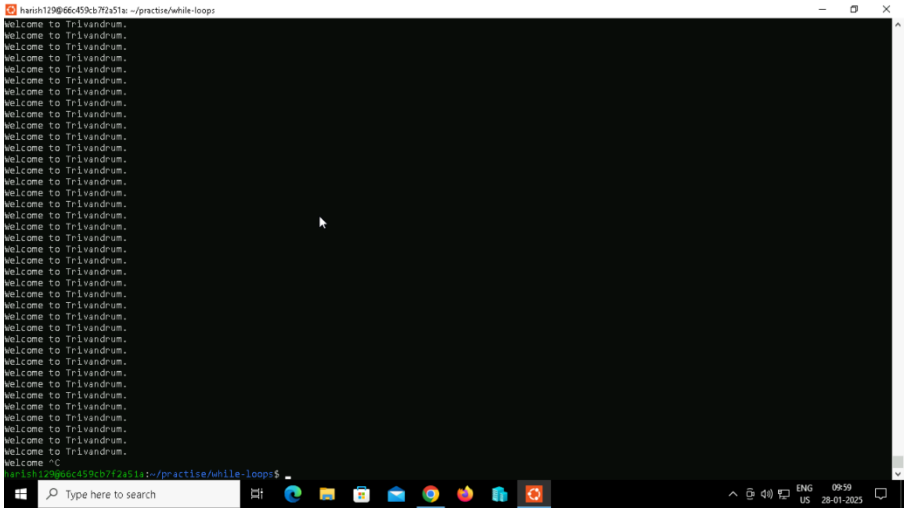


4. Infinite While-loop :



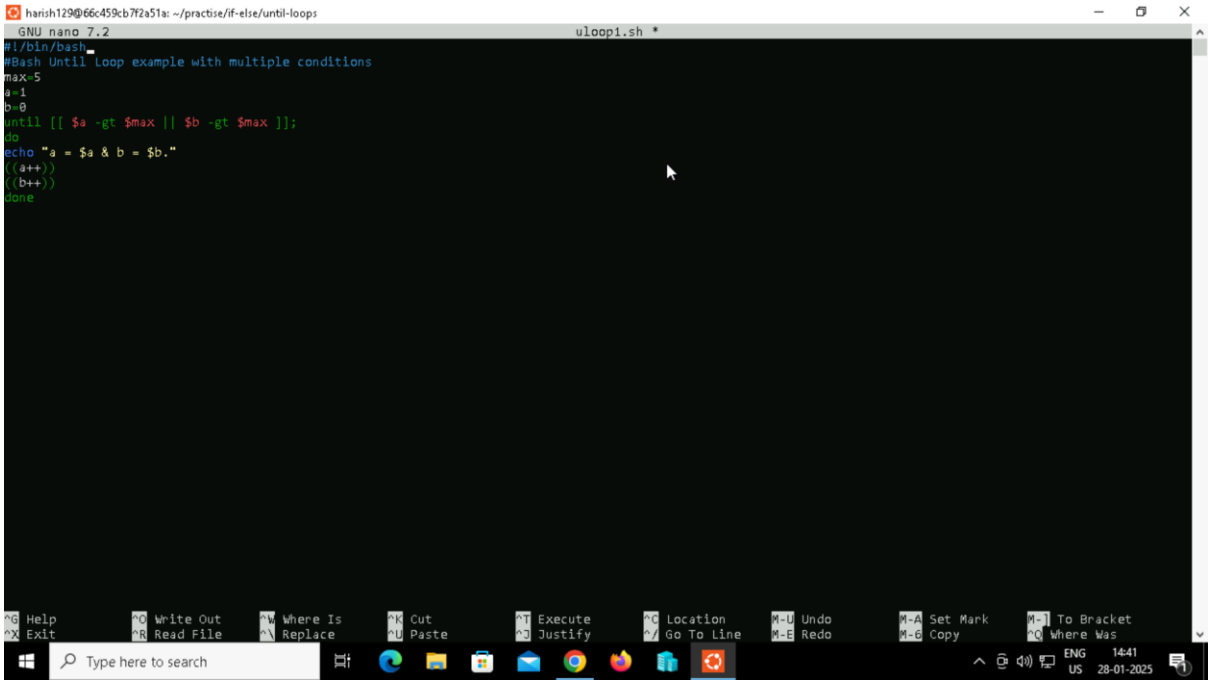
Output:



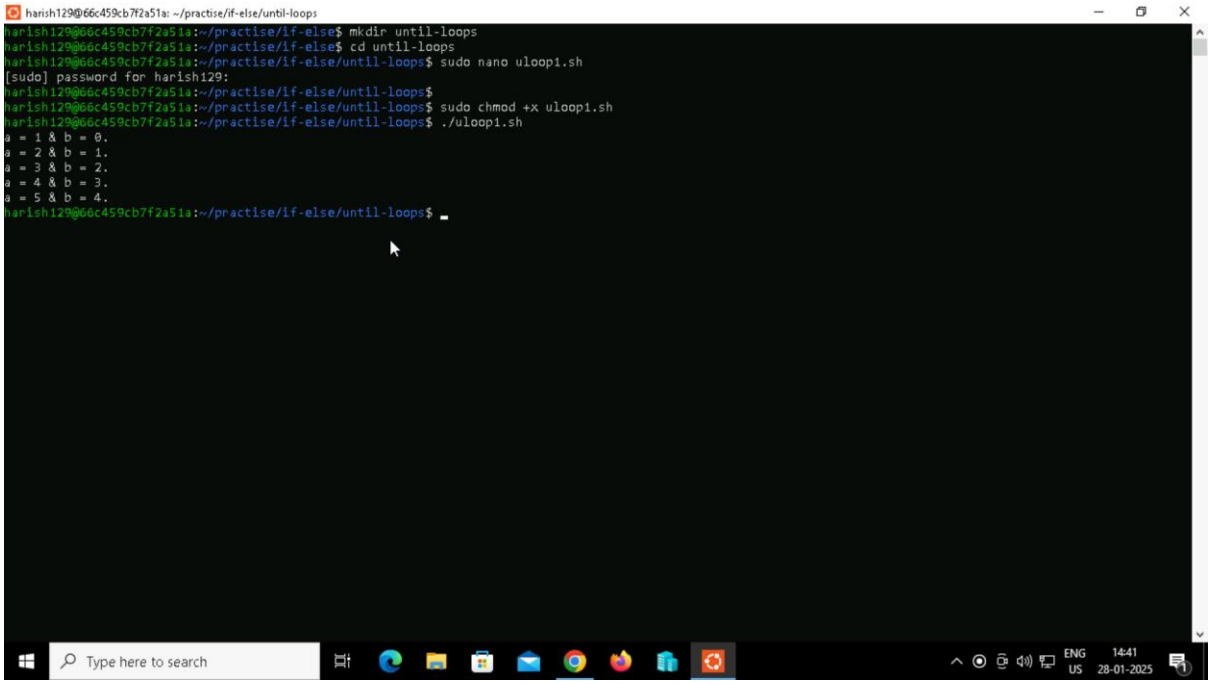


UNTIL LOOPS:

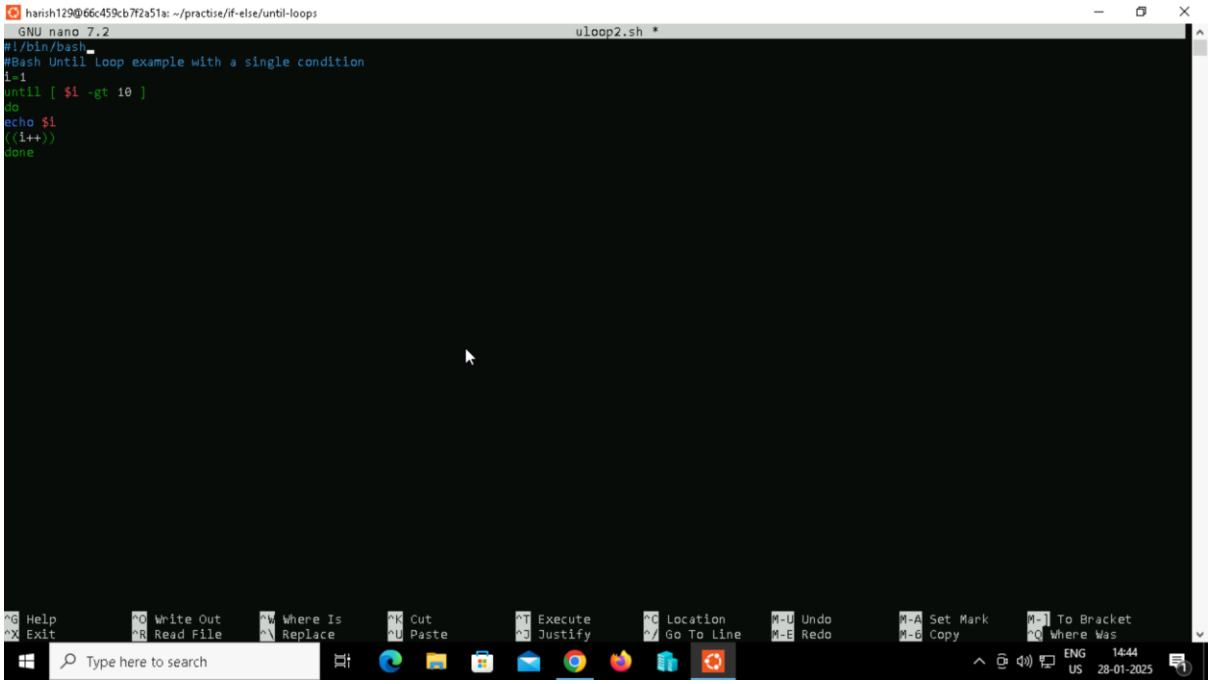
1. Until loop with Multiple Conditions:



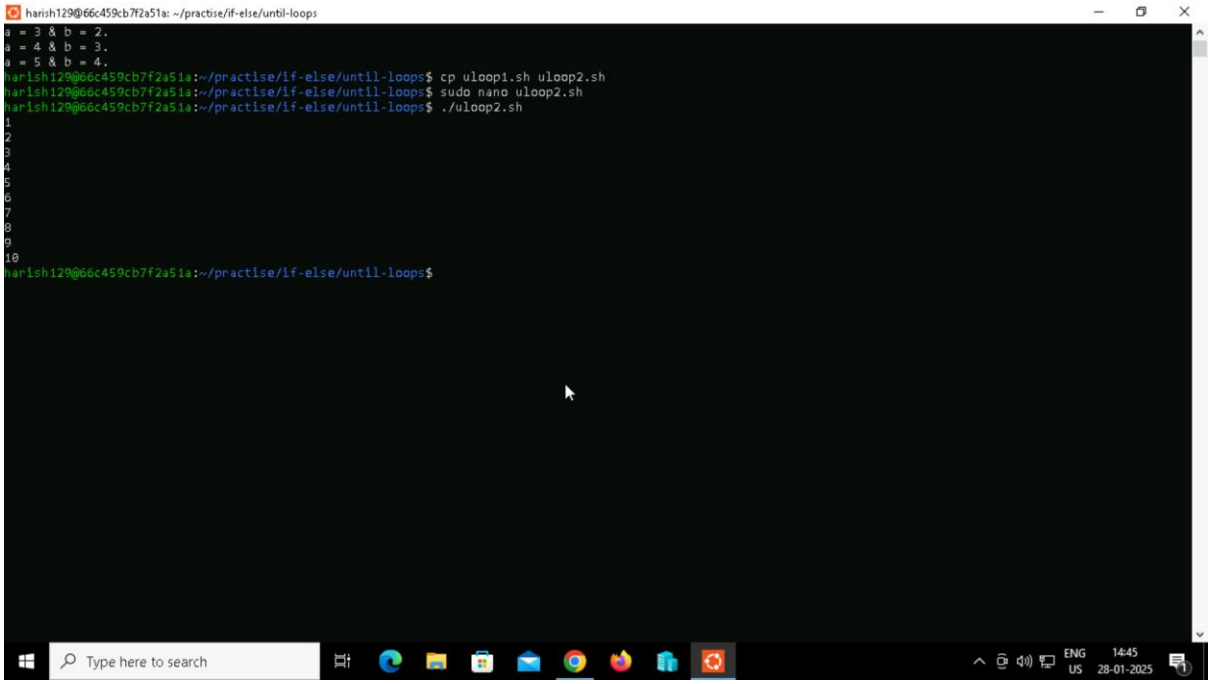
Output



2. Until loop with Single Conditions:

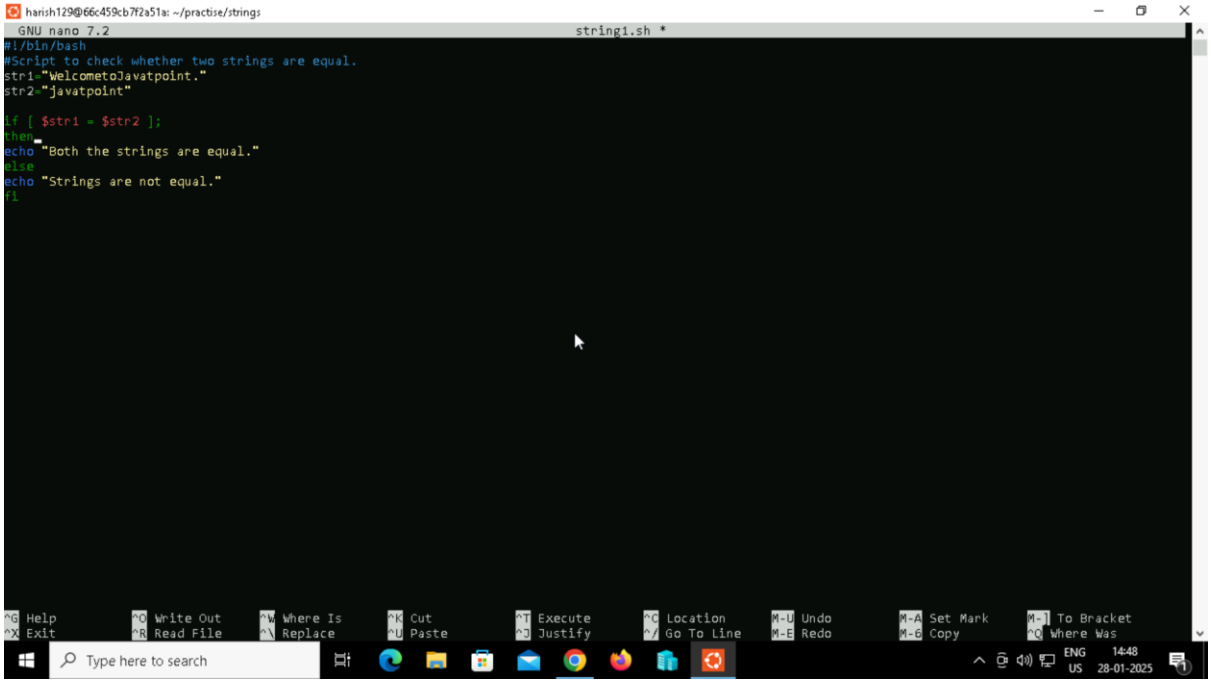


Output:

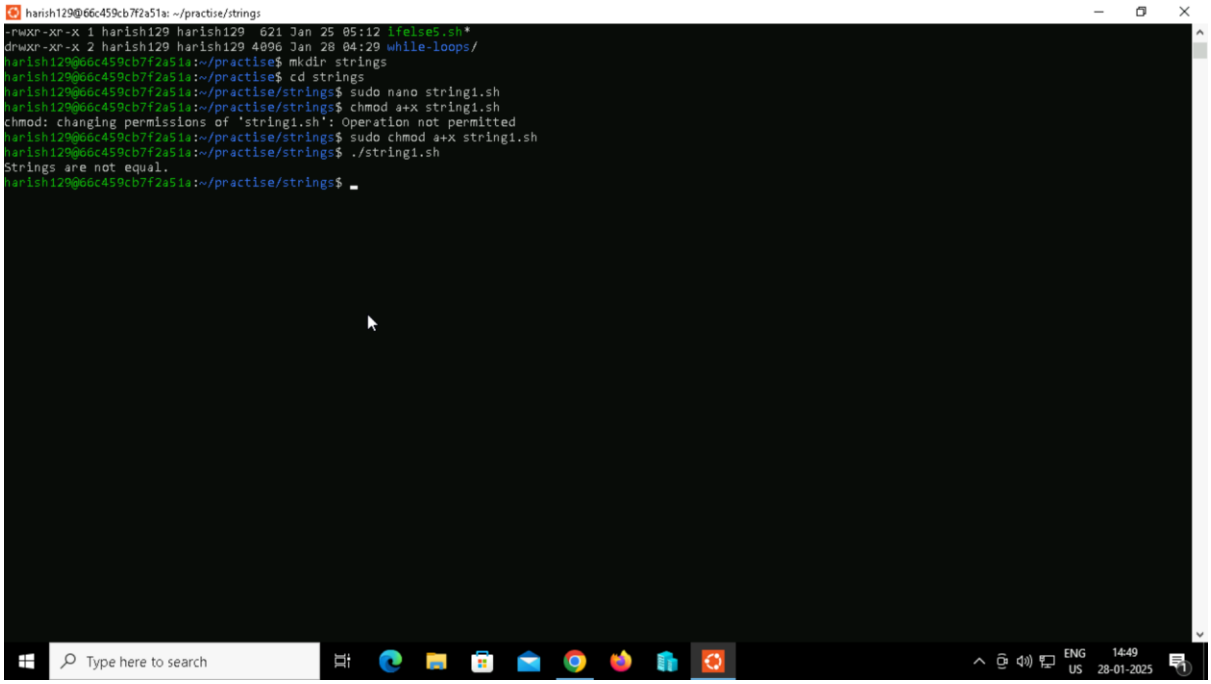


BASH STRINGS:

1. Equal Operator:



Output:



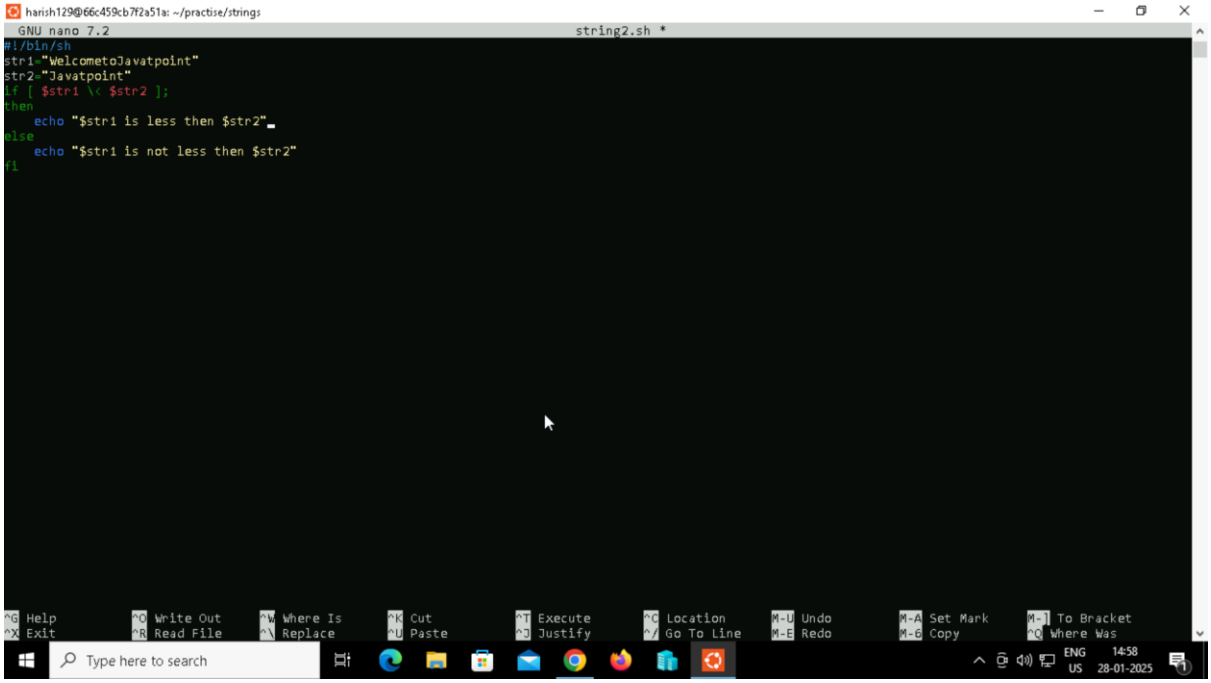
2. Not Equal Operator:

```
Select harish129@66c459cb7f2a51a: ~/practise/strings
  GNU nano 7.2                                   string1.sh *
#!/bin/bash
#Script to check whether two strings are equal.
str1="javatpoint."
str2="javatpoint"

if [ $str1 = $str2 ];
then
echo "Both the strings are equal."
else
echo "Strings are not equal."
fi

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was
```

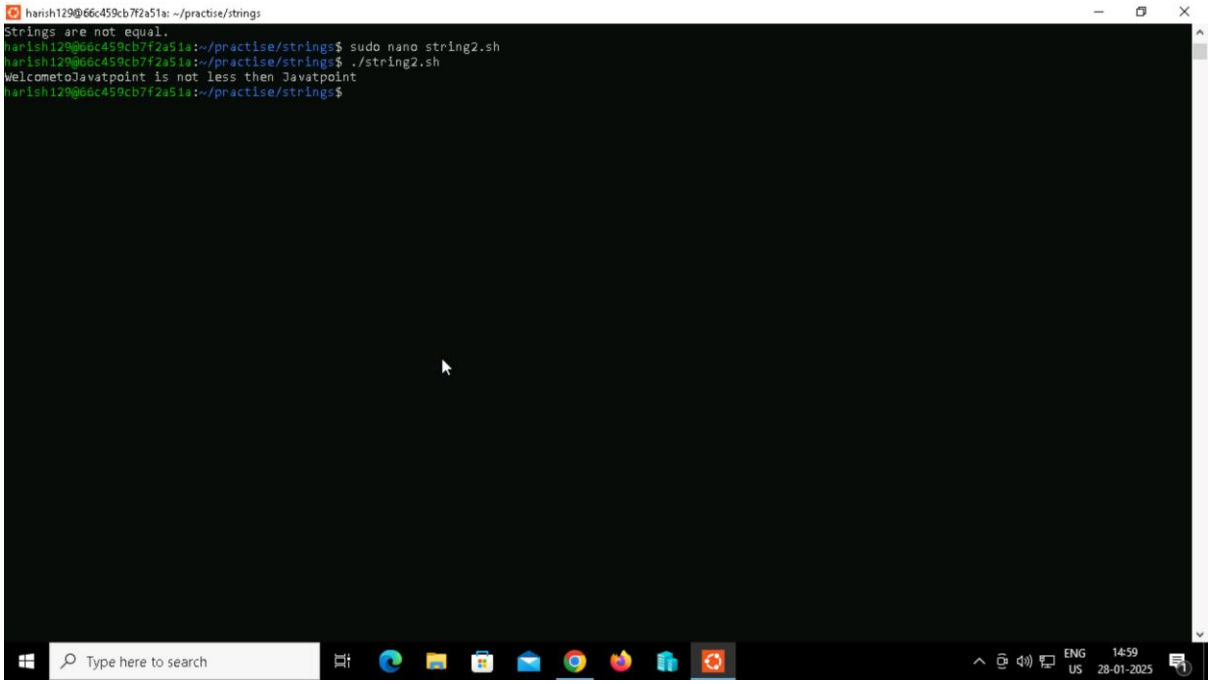
Output:

```
harish129@66c459cb7f2a51a: ~/practise/strings
chmod: changing permissions of 'string1.sh': Operation not permitted
harish129@66c459cb7f2a51a:~/practise/strings$ sudo chmod a+x string1.sh
harish129@66c459cb7f2a51a:~/practise/strings$ ./string1.sh
Strings are not equal.
harish129@66c459cb7f2a51a:~/practise/strings$ cp string1.sh string2.sh
harish129@66c459cb7f2a51a:~/practise/strings$ sudo nano string2.sh
harish129@66c459cb7f2a51a:~/practise/strings$ sudo nano string2.sh
harish129@66c459cb7f2a51a:~/practise/strings$ sudo nano string1.sh
harish129@66c459cb7f2a51a:~/practise/strings$ ./string1.sh
Strings are not equal.
harish129@66c459cb7f2a51a:~/practise/strings$
```

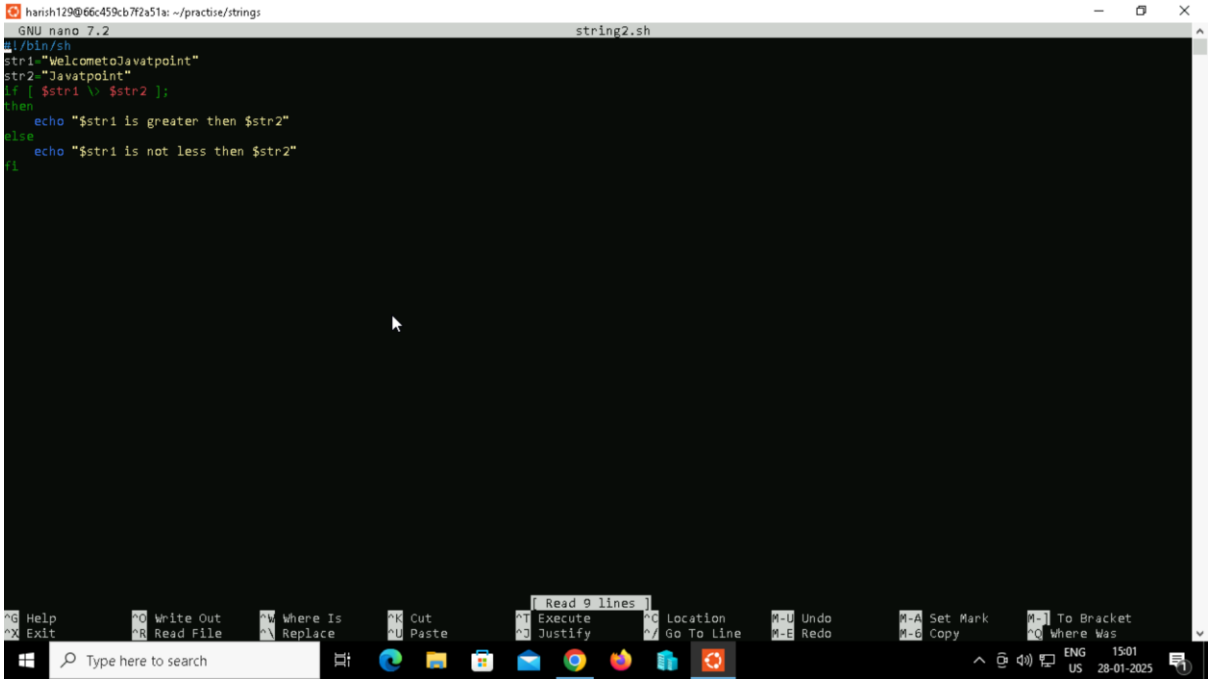
3. Less Than Operator:



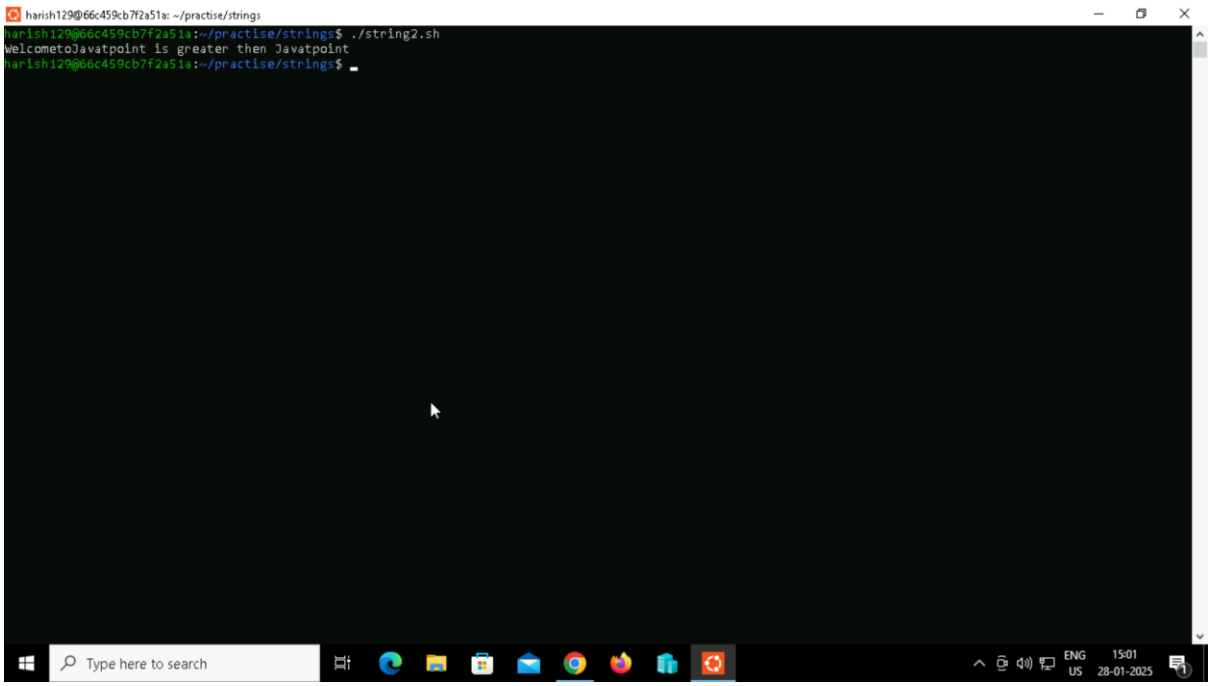
Output:



4. Greater Than Operator:



Output:



5. String Length

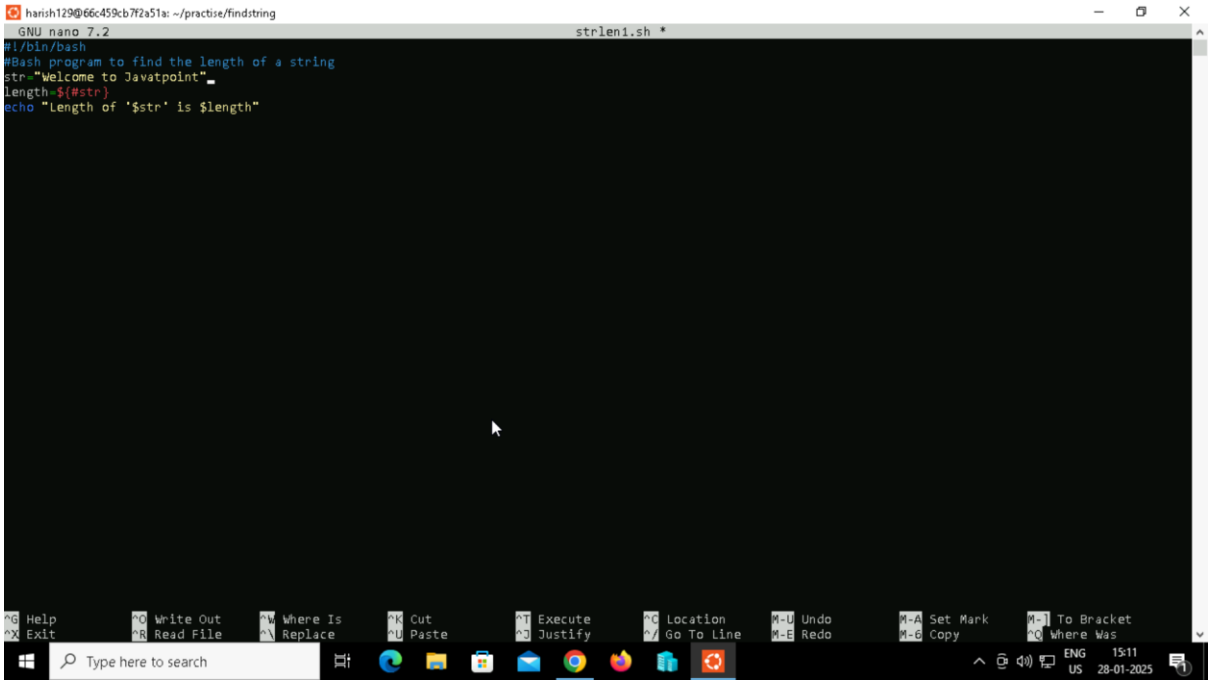
```
harish129@66c459cb7f2a51a: ~/practise/strings
  GNU nano 7.2                                   string3.sh *
#!/bin/sh
str="WelcometoJavatpoint"
if [ -n $str ];
then
    echo "String is not empty"
else
    echo "String is empty"
fi

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was
```

Output:

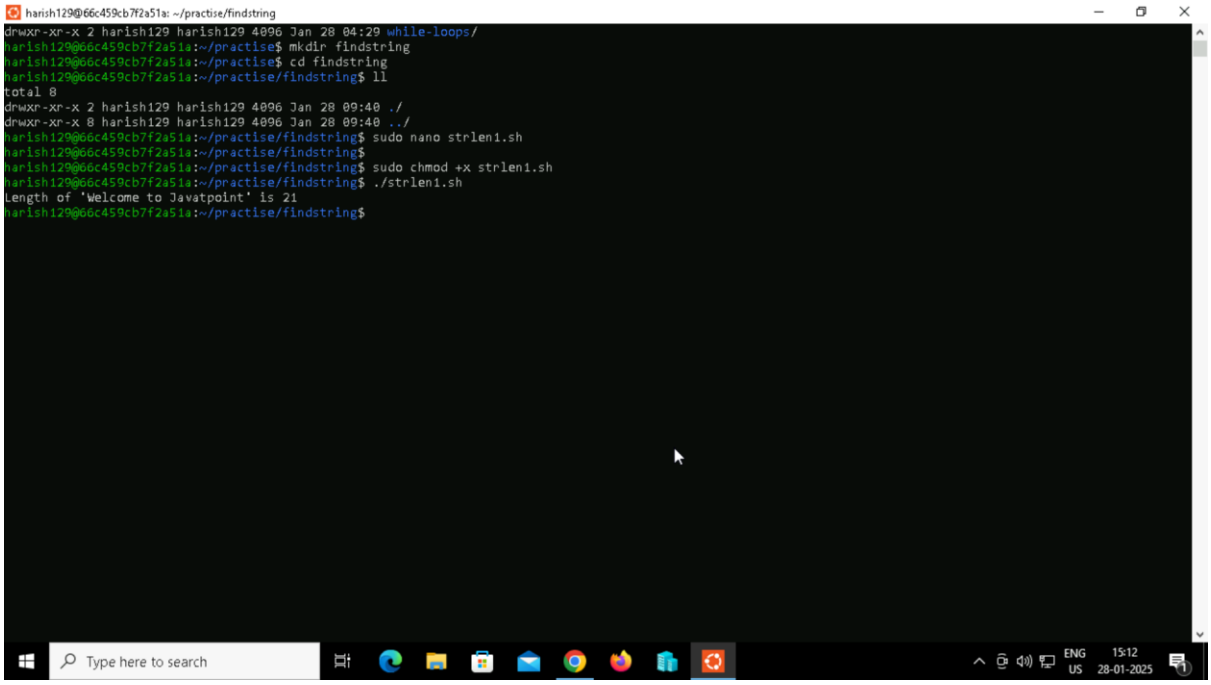
```
harish129@66c459cb7f2a51a: ~/practise/strings
harish129@66c459cb7f2a51a:~/practise/strings$ ./string2.sh
WelcometoJavatpoint is greater then Javatpoint
harish129@66c459cb7f2a51a:~/practise/strings$ sudo nano string2.sh
harish129@66c459cb7f2a51a:~/practise/strings$ sudo nano string3.sh
harish129@66c459cb7f2a51a:~/practise/strings$ sudo chmod +x string3.sh
harish129@66c459cb7f2a51a:~/practise/strings$ ./string3.sh
String is not empty
harish129@66c459cb7f2a51a:~/practise/strings$
```

BASH FIND:

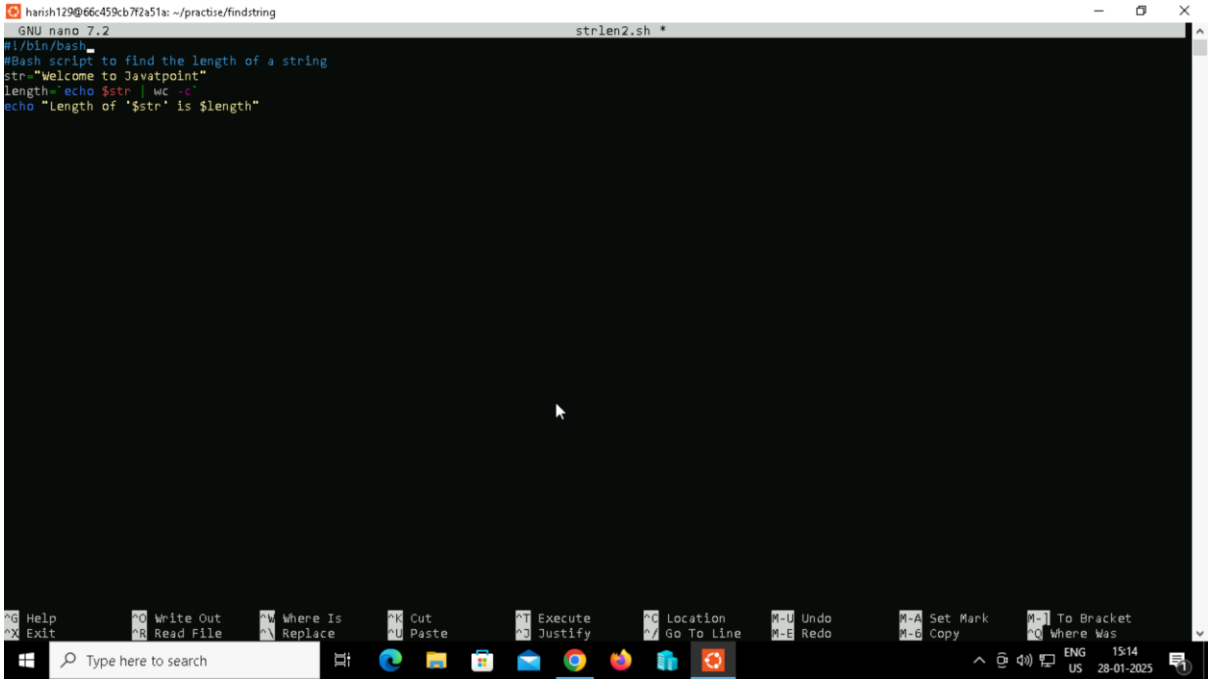
1. Length Of a String:



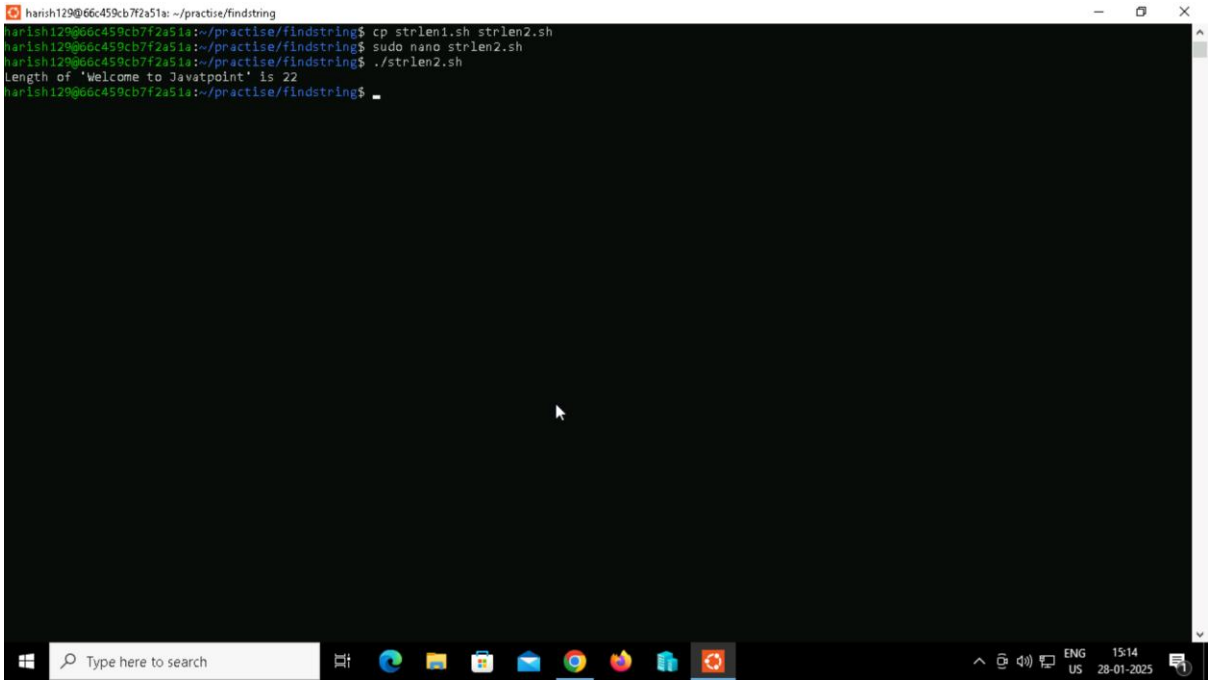
Output:



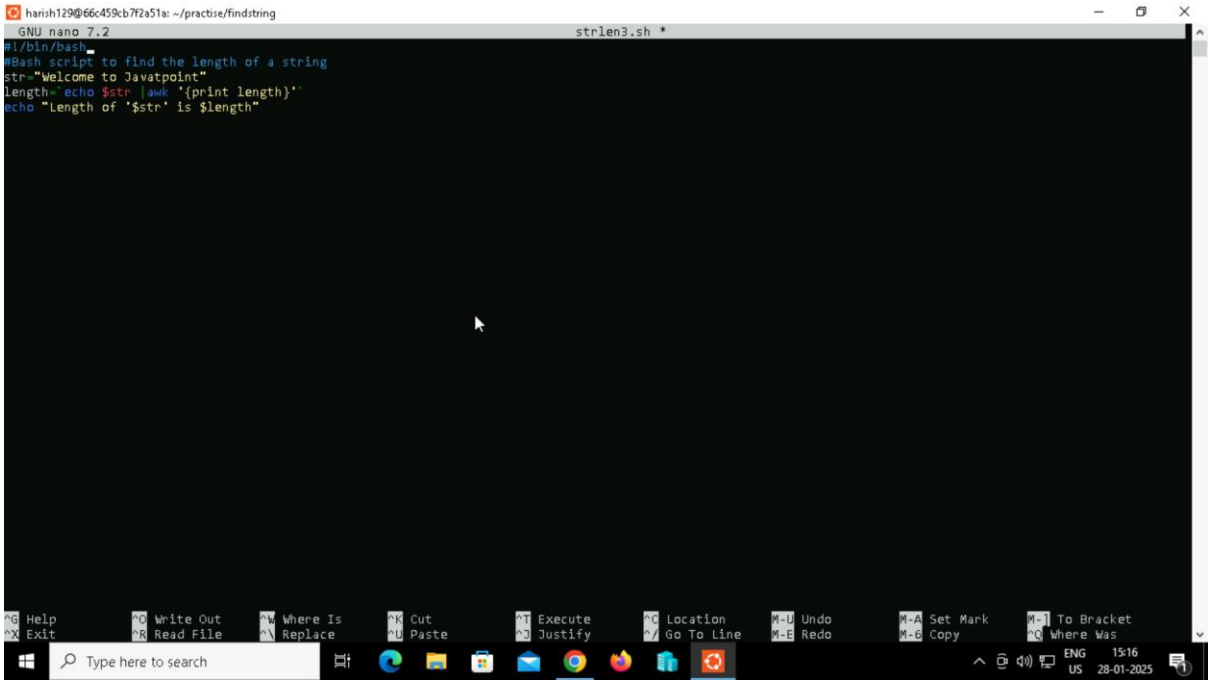
2. Length Of String using “WC” (word count)



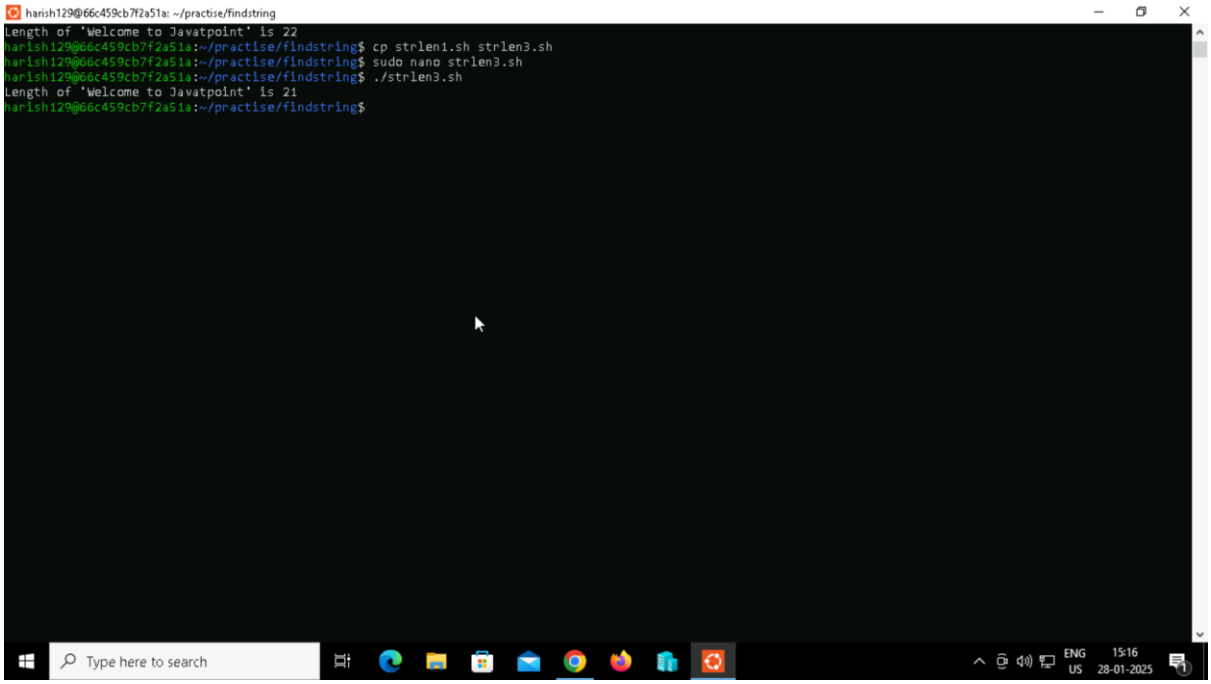
Output:



3. Length Of String using “AWK”

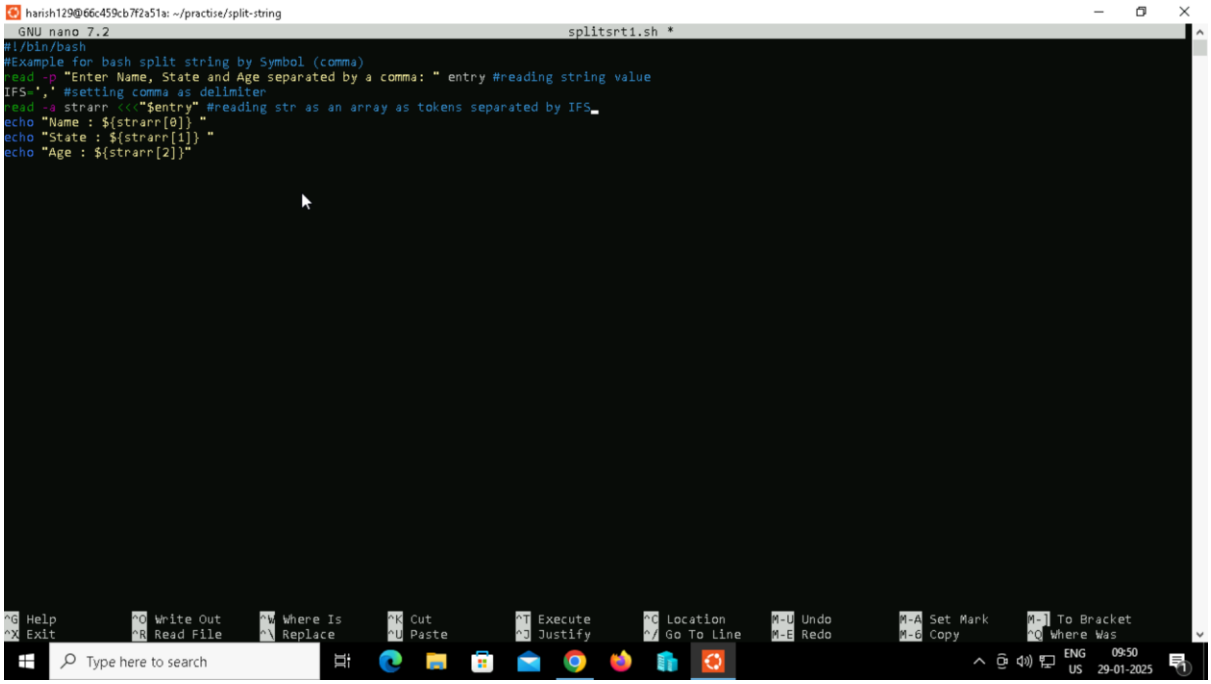


Output:

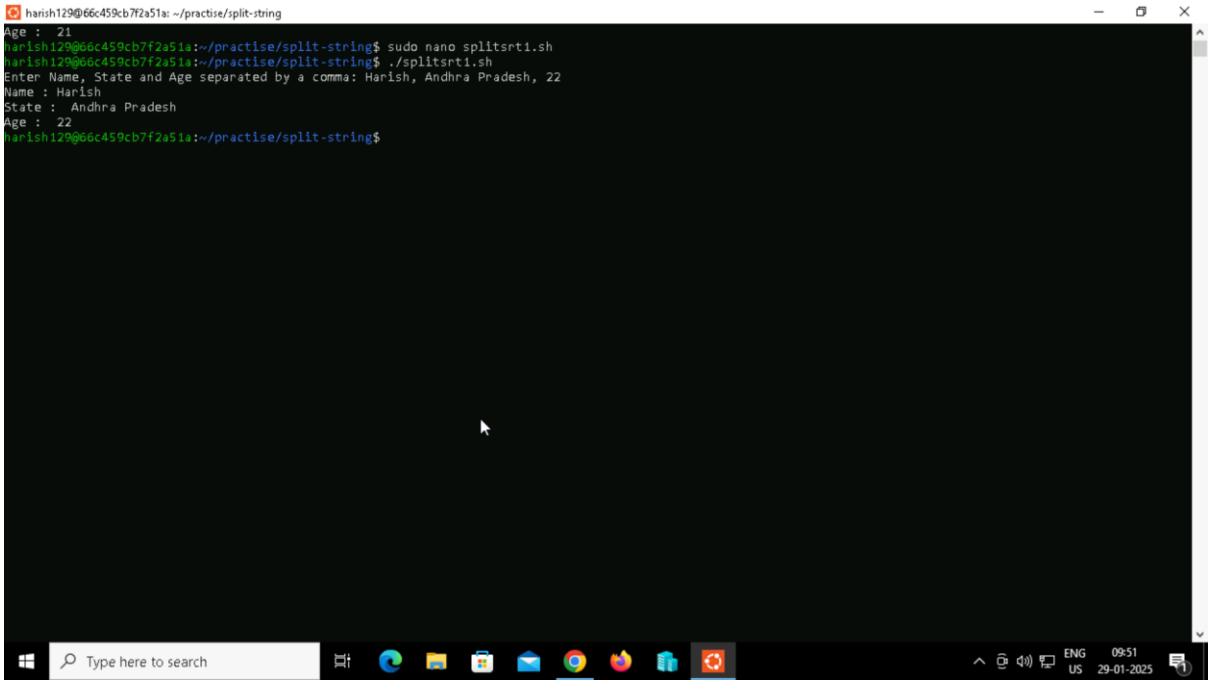


BASH SPLIT STRING:

1. Split String by Symbol



Output:



2. Split String using Trim command:

```
harish129@66c459cb7f2a51a: ~/practise/split-string
  GNU nano 7.2                                   trim.sh *
#!/bin/bash
# Example to split a string using tr command

my_str="We;welcome;you;on;javatpoint."
my_arr=($(echo $my_str | tr ";" "\n"))

for i in "${my_arr[@]}"
do
  echo $i
done

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was
```

Output:

```
harish129@66c459cb7f2a51a: ~/practise/split-string
Two strings must be given when translating.
Try 'tr --help' for more information.
harish129@66c459cb7f2a51a:~/practise/split-string$ sudo nano trim.sh
harish129@66c459cb7f2a51a:~/practise/split-string$
harish129@66c459cb7f2a51a:~/practise/split-string$ ./trim.sh
We
welcome
you
on
javatpoint.
harish129@66c459cb7f2a51a:~/practise/split-string$
```

3.Split String by another string:

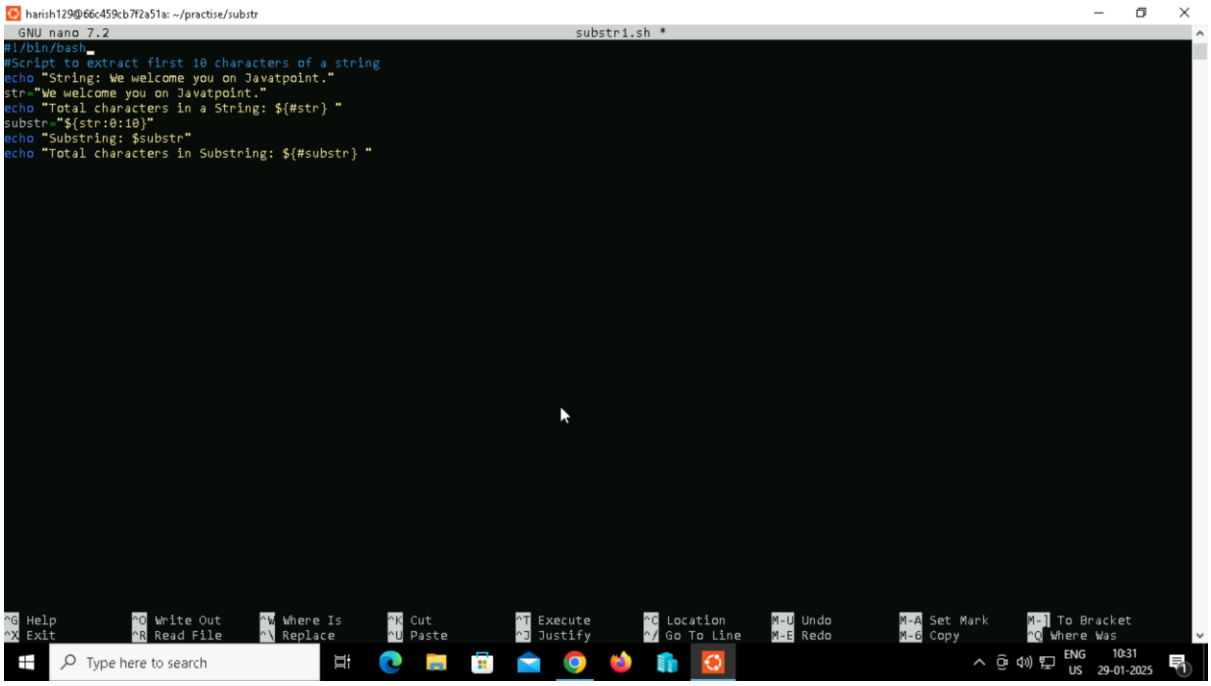
```
harish129@66c459cb7f2a51a: ~/practise/split-string
  GNU nano 7.2                                   splitstr2.sh *
#!/bin/bash
#Example for bash split string by another string
str="WeLearnWelcomeLearnYouLearnOnLearnJavatpoint"
delimiter=Learn
s=$str$delimiter
array=();
while [[ $s ]];
do
array+=( "${s%%"$delimiter"*}" );
s=${s#*"$delimiter"};
done;
declare -p array

^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo      M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line  M-E Redo      M-6 Copy       ^Q Where Was
```

Output:

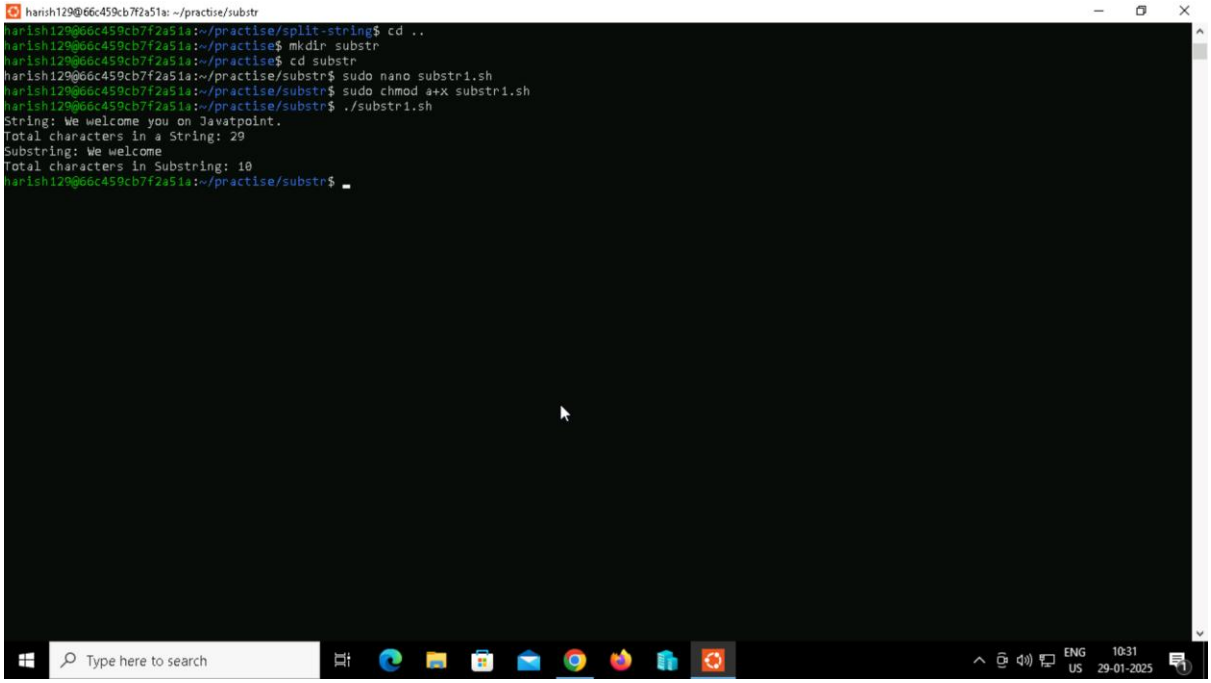
```
harish129@66c459cb7f2a51a: ~/practise/split-string
harish129@66c459cb7f2a51a:~/practise/split-string$ ll
total 20
drwxr-xr-x 2 harish129 harish129 4096 Jan 29 05:13 ./
drwxr-xr-x 9 harish129 harish129 4096 Jan 29 04:10 ../
-rwxr-xr-x 1 root      root       346 Jan 29 04:20 splitstr1.sh*
-rwxr-xr-x 1 harish129 harish129  346 Jan 29 05:13 splitstr2.sh*
-rwxr-xr-x 1 root      root       180 Jan 29 05:10 trim.sh*
harish129@66c459cb7f2a51a:~/practise/split-string$ sudo nano splitstr2.sh
harish129@66c459cb7f2a51a:~/practise/split-string$ ./splitstr2.sh
declare -a array=([0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
harish129@66c459cb7f2a51a:~/practise/split-string$
```

BASH SUB-STRINGS:

1. Substring Till specific characters.



Output:



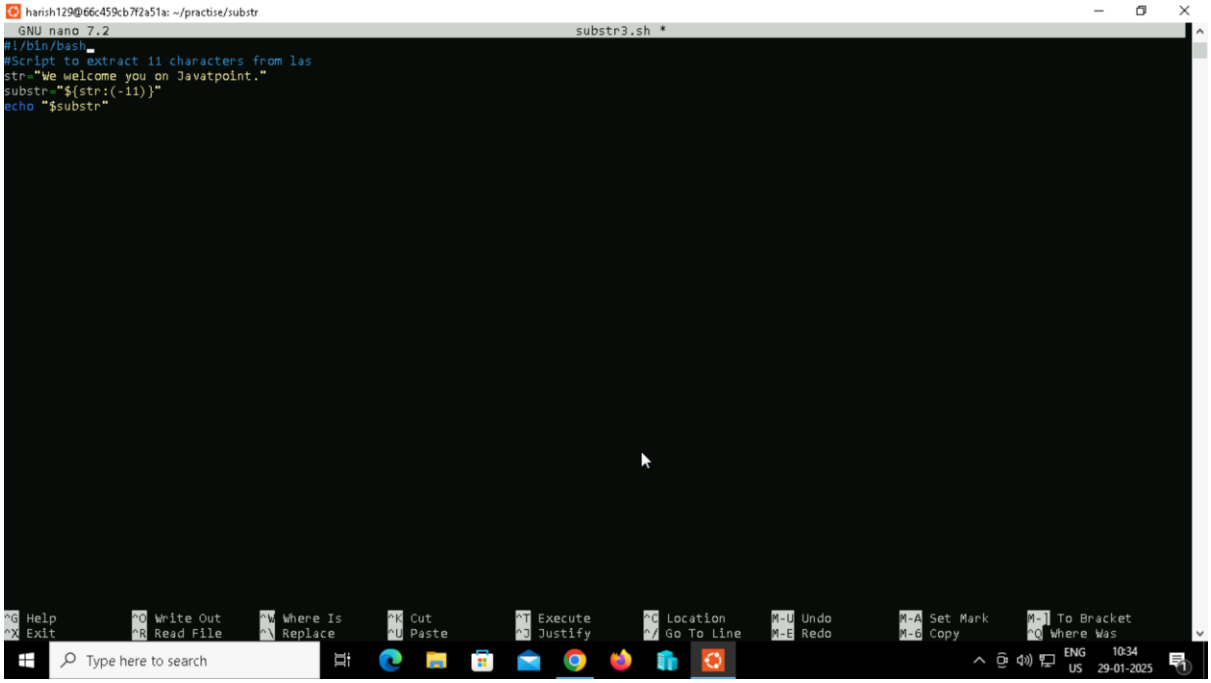
2. Substring from specific character:

```
harish129@66c459cb7f2a51a: ~/practise/substr
  GNU nano 7.2                                substr2.sh *
#!/bin/bash
#Script to print from 11th character onwards
str="We welcome you on Javatpoint."
substr="${str:11}"
echo "$substr"

^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute   ^C Location  M-U Undo      M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy       ^Q Where Was
```

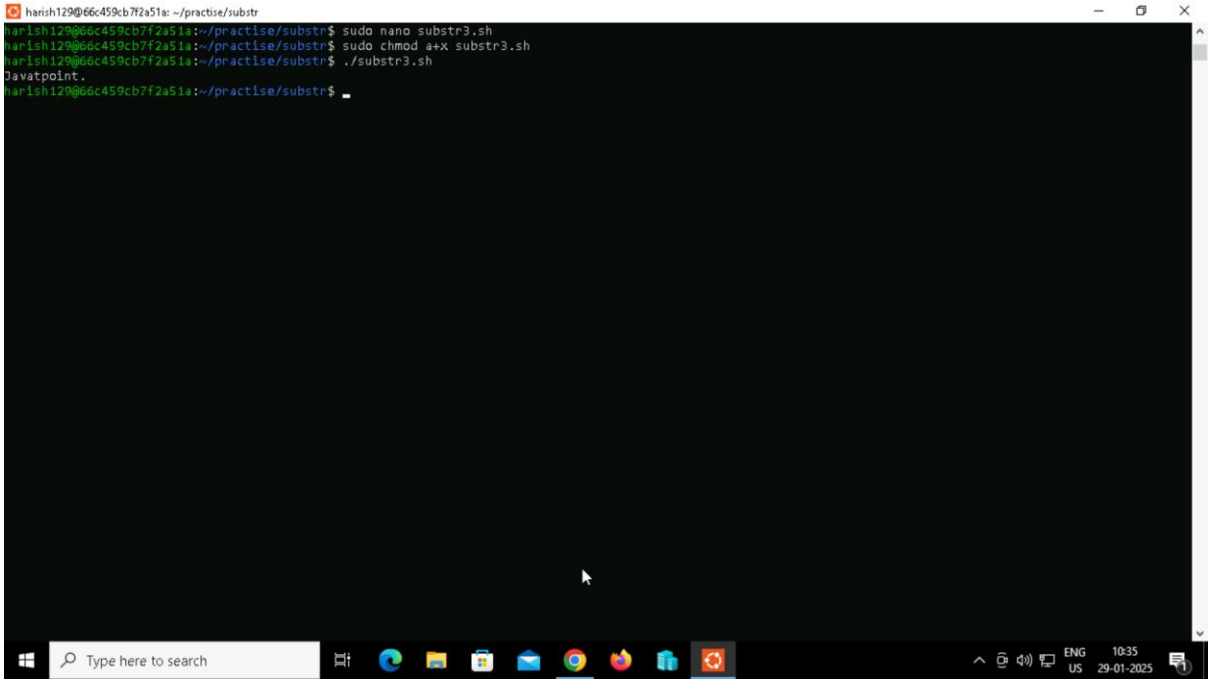
Output:

```
harish129@66c459cb7f2a51a: ~/practise/substr
harish129@66c459cb7f2a51a:~/practise/substr$ sudo nano substr2.sh
harish129@66c459cb7f2a51a:~/practise/substr$ sudo chmod a+x substr2.sh
harish129@66c459cb7f2a51a:~/practise/substr$ ./substr2.sh
you on Javatpoint.
harish129@66c459cb7f2a51a:~/practise/substr$
```

3. specific character from the last:

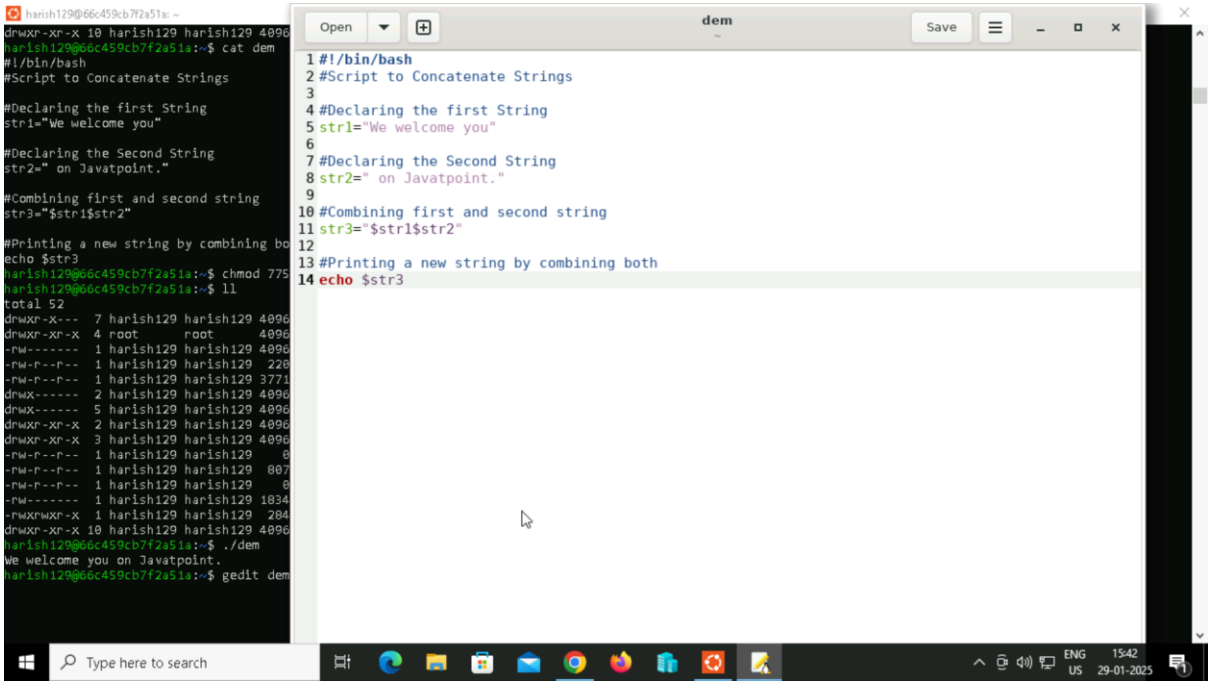


Output:

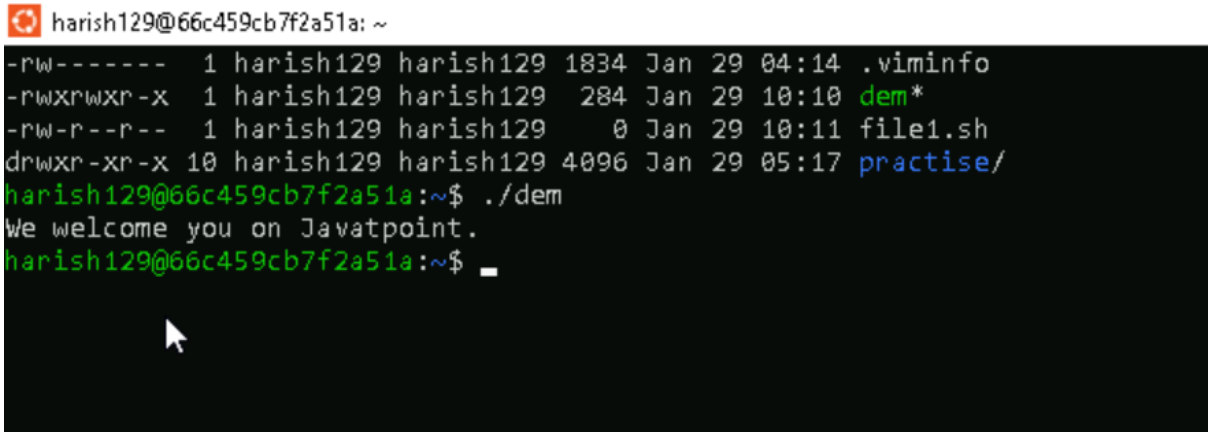


Concatenate Strings :

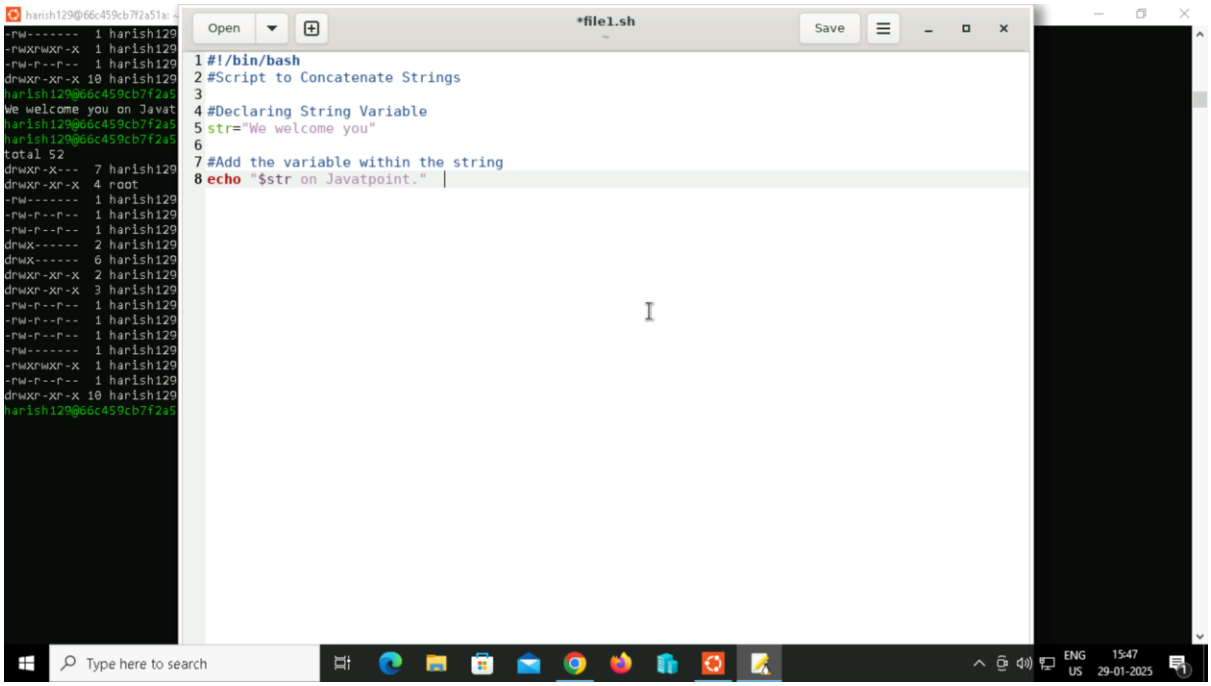
1.



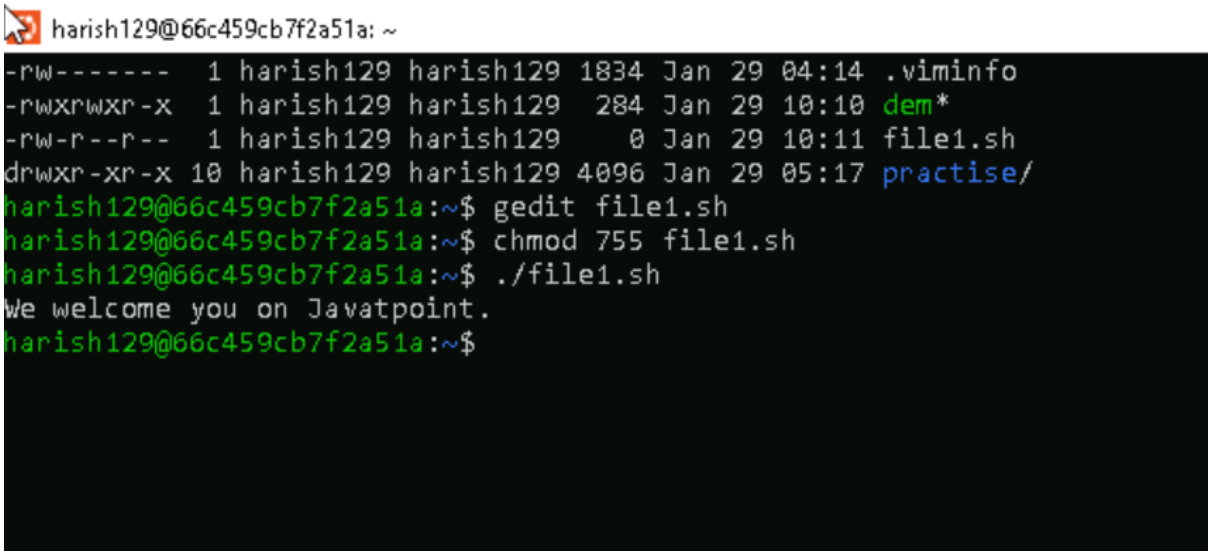
Output:



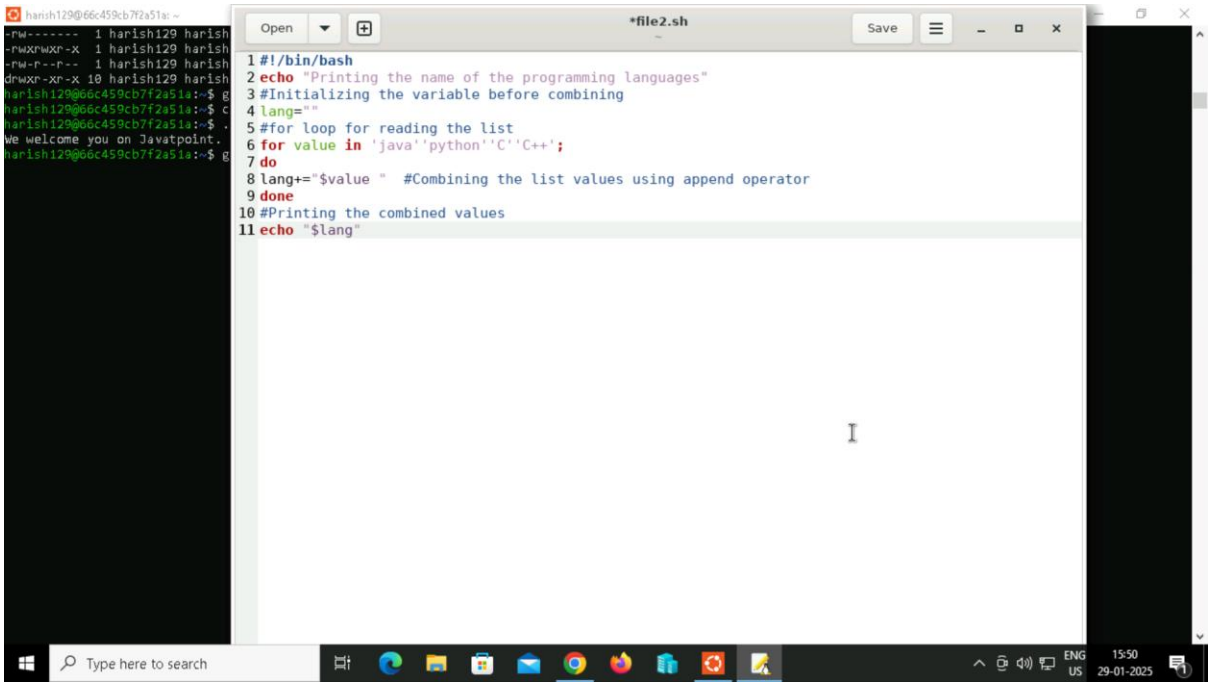
2. Using Double Quotes



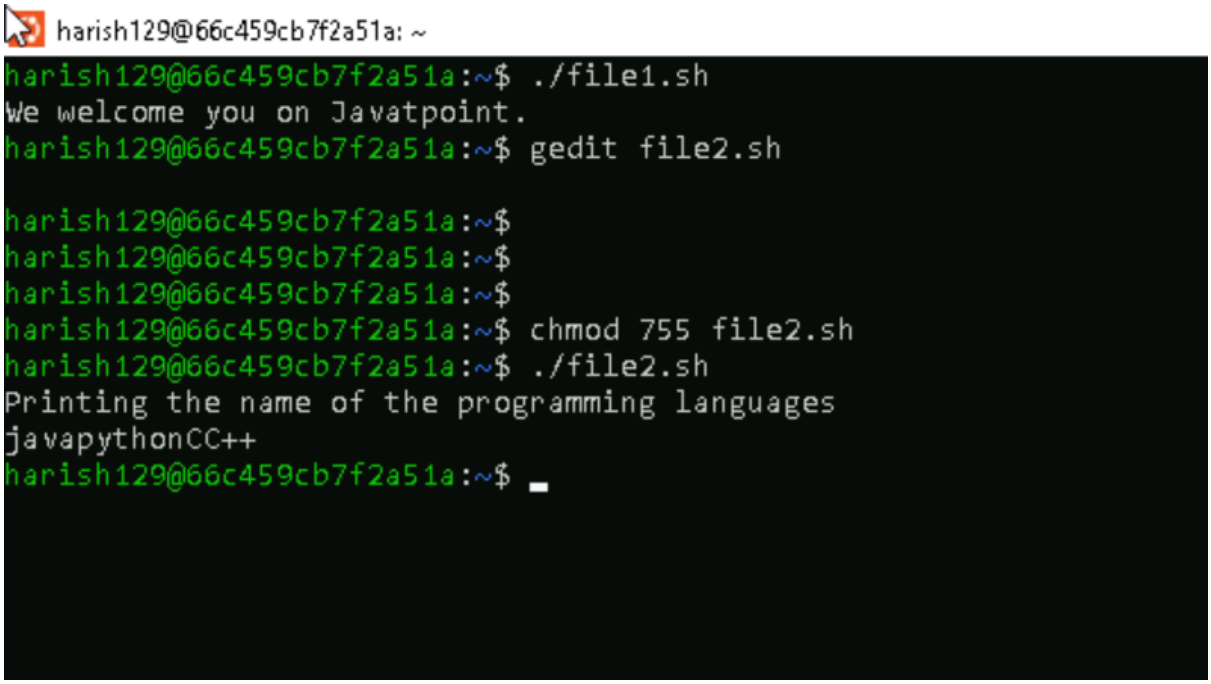
Output:



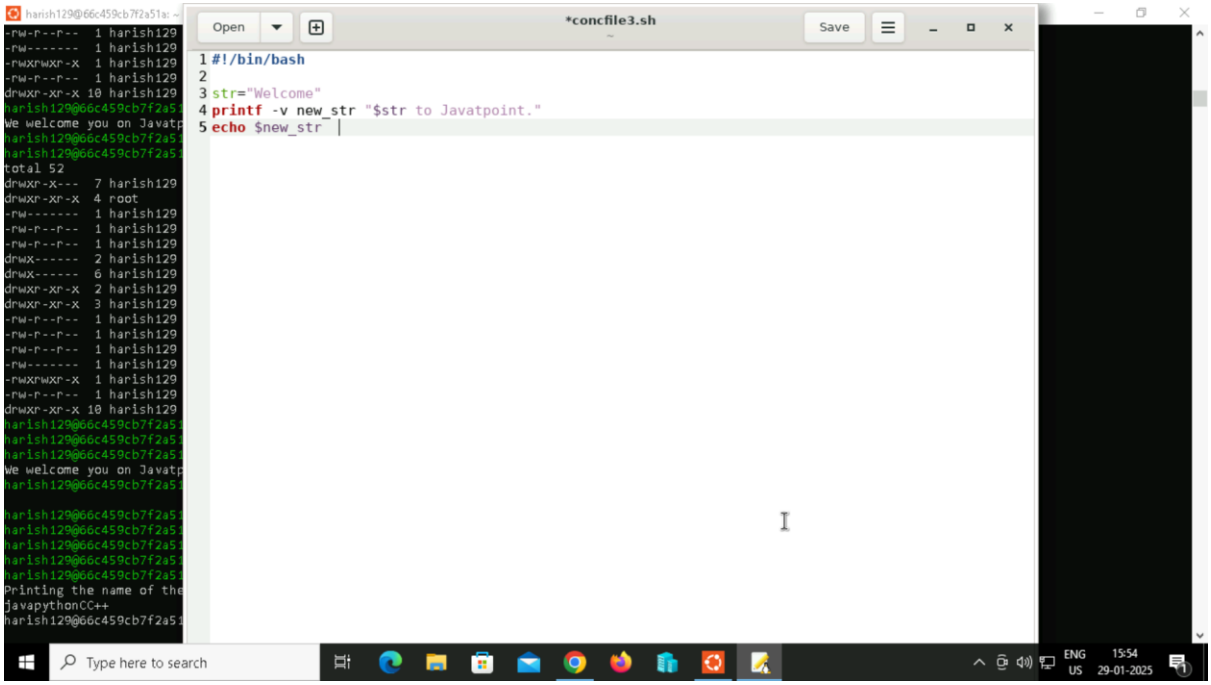
3. using append operator with loop:



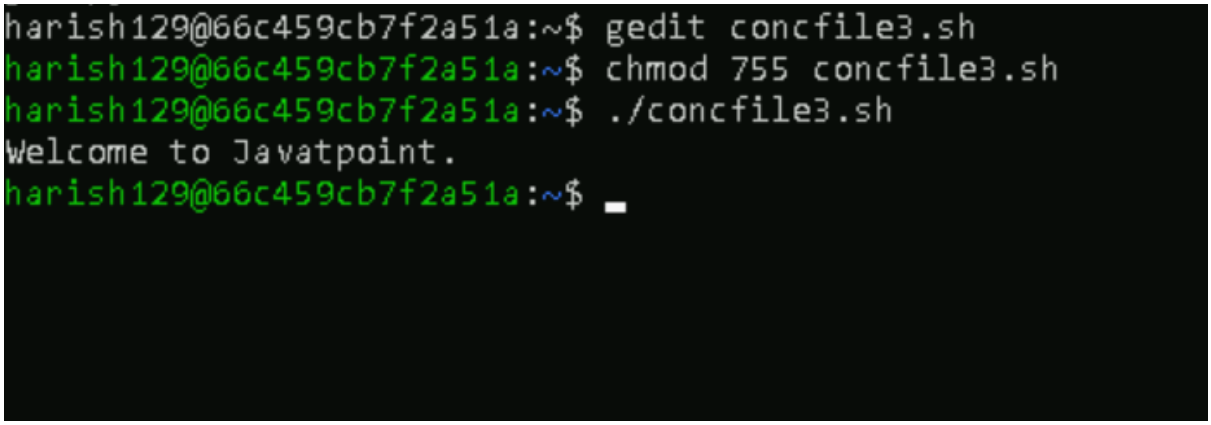
Output:



4. Using Printf function:

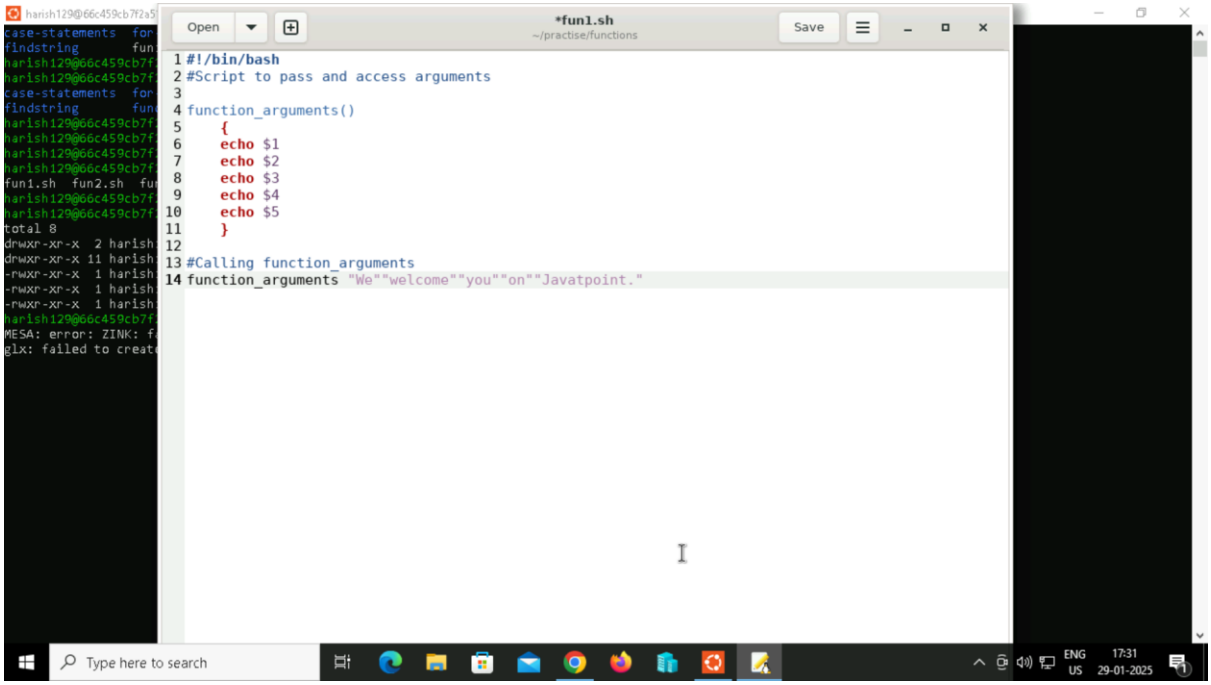


Output:



FUNCTIONS:

1. Passing Arguments:



Output:

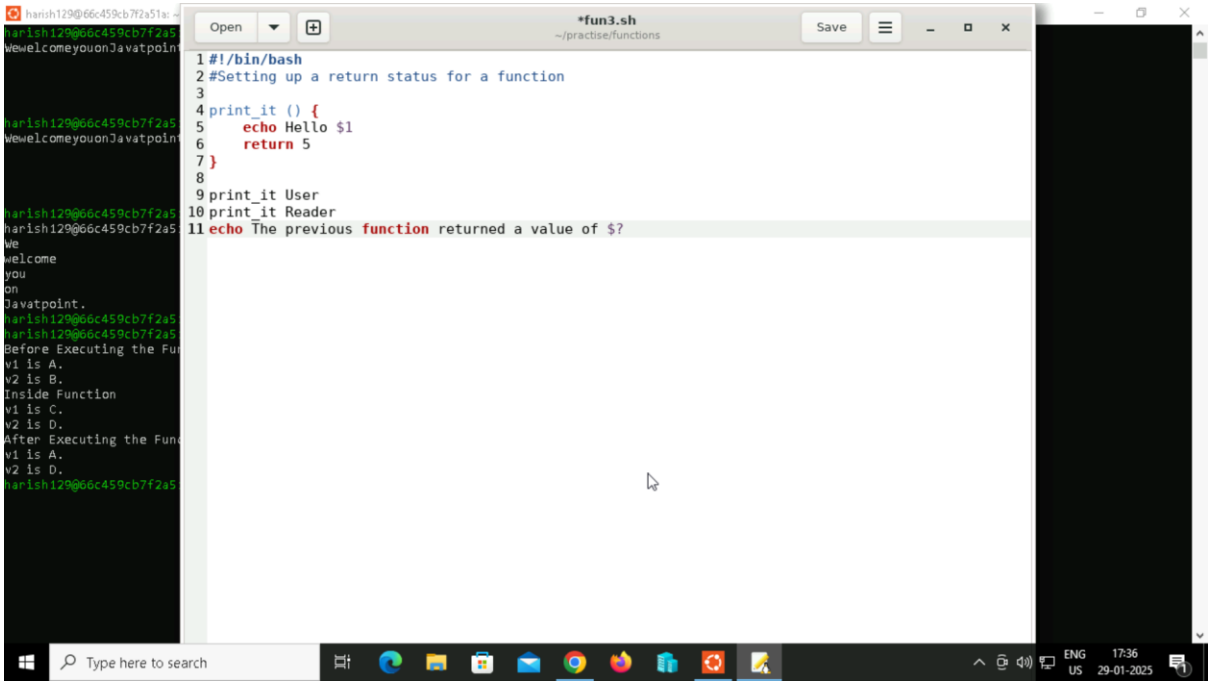
2. Variable Scope:

```
#!/bin/bash

v1='A'
v2='B'

my_var () {
local v1='C'
v2='D'
echo "Inside Function"
echo "v1 is $v1."
echo "v2 is $v2."
}

echo "Before Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."

my_var
echo "After Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
```

Output:

```
harish129@66c459cb7f2a51a:~/practise/functions$ gedit fun2.sh
harish129@66c459cb7f2a51a:~/practise/functions$ ./fun2.sh
Before Executing the Function
v1 is A.
v2 is B.
Inside Function
v1 is C.
v2 is D.
After Executing the Function
v1 is A.
v2 is D.
harish129@66c459cb7f2a51a:~/practise/functions$
```

3. Return Values:



Output:

```
harish129@66c459cb7f2a51a:~/practise/functions$ gedit fun3.sh
harish129@66c459cb7f2a51a:~/practise/functions$ ./fun3.sh
Hello User
Hello Reader
The previous function returned a value of 5
harish129@66c459cb7f2a51a:~/practise/functions$
```

ARRAYS:

1. Print elements using indexes:

```
#!/bin/bash
#Script to print an element of an array with an index of 2

#declaring the array
declare -a example_array=( "Welcome""To""Javatpoint" )

#printing the element with index of 2
echo ${example_array[2]}
```

Output:

```
harish129@66c459cb7f2a51a:~/practise/arrays$ gedit arr1.sh
harish129@66c459cb7f2a51a:~/practise/arrays$ ./arr1.sh
Javatpoint
harish129@66c459cb7f2a51a:~/practise/arrays$
```

EX-2:

```
#!/bin/bash
# Script to print all the elements of the array

# Declaring the array with separate elements
declare -a example_array=( "Welcome" "To" "Javatpoint" )

# Printing all elements
echo "${example_array[@]}"
```

```
harish129@66c459cb7f2a51a:~/practise/arrays$ gedit arr1.sh
harish129@66c459cb7f2a51a:~/practise/arrays$ ./arr1.sh
Welcome To Javatpoint
harish129@66c459cb7f2a51a:~/practise/arrays$
```

2. Printing Indexes:

```
#!/bin/bash
# Script to print the keys (indexes) of the array

# Declaring the array with separate elements
declare -a example_array=( "Welcome" "To" "Javatpoint" )

# Printing the keys (indexes)
echo "${!example_array[@]}"
```

Output:

```
harish129@66c459cb7f2a51a:~/practise/arrays$ gedit arr2.sh
harish129@66c459cb7f2a51a:~/practise/arrays$ ./arr2.sh
0 1 2
harish129@66c459cb7f2a51a:~/practise/arrays$
```

3. Array Length:

```
#!/bin/bash

#Declaring the Array
declare -a example_array=( "Welcome""To""Javatpoint" )

#Printing Array Length
echo "The array contains ${#example_array[@]} elements"
```

Output:

```
harish129@66c459cb7f2a51a:~/practise/arrays$ gedit arr3.sh
harish129@66c459cb7f2a51a:~/practise/arrays$ ./arr3.sh
The array contains 1 elements
harish129@66c459cb7f2a51a:~/practise/arrays$
```