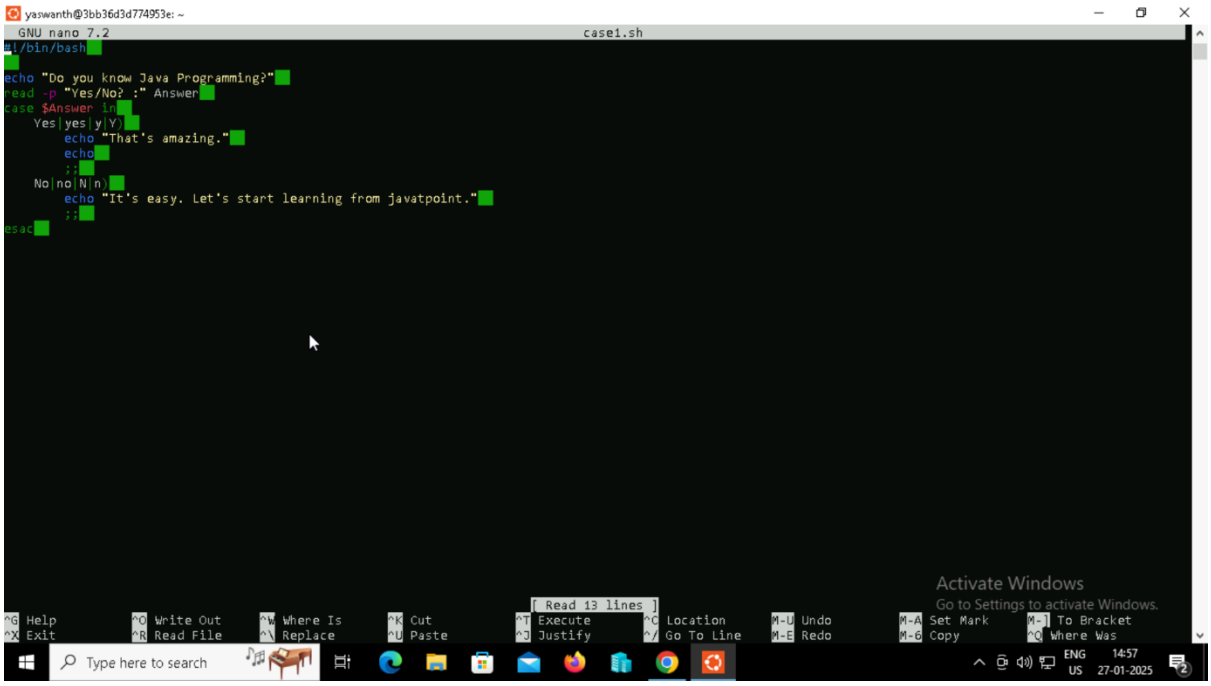
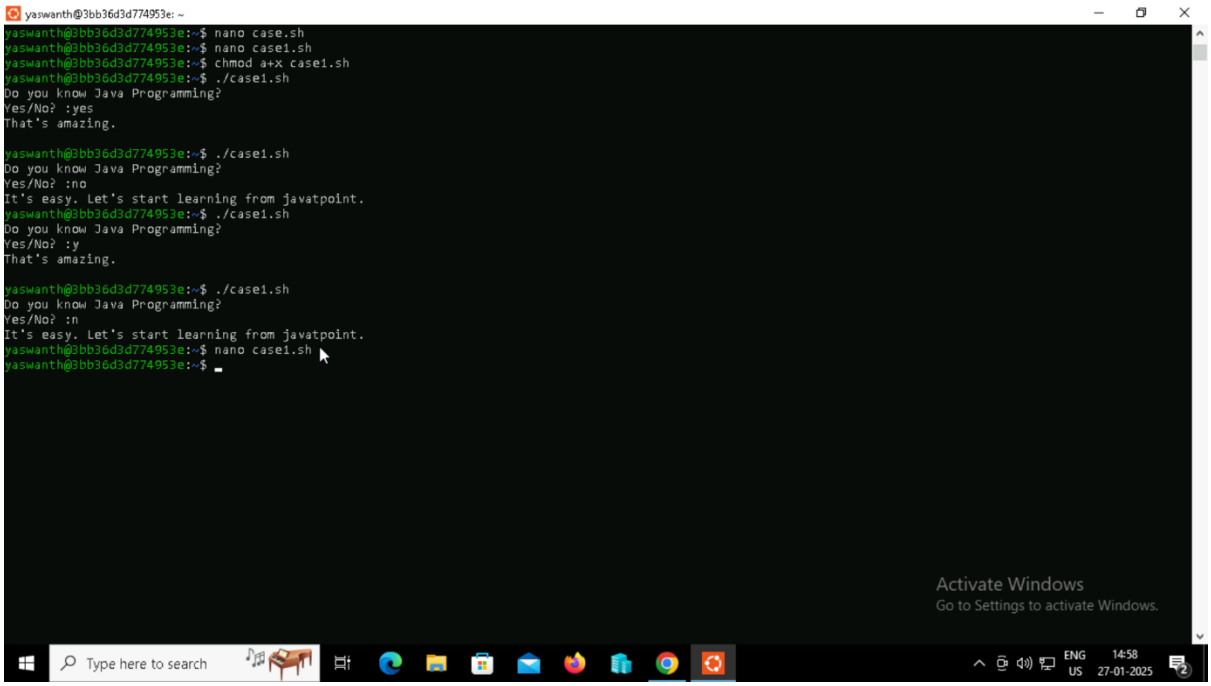
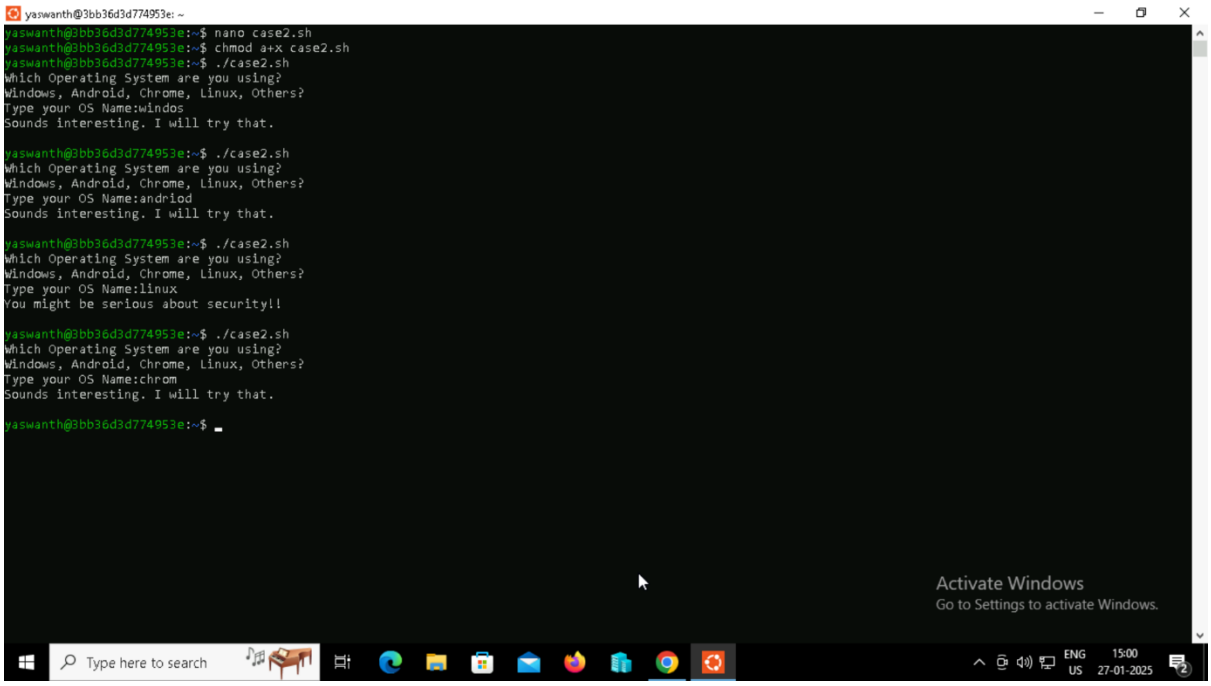
BASH PROGRAM CASE AND FOR LOOPS:

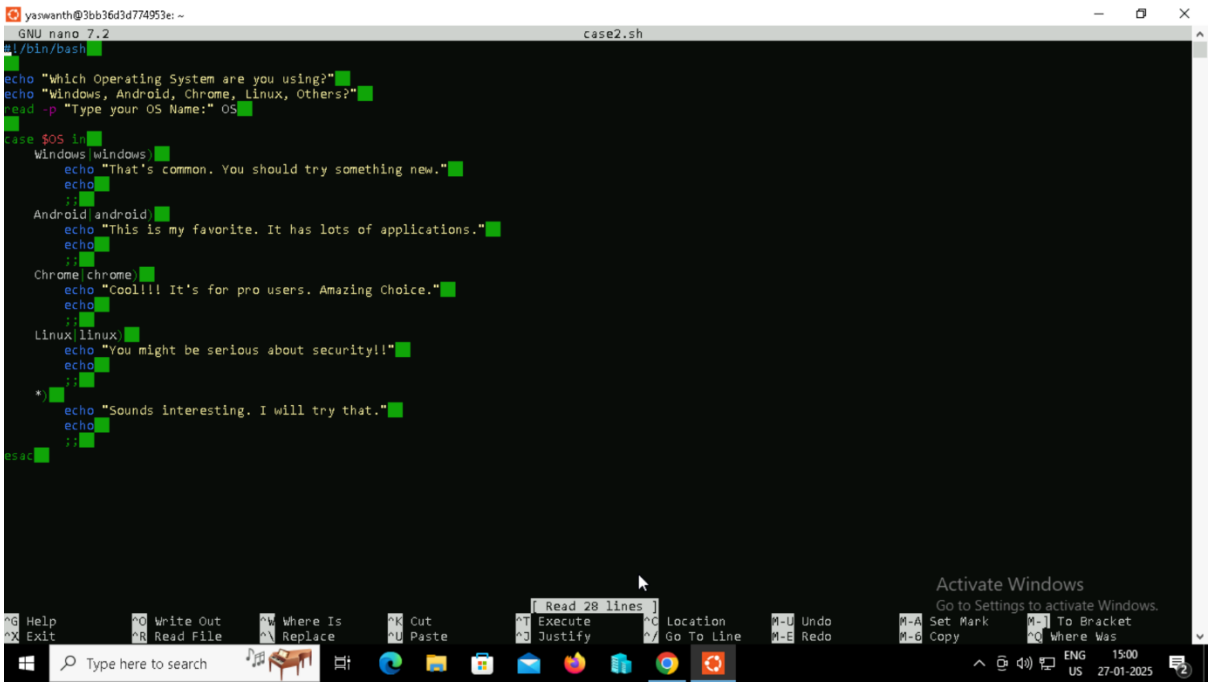
1. In this example, we have defined a simple scenario to demonstrate the use of the Case statement.



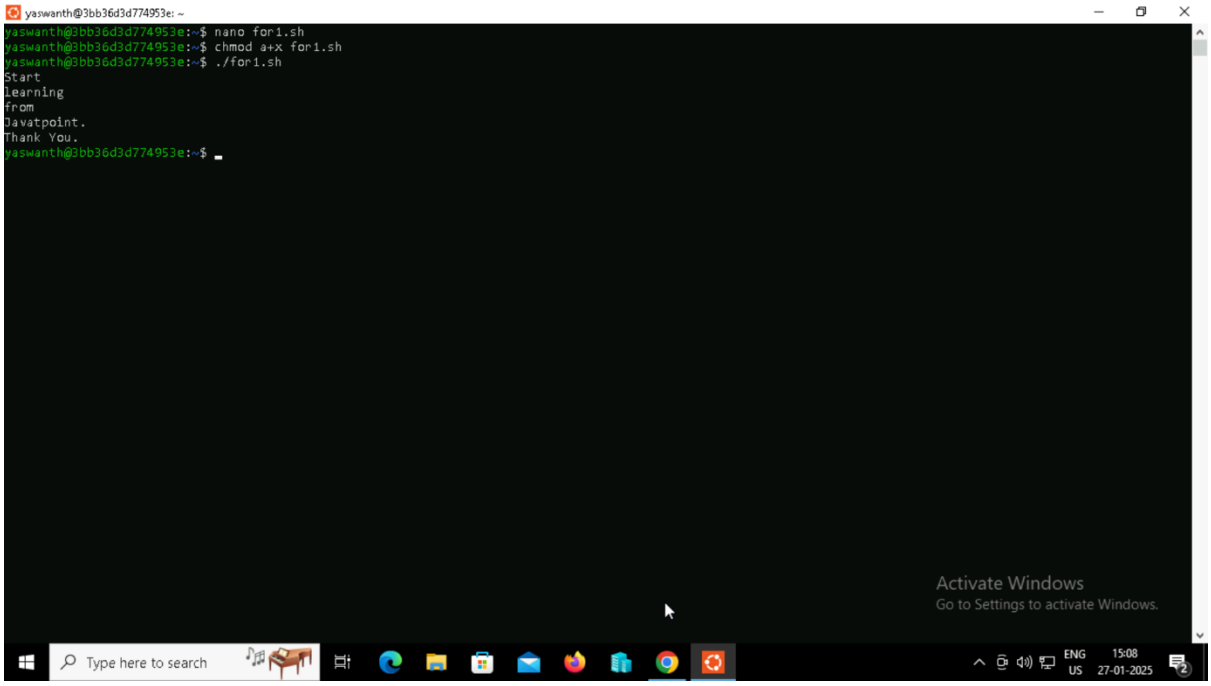


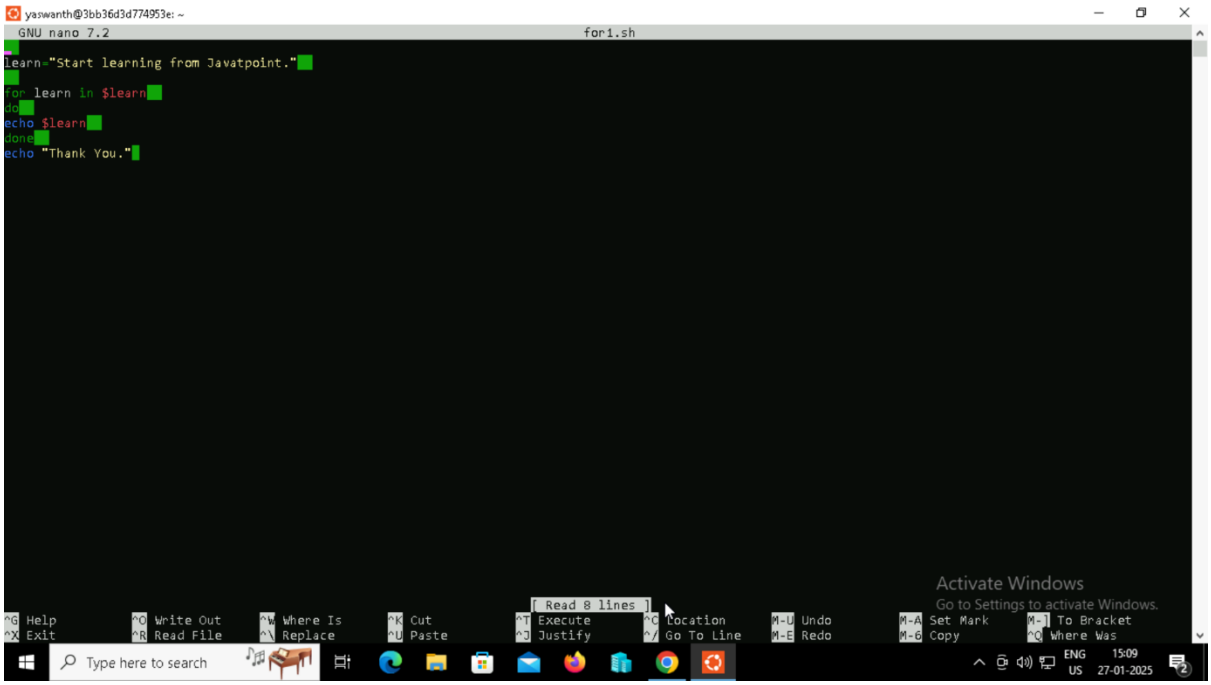
2.In this example, we have defined a combined scenario where there is also a default case when no previous matched case is found**.**



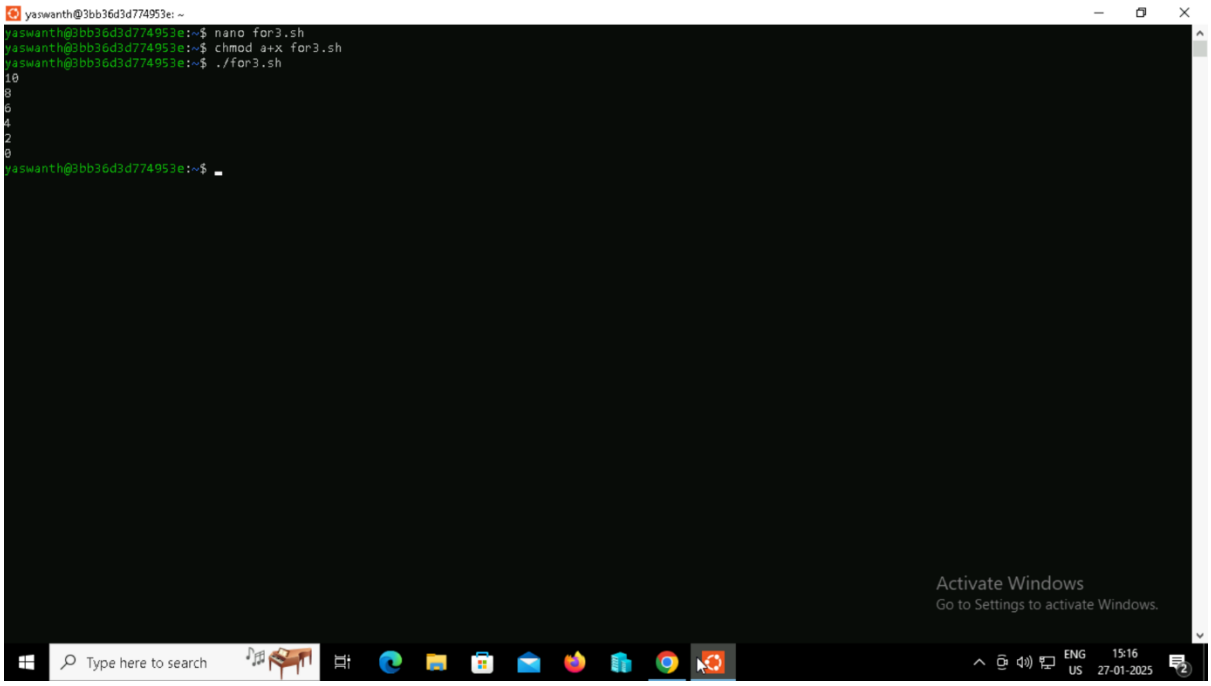


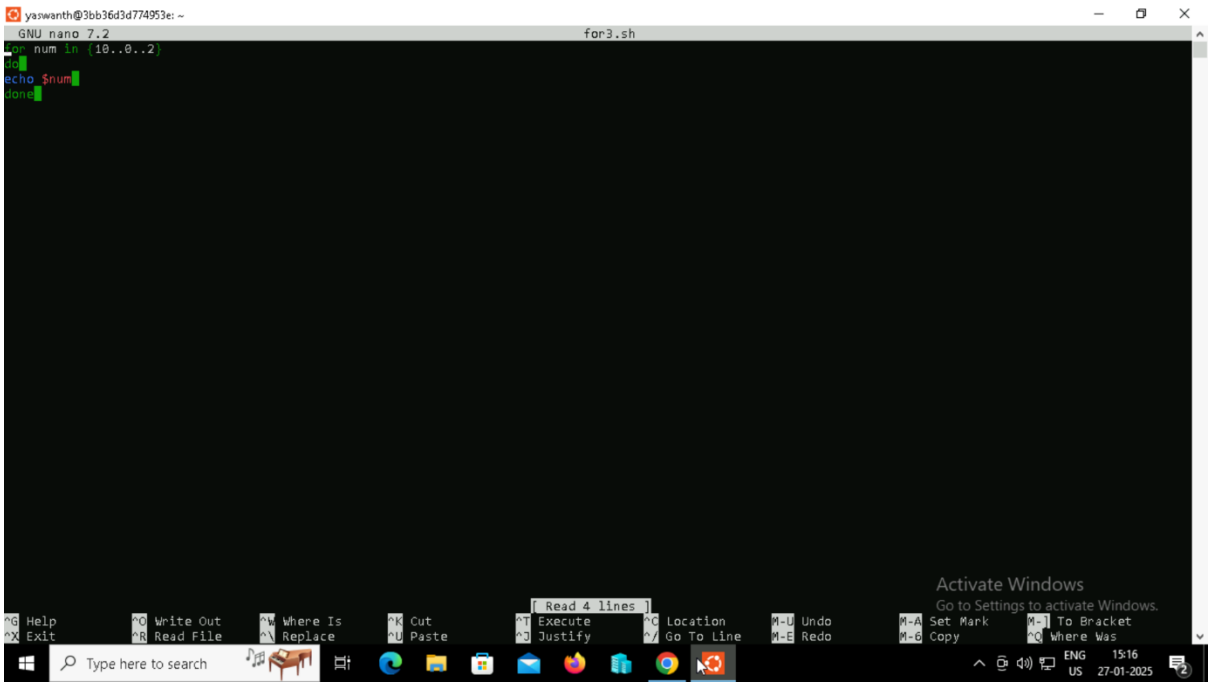
3. Basic 'For Loop' Example in bash



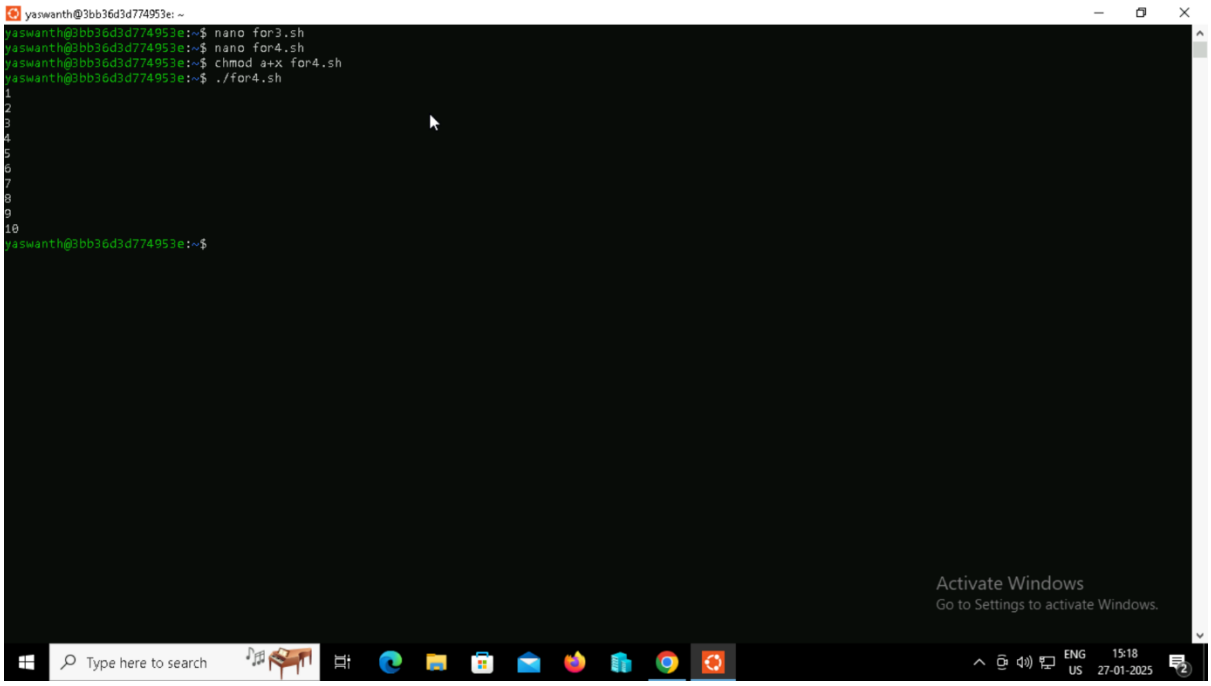


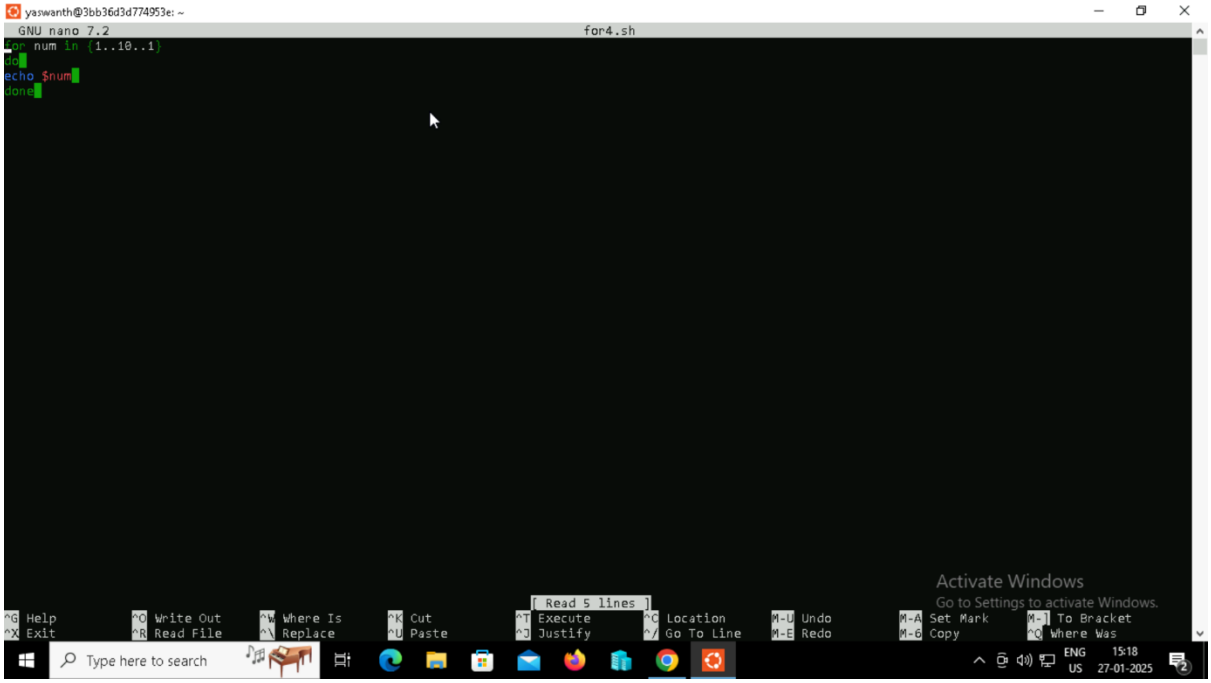
4. For Loop to Read a Range with Increment/Decrement:



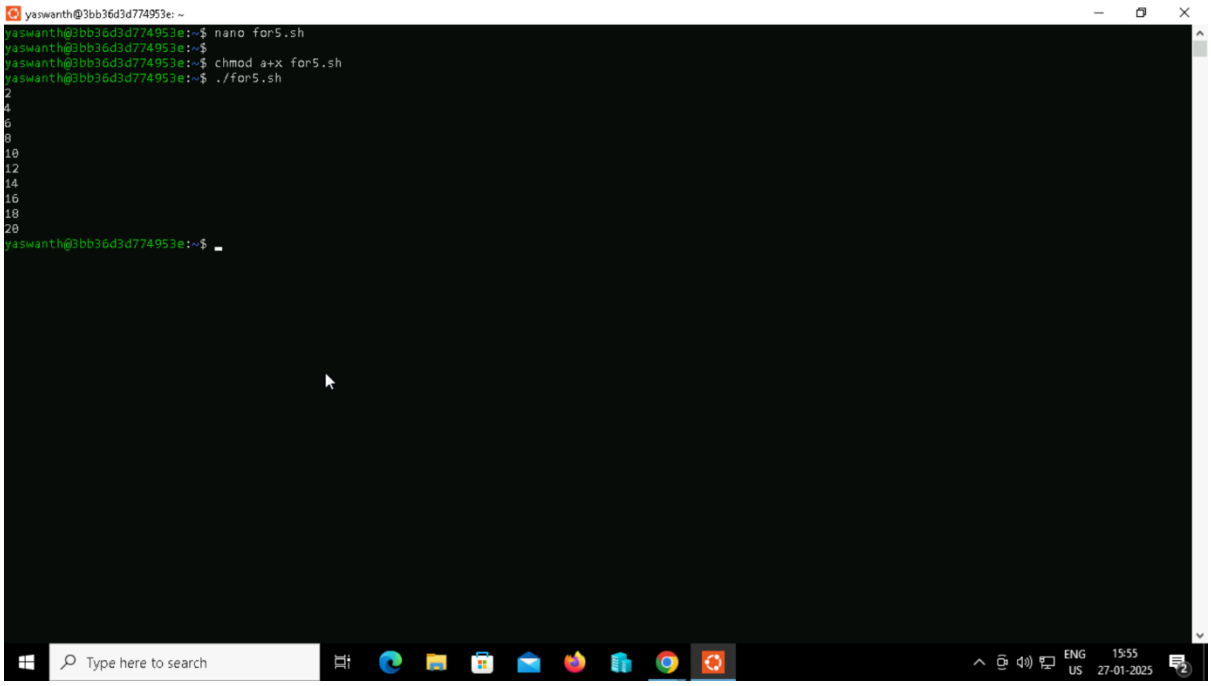


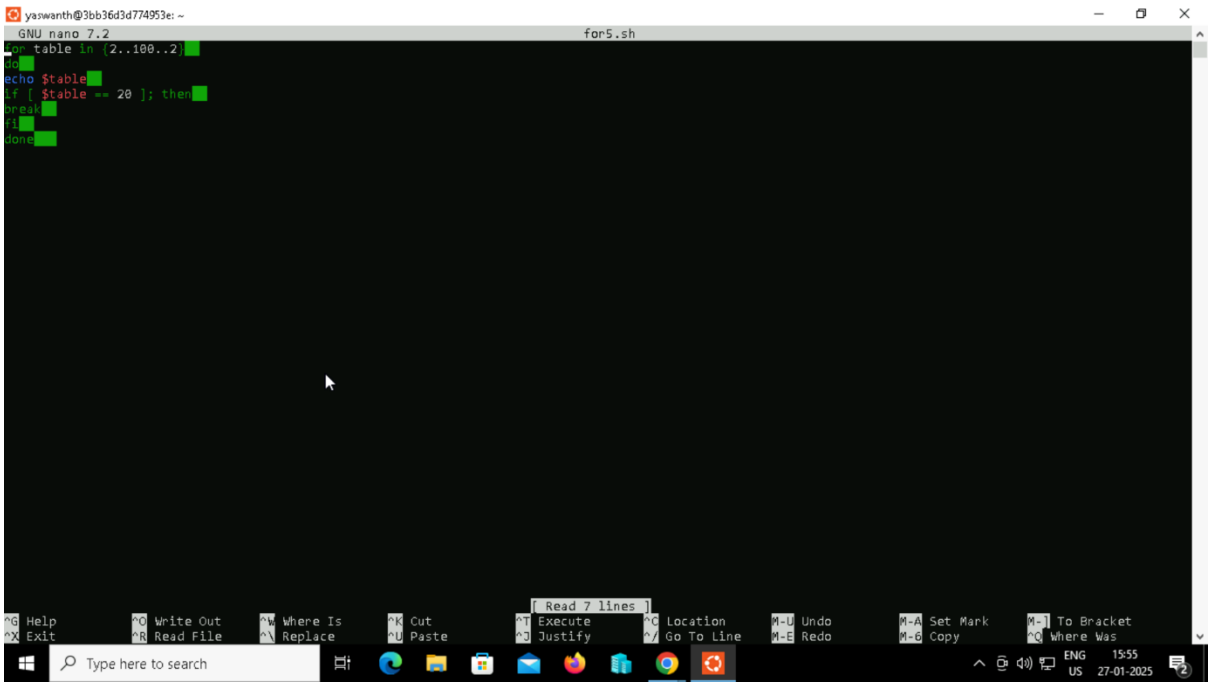
5. This is the basic example to print a series of numbers from 1 to 10.



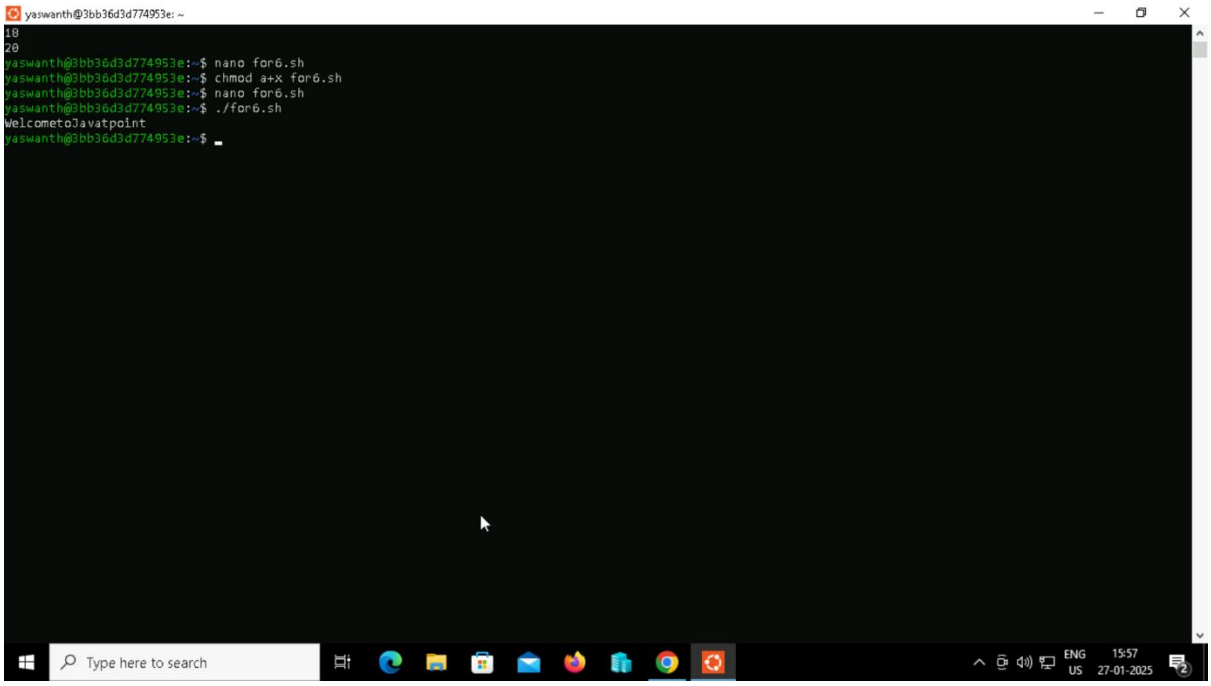


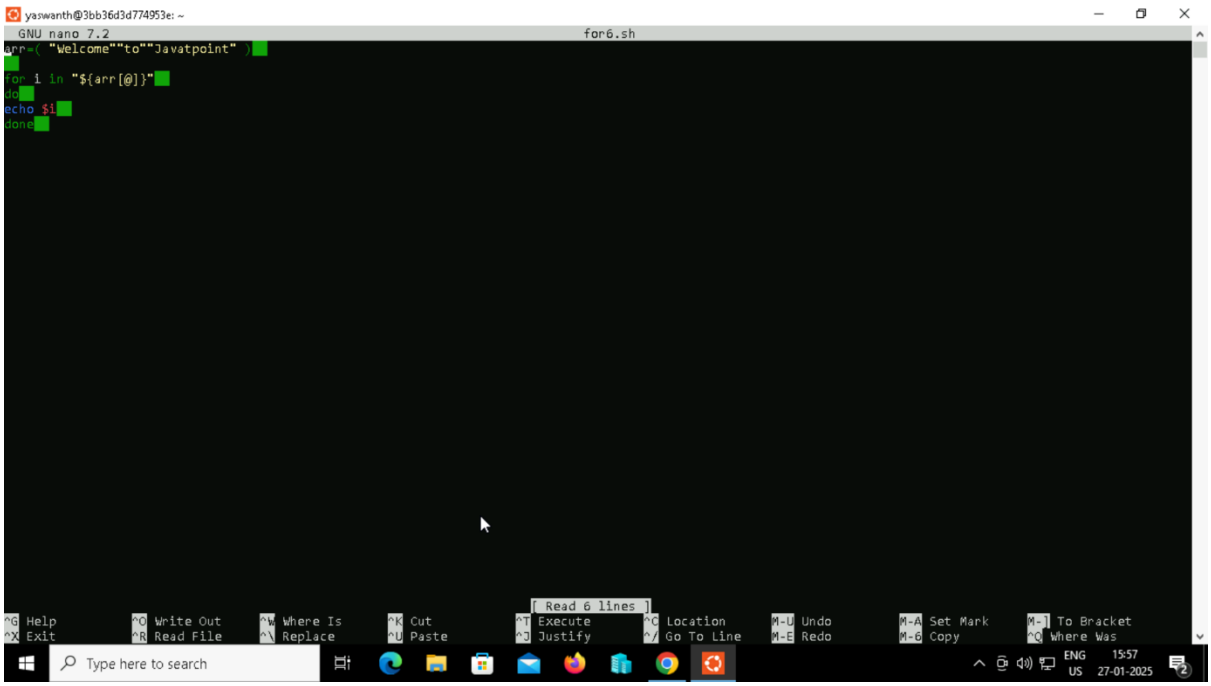
6. For Loop with a Break Statement :



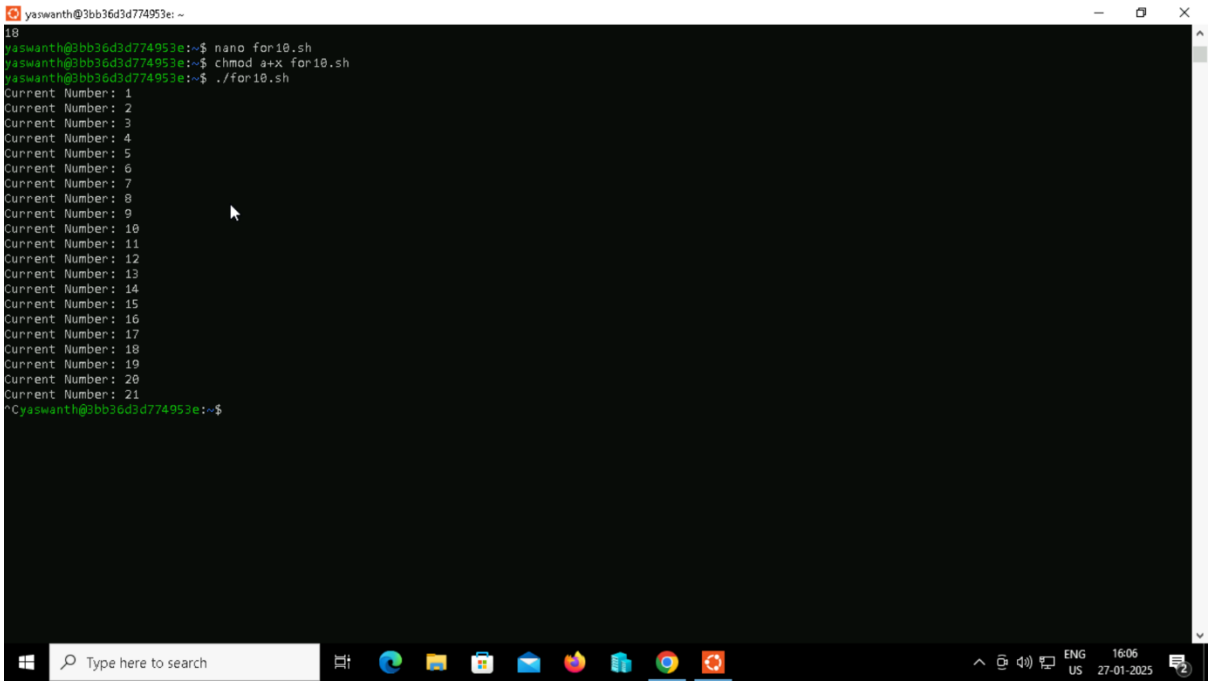


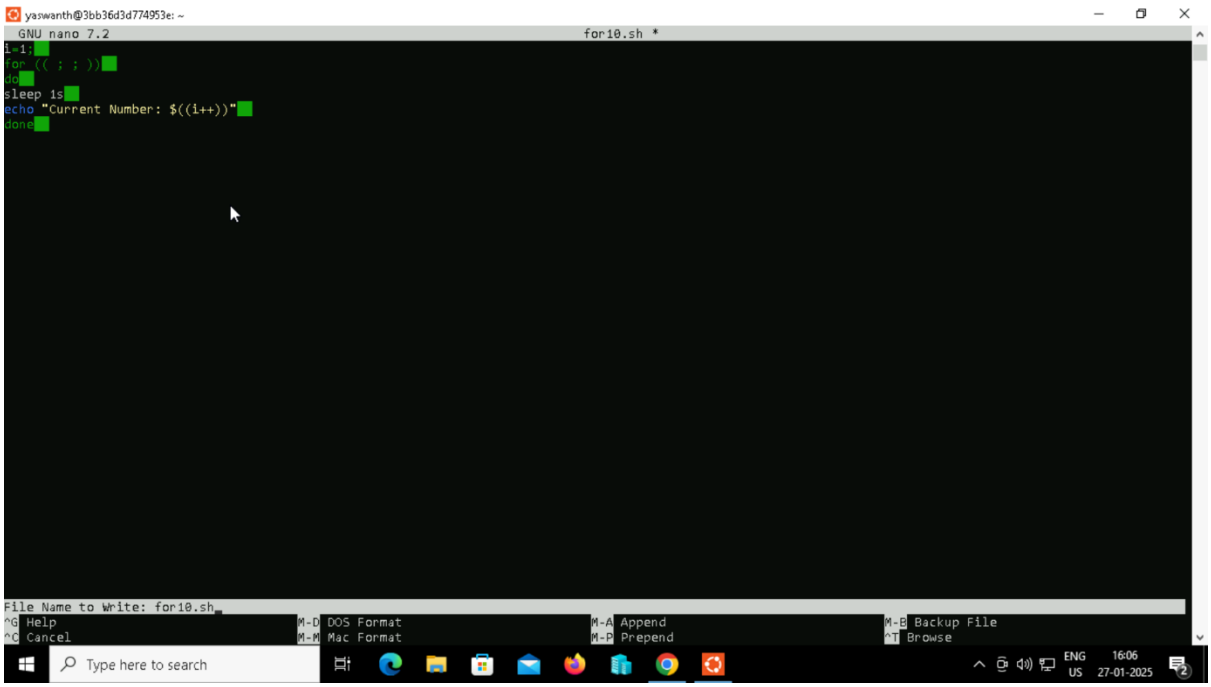
7. For Loop to Read Array Variables :



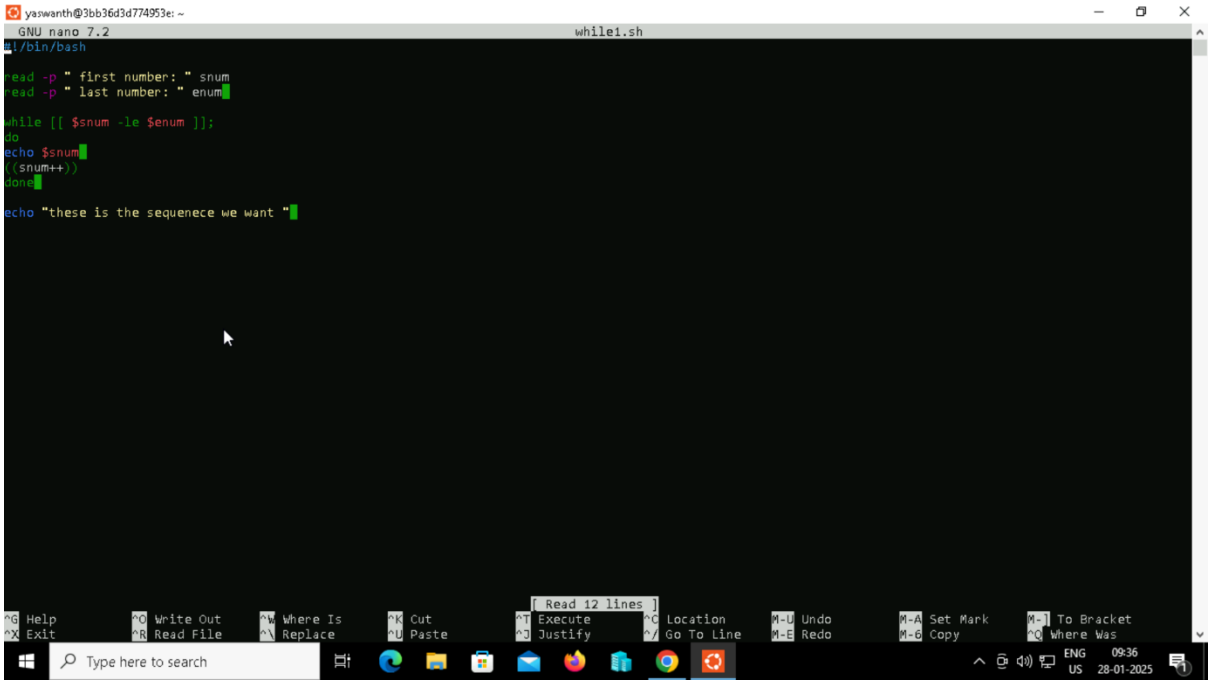


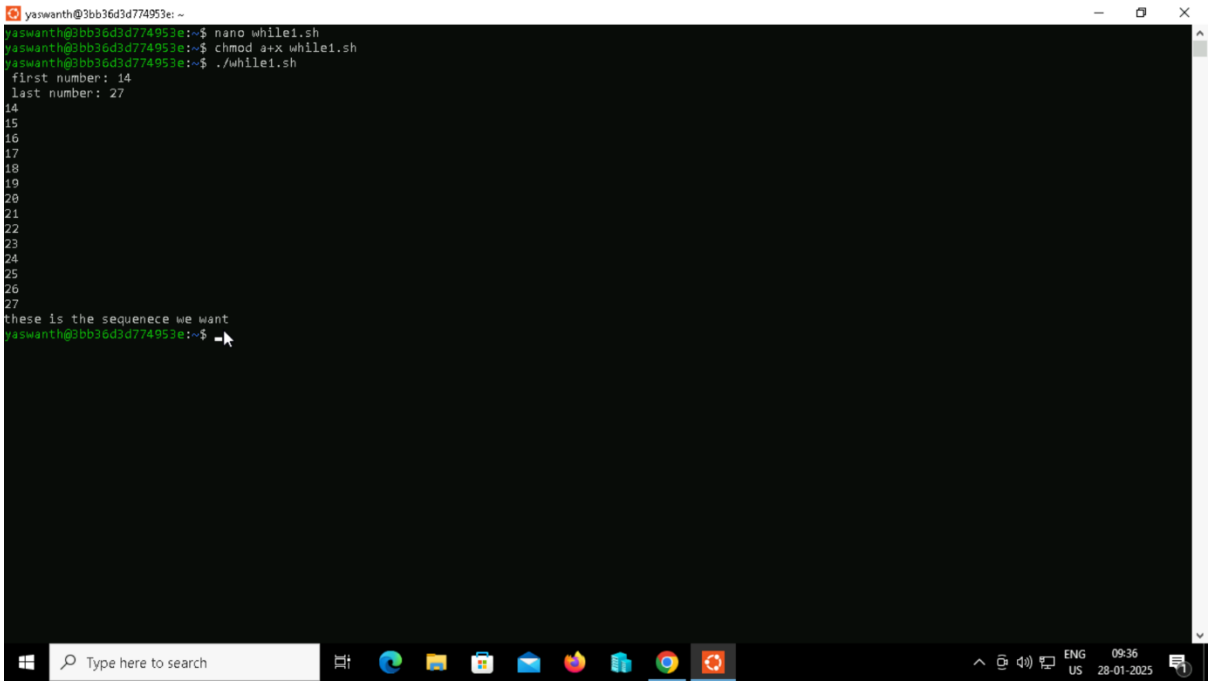
8 . Infinite Bash For Loop :



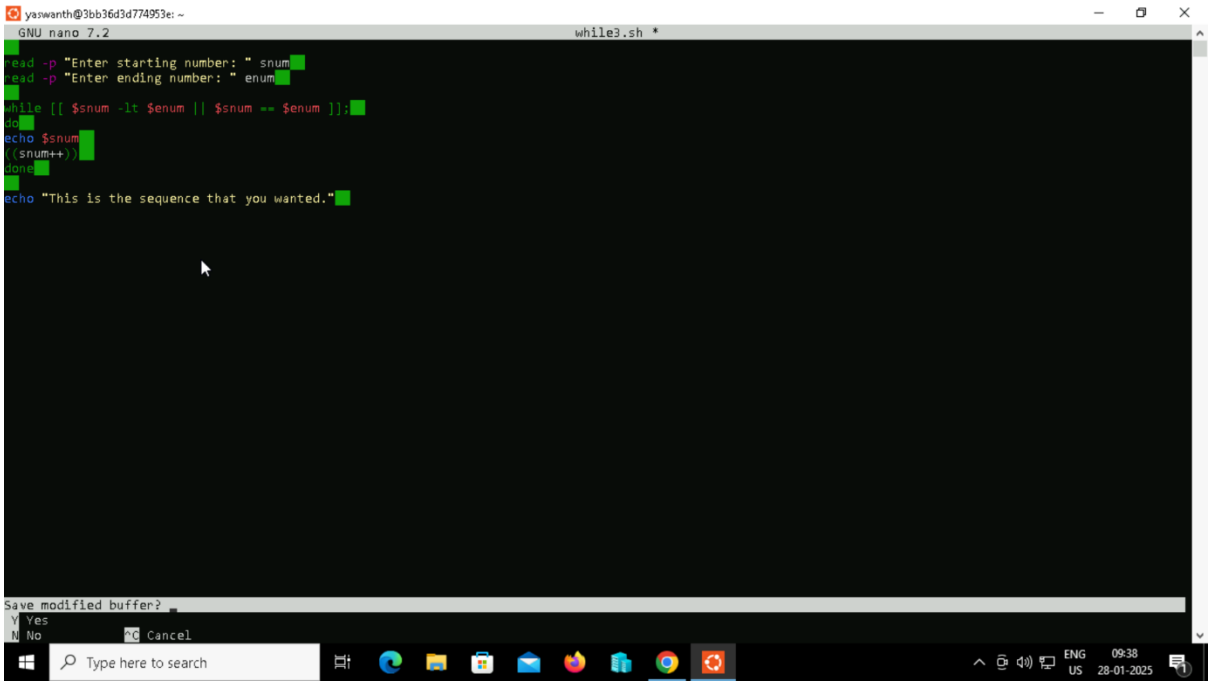


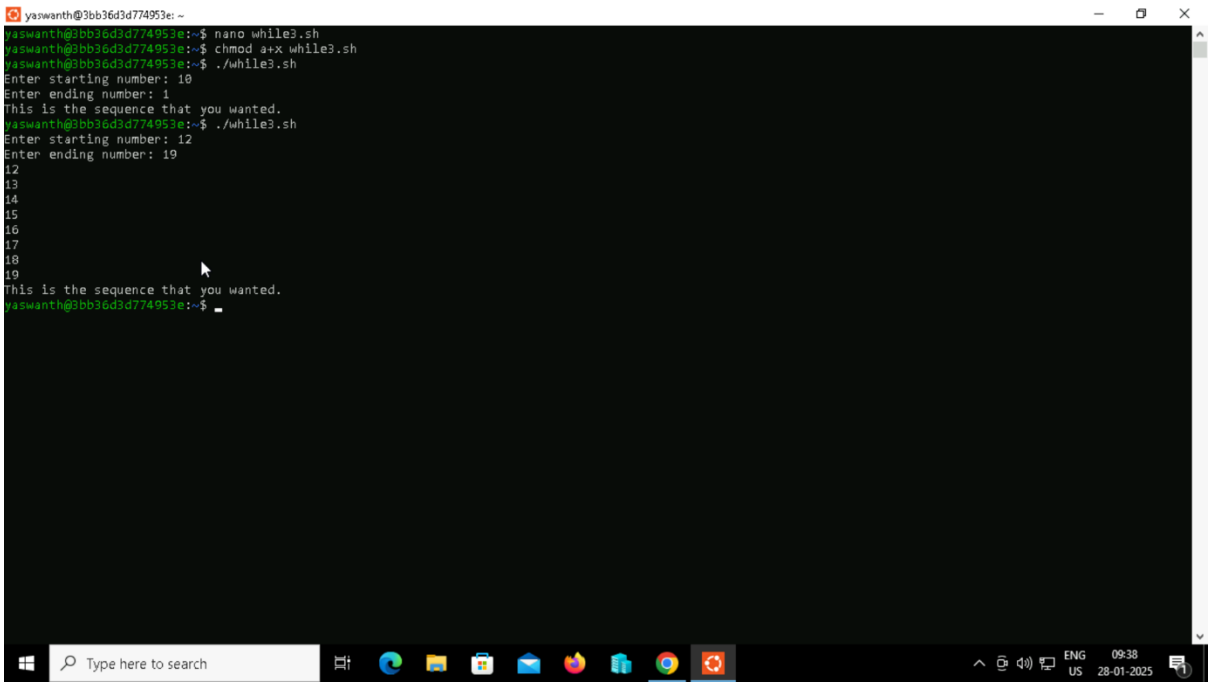
9. while loop is used with a single condition in expression. It is the basic example of while loop which will print series of numbers as per user input:



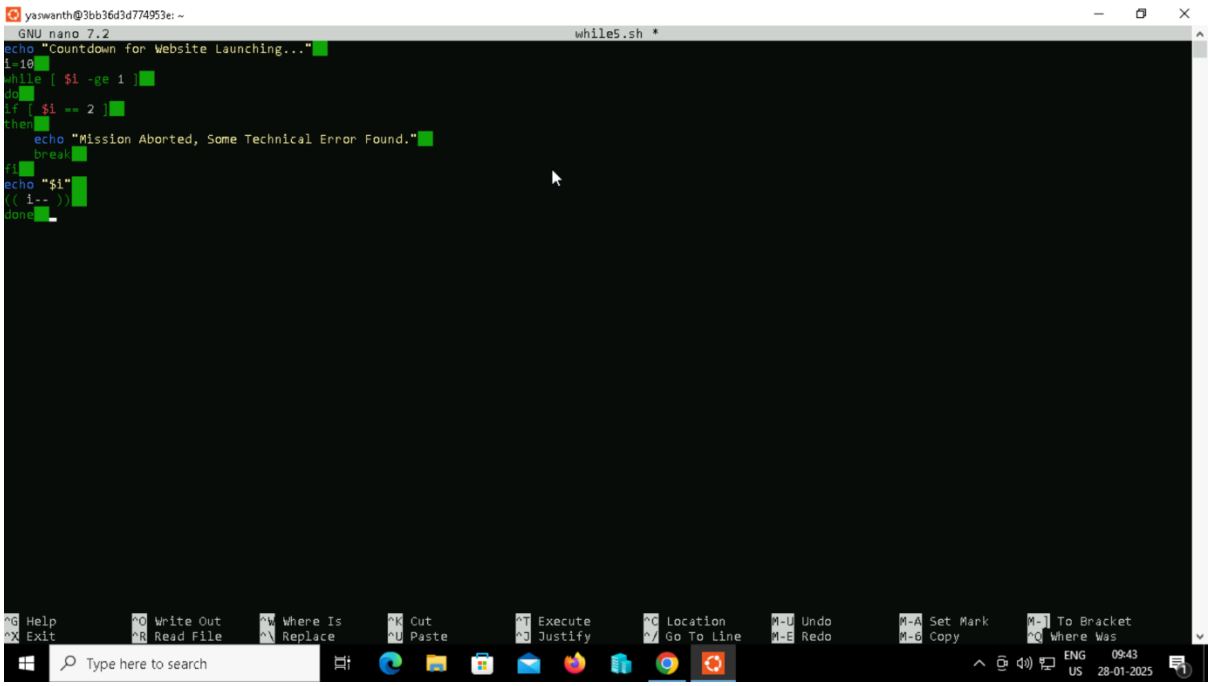


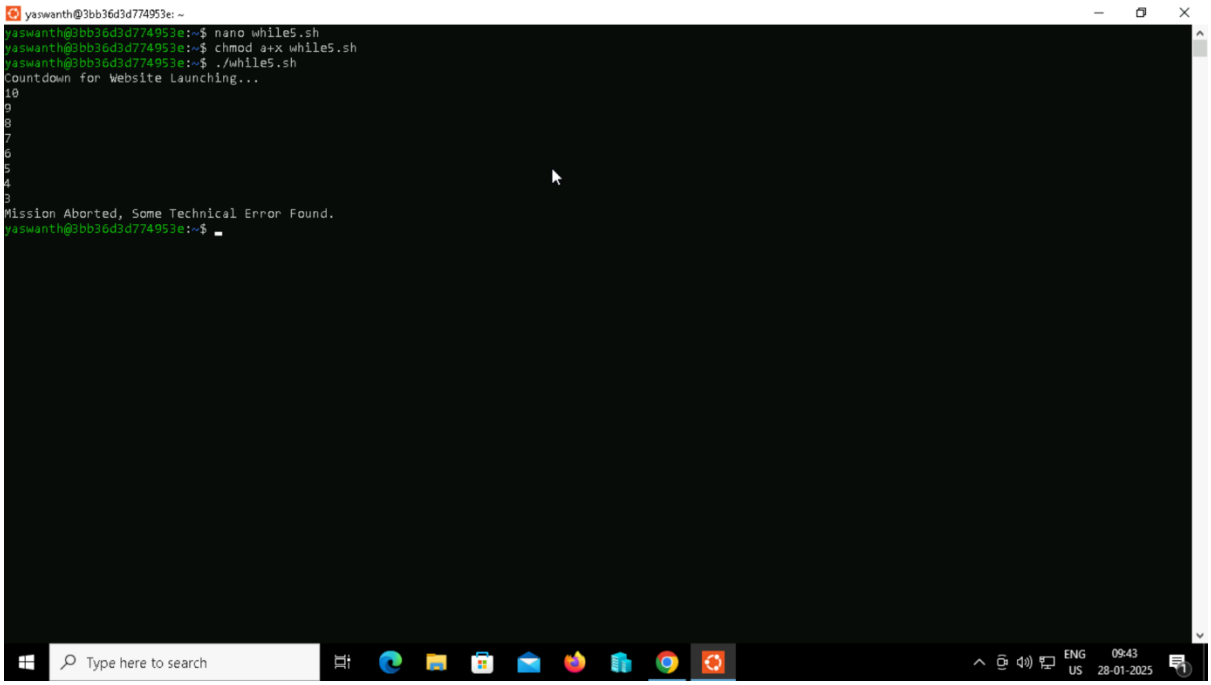
10.while loop with multiple conditions in the expression



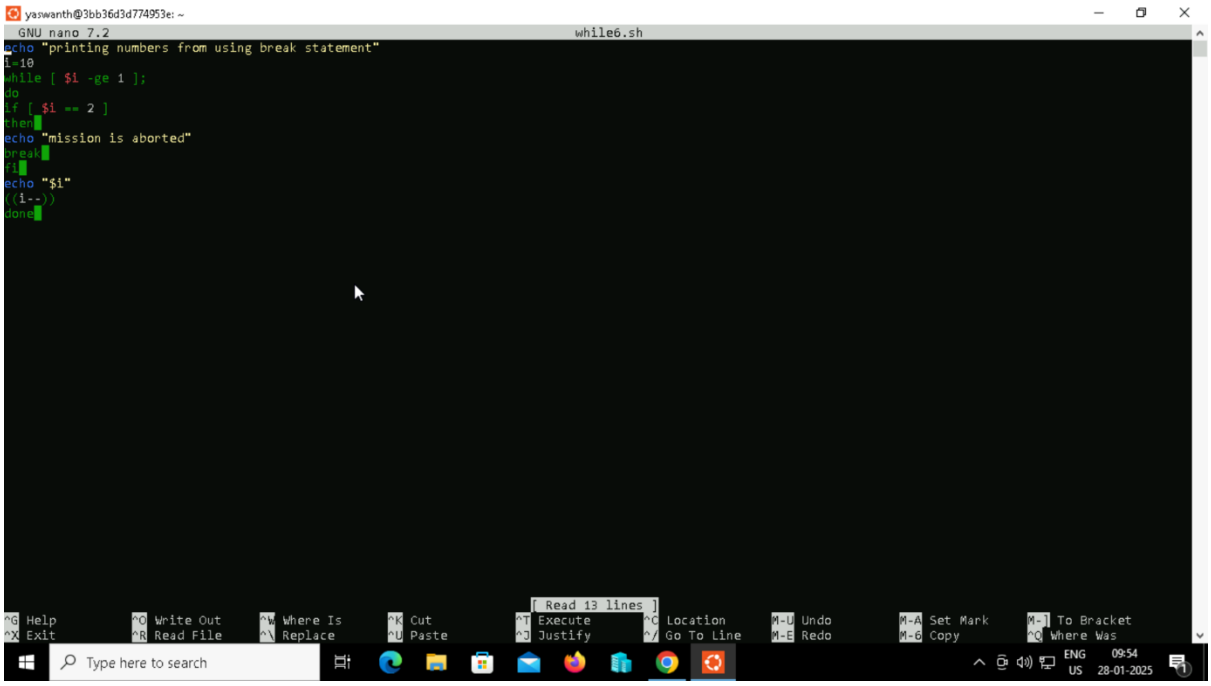


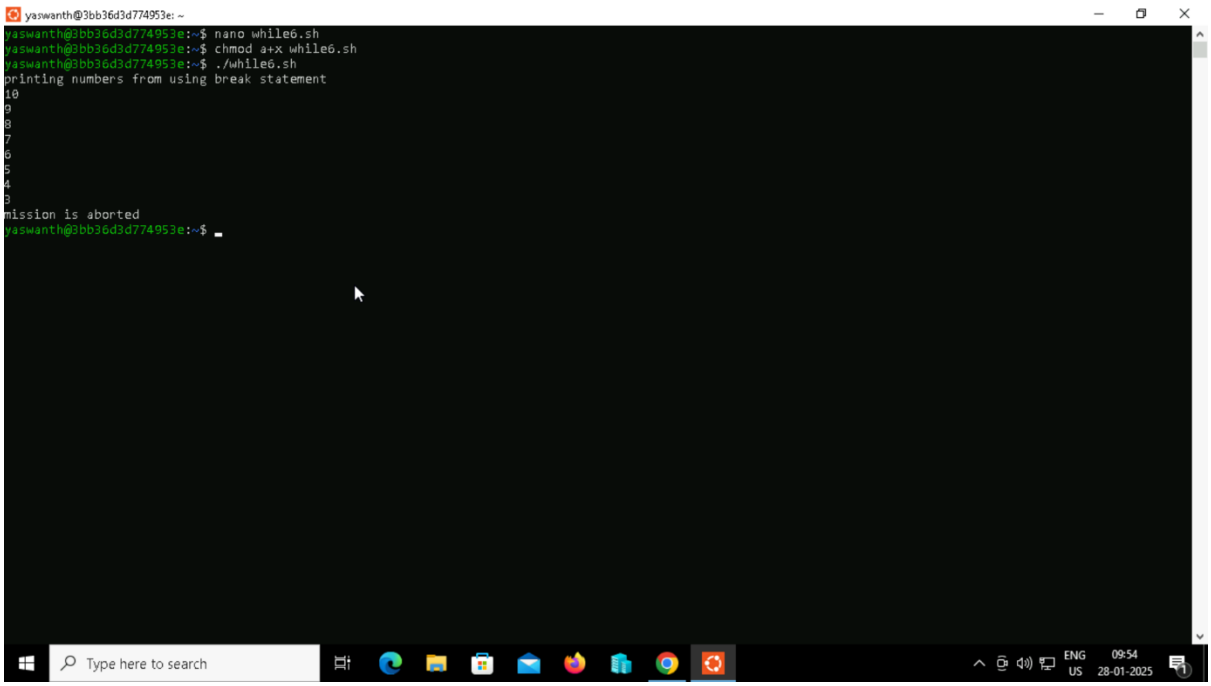
11.A break statement can be used to stop the loop as per the applied condition.



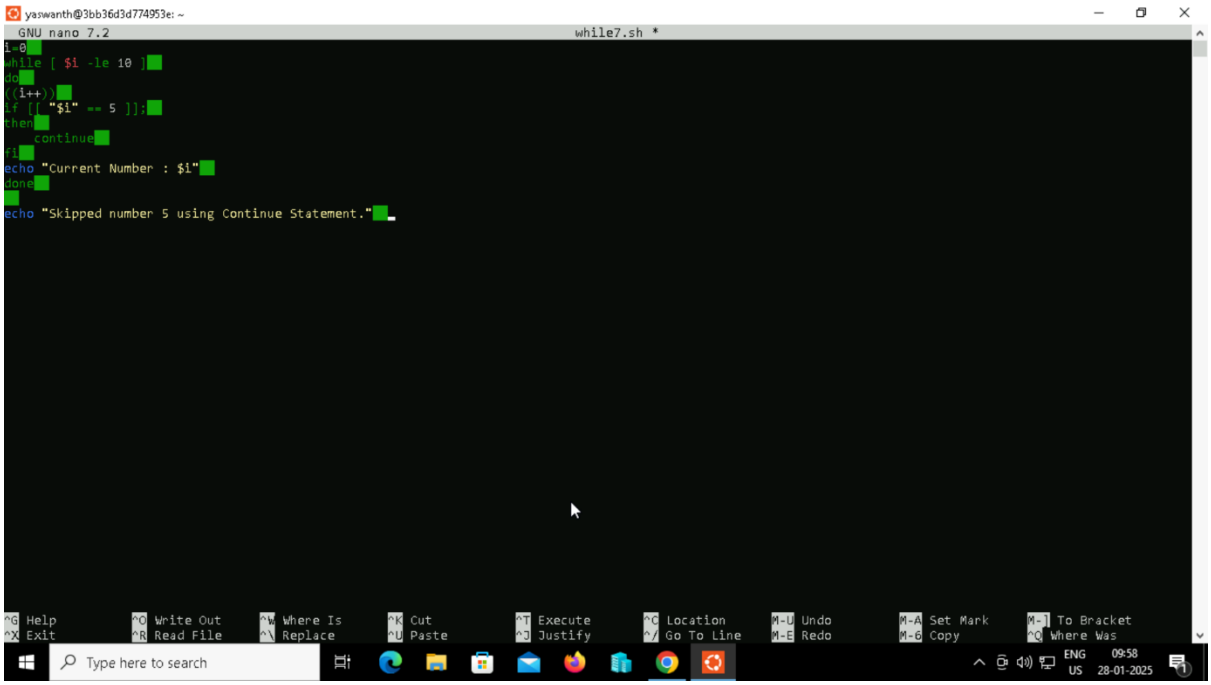


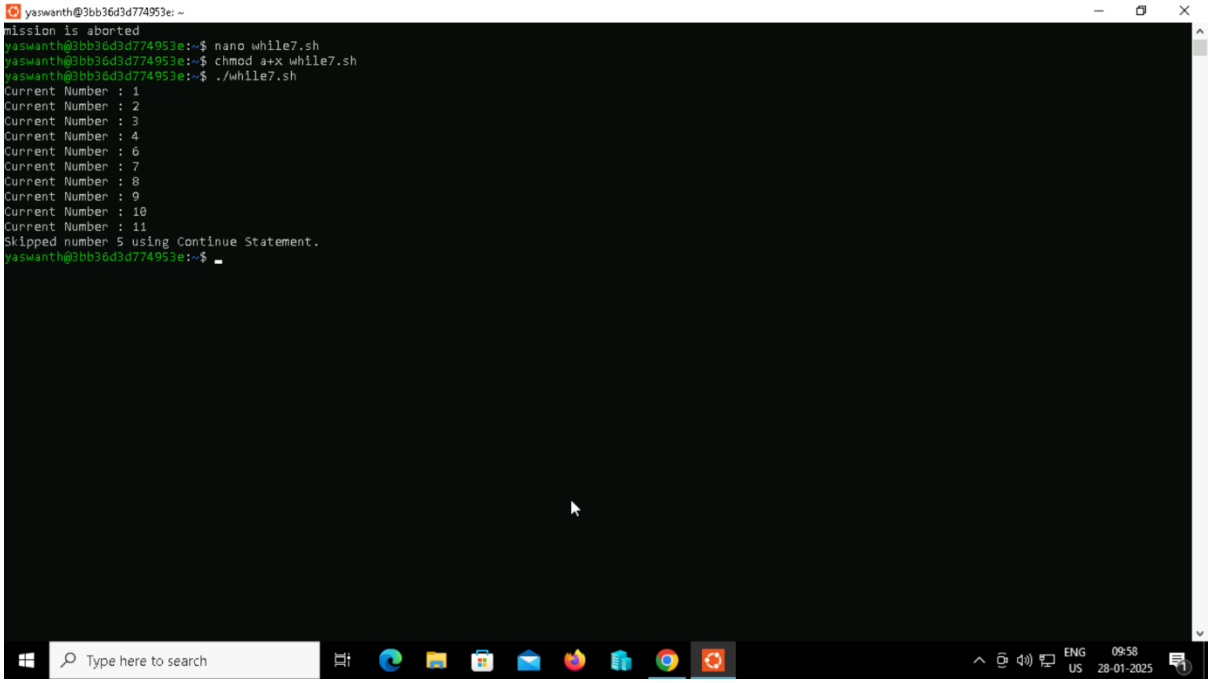
11.A break statement can be used to stop the loop as per the applied condition



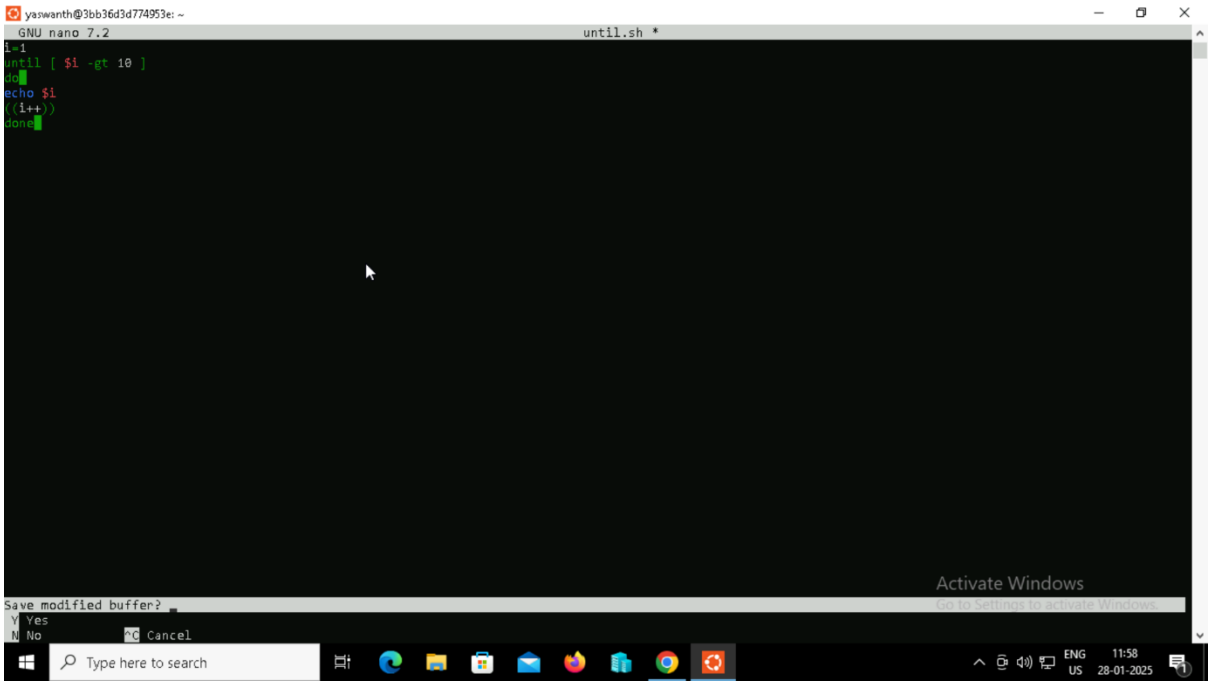


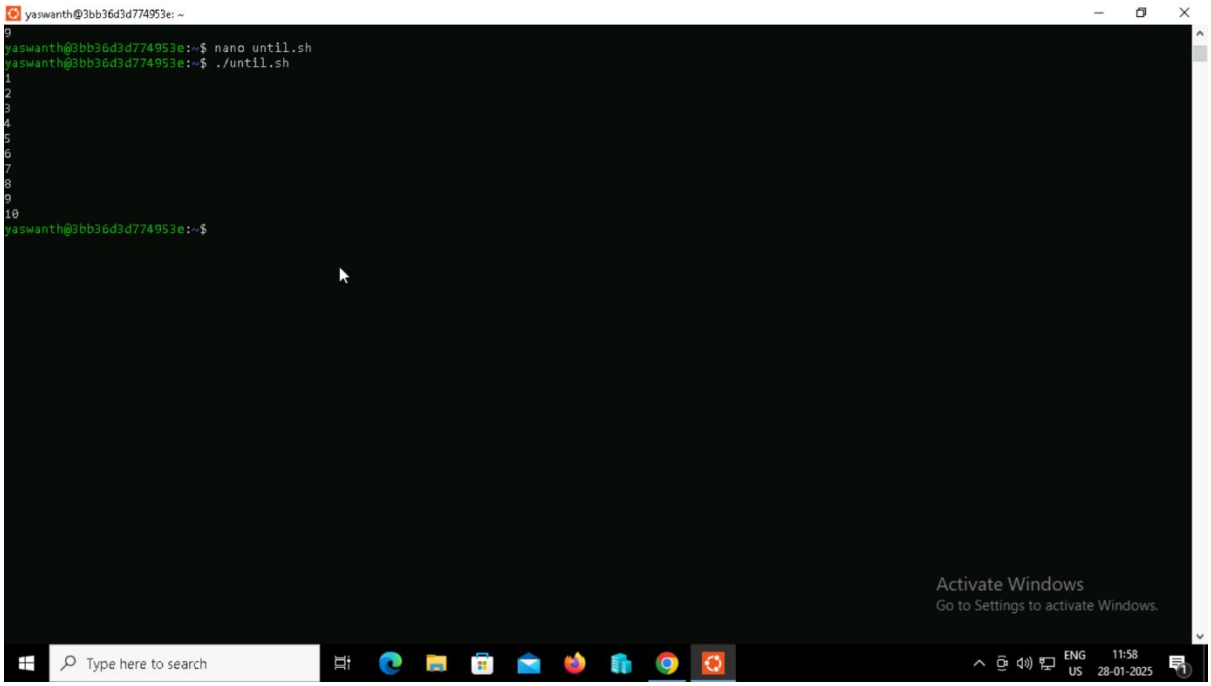
12.A continue statement can be used to skip the iteration for a specific condition inside the while loop.



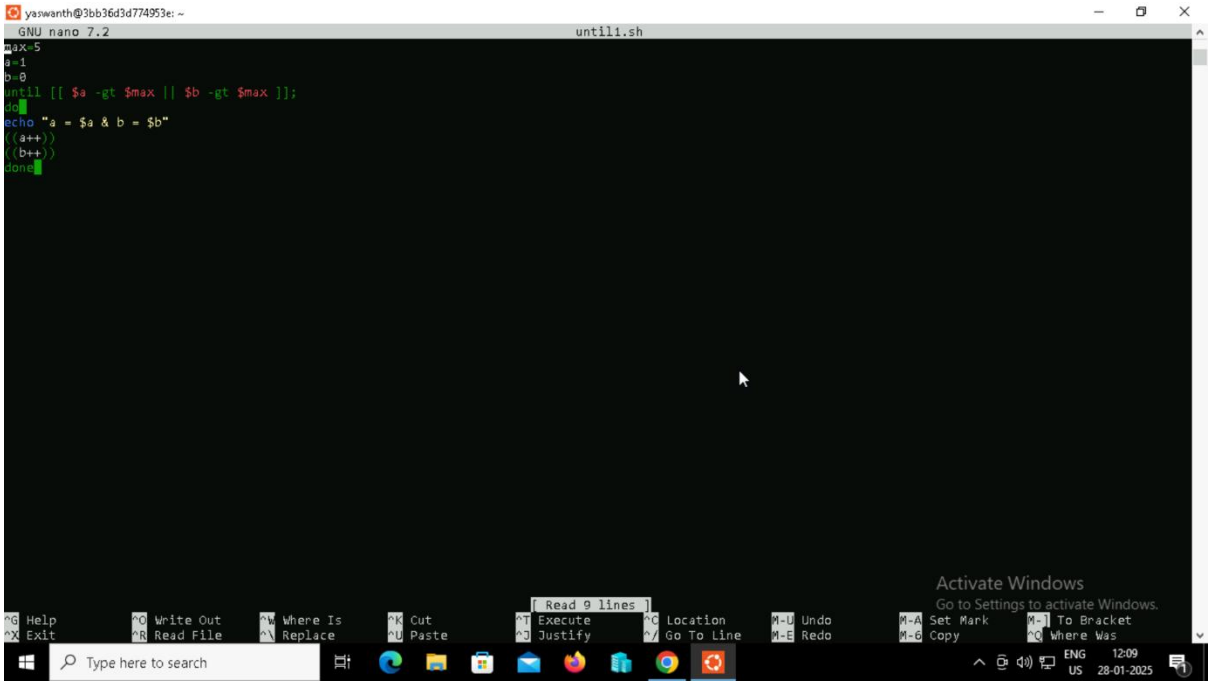


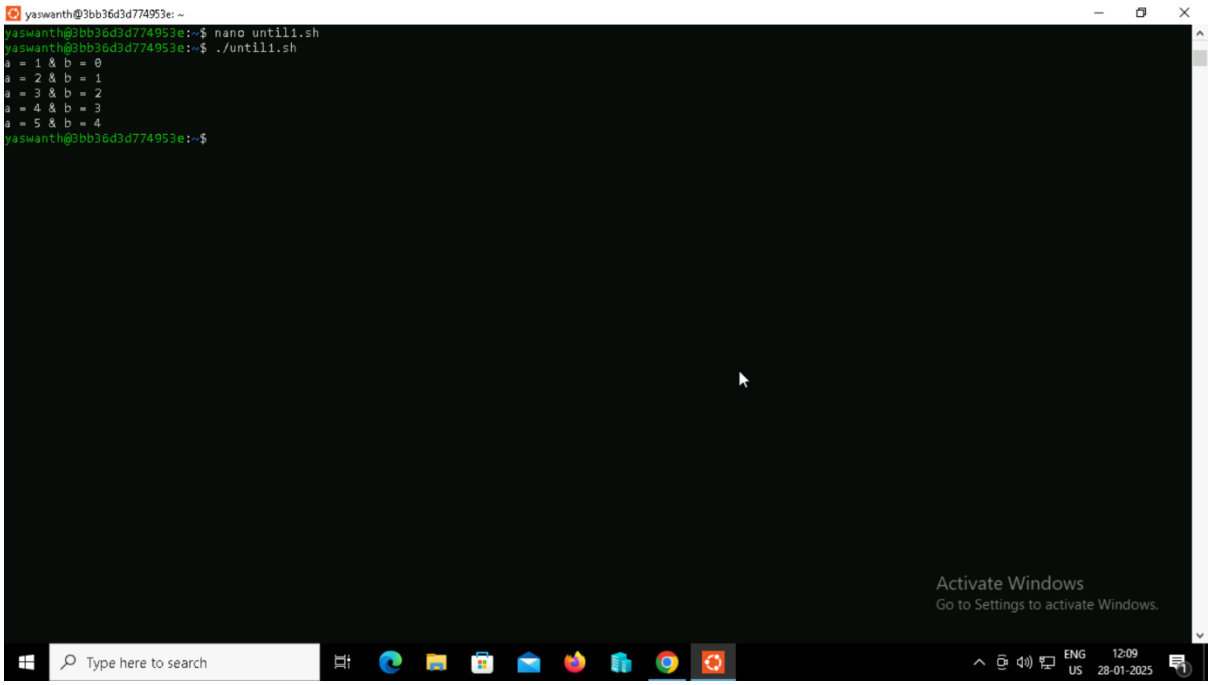
13. the until loop contains a single condition in expression. It is the basic example of until loop which will print series of numbers from 1 to 10



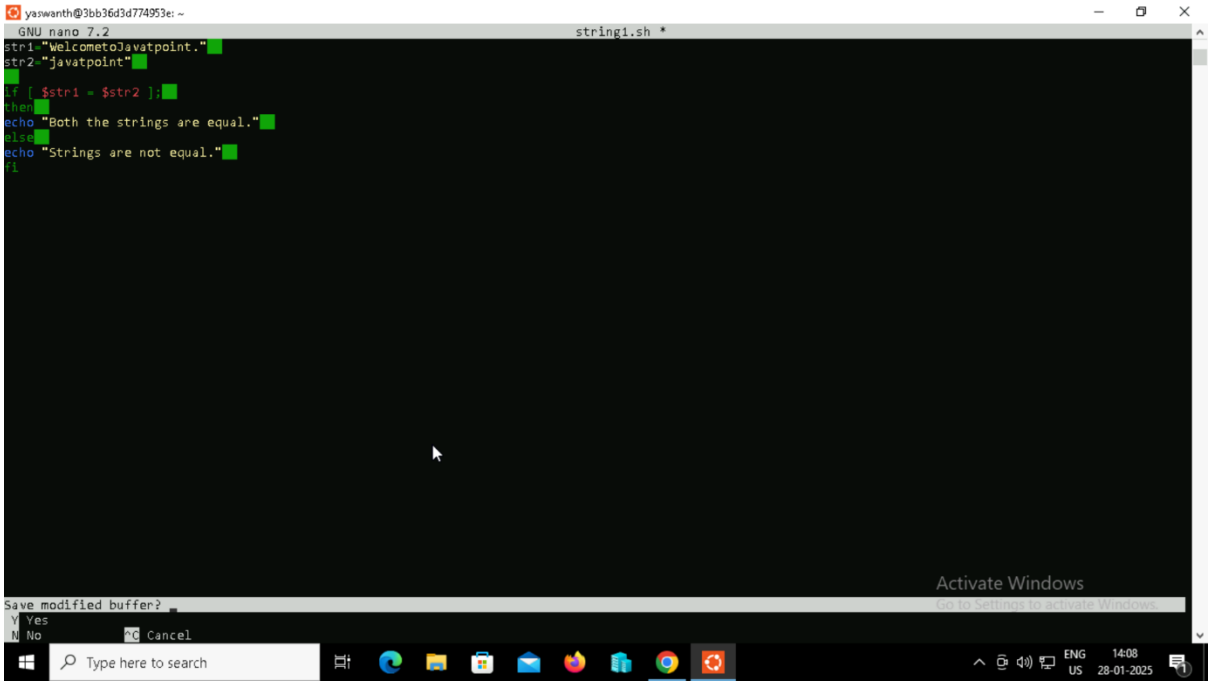


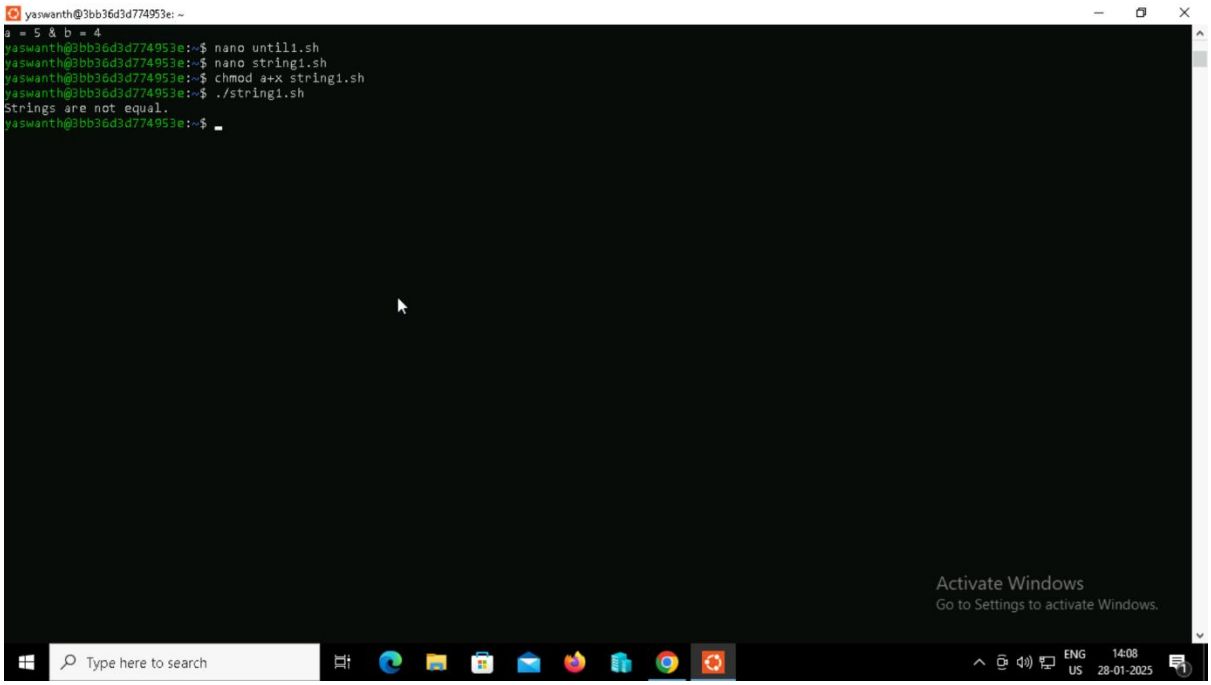
14. with multiple conditions in an expression



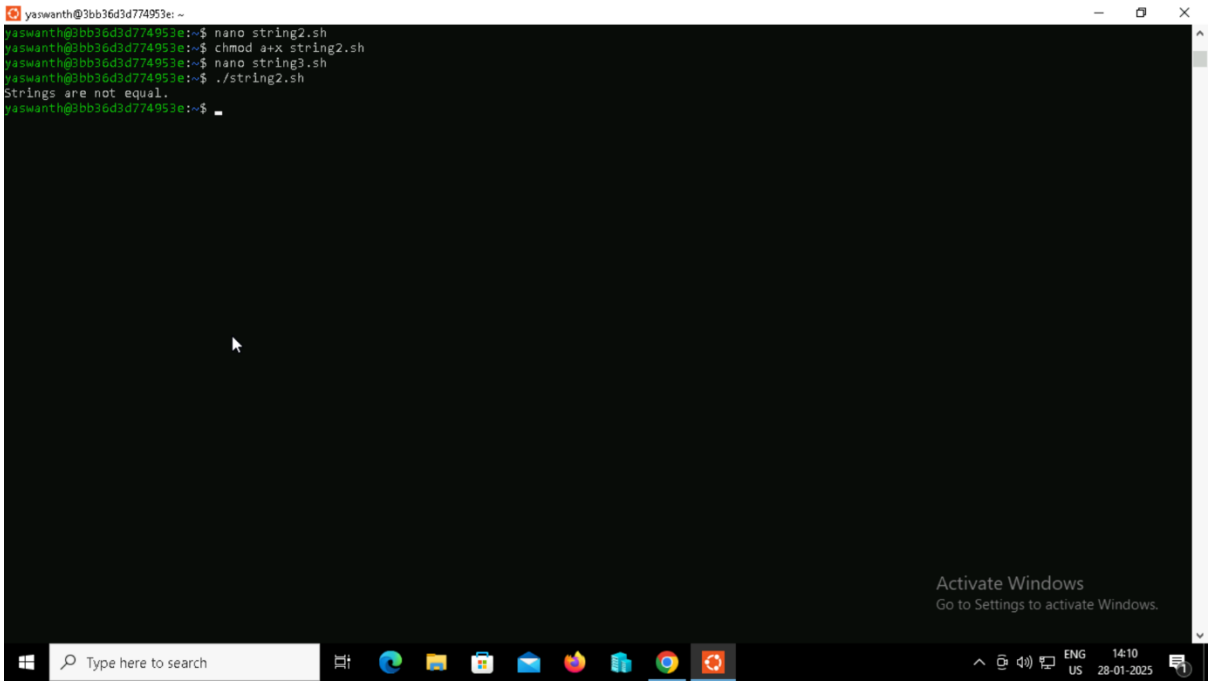


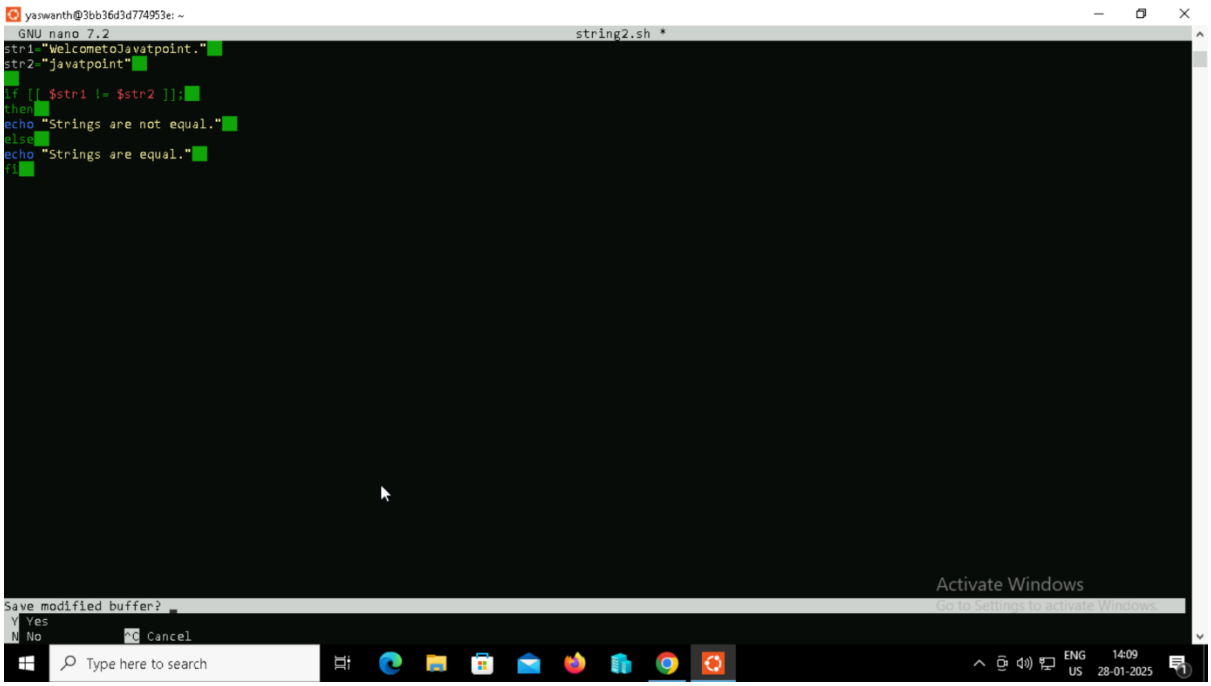
15. An equal operator (=) is used to check whether two strings are equal.



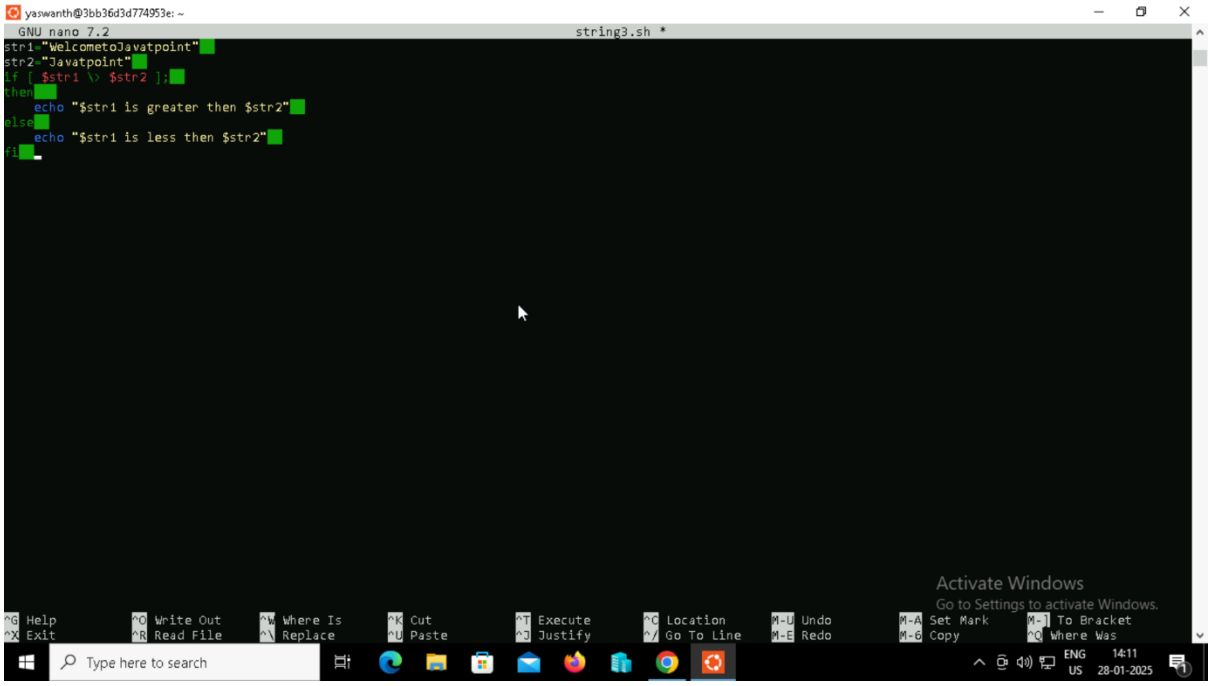


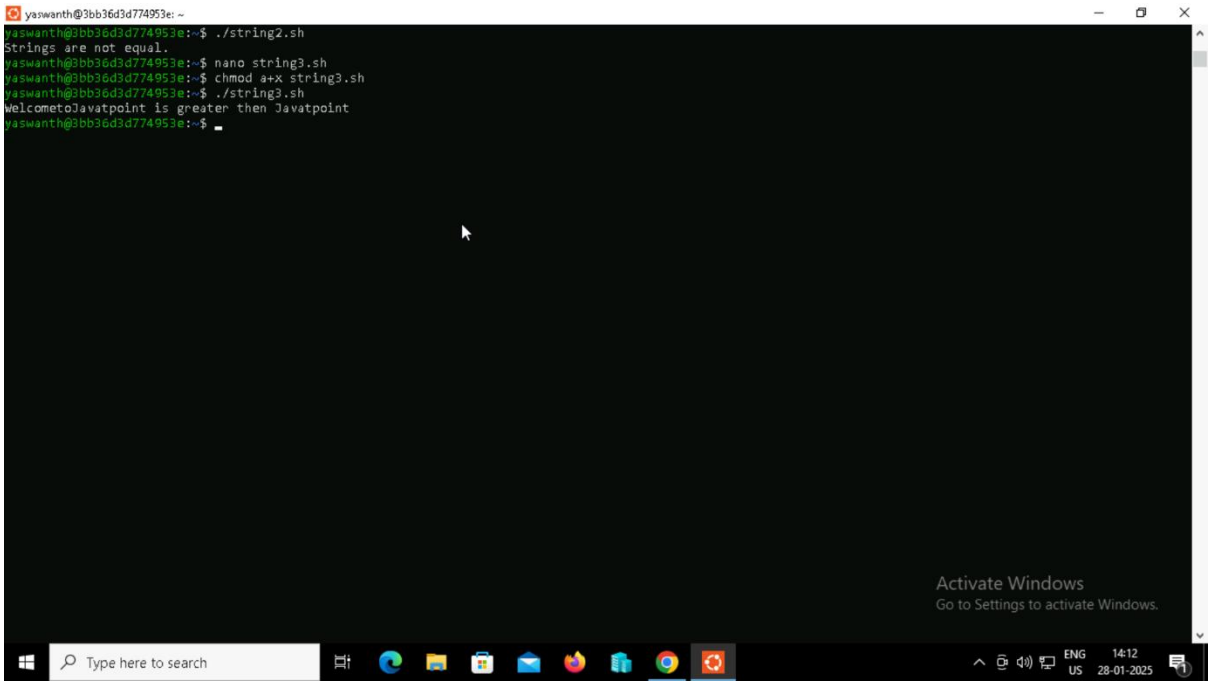
16. Not equal operator (!=) is used to define that strings are not equal.



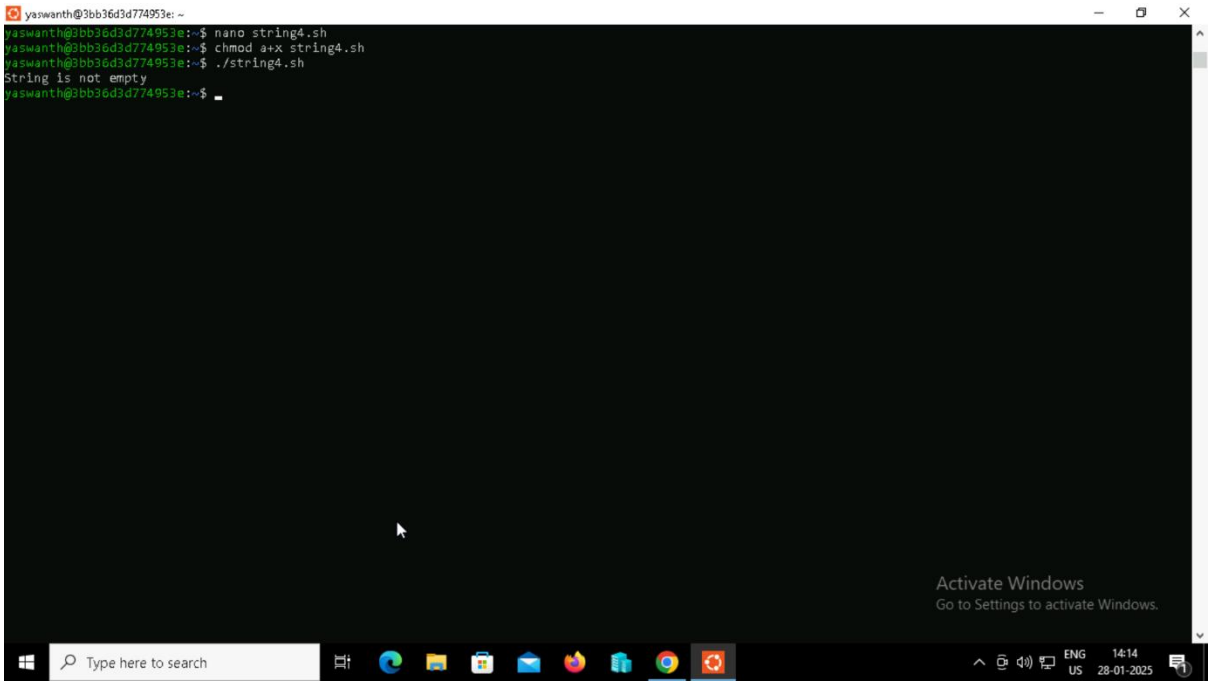


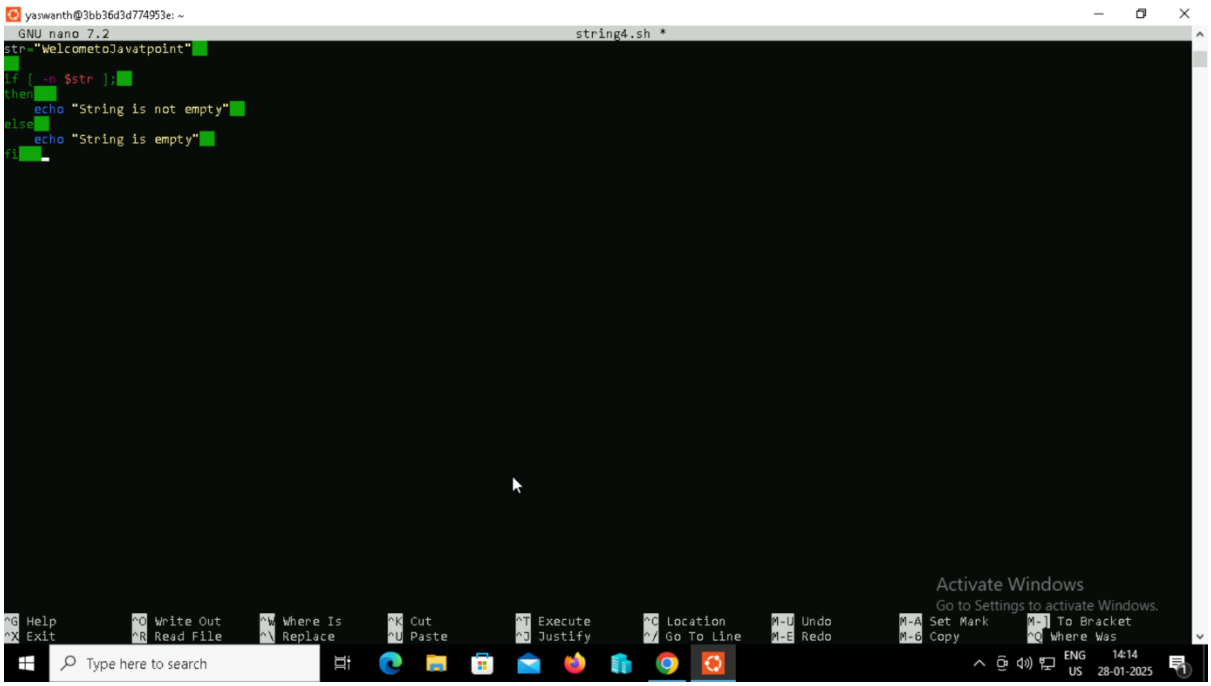
17. The 'greater than operator (\>)` is used to check if string1 is greater than string2.



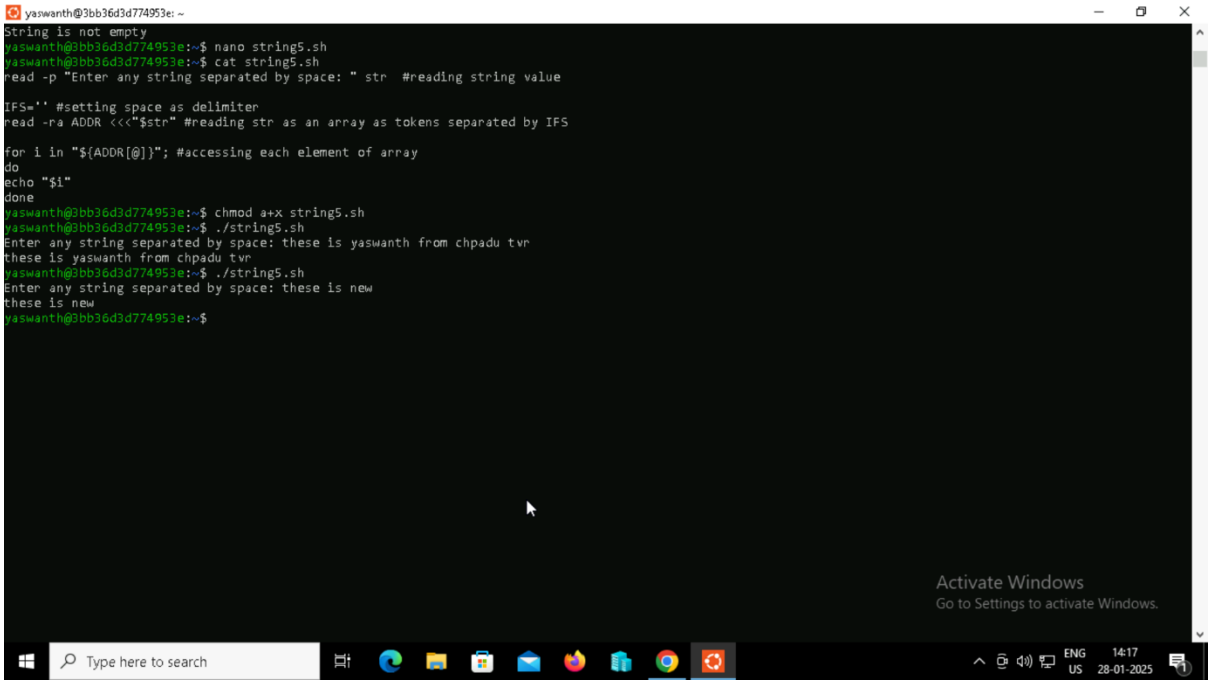


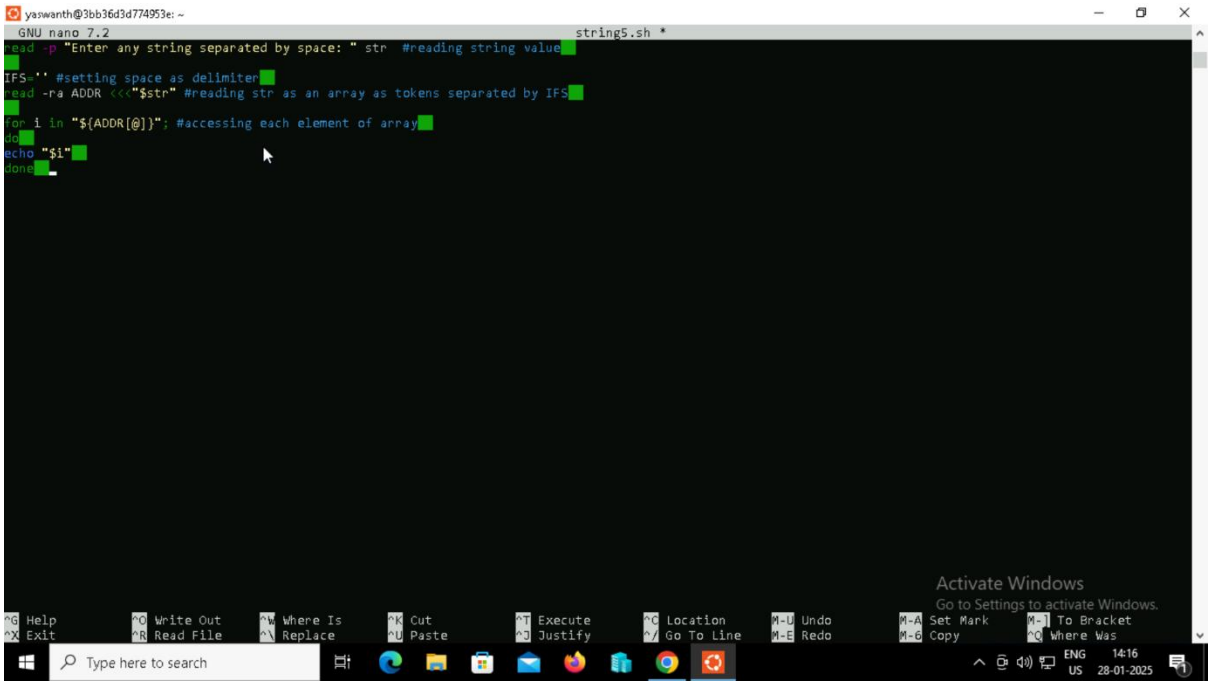
18. This operator is used to check if the string is zero or greater than zero



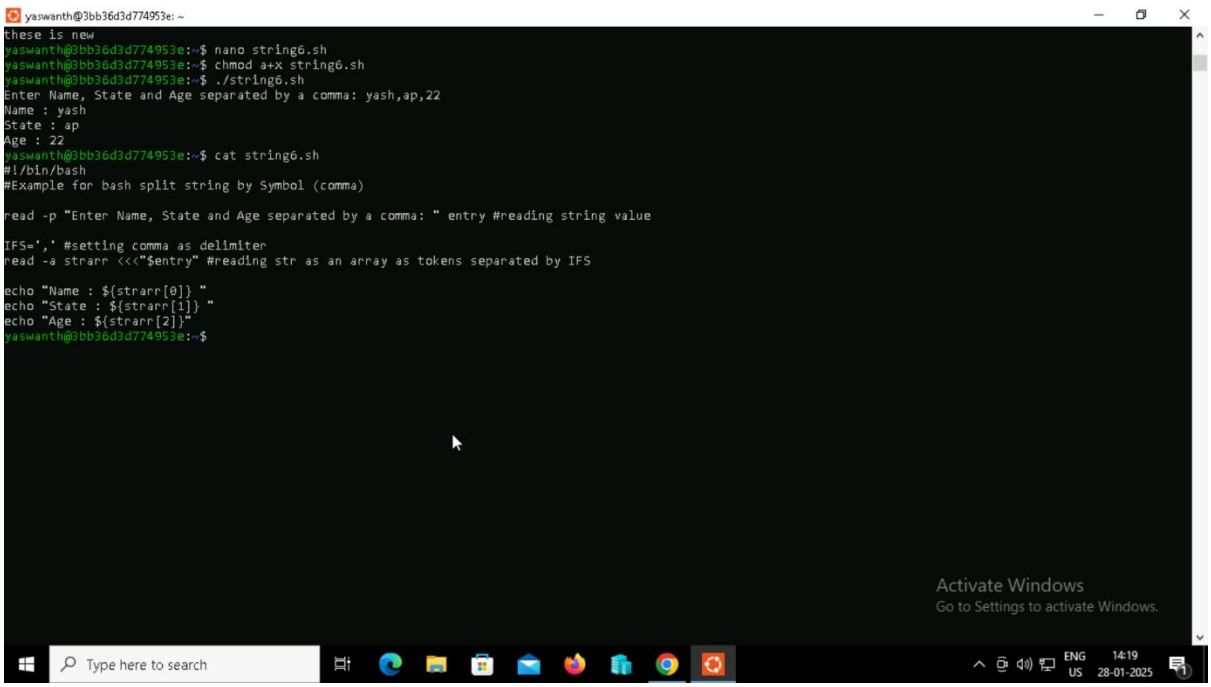


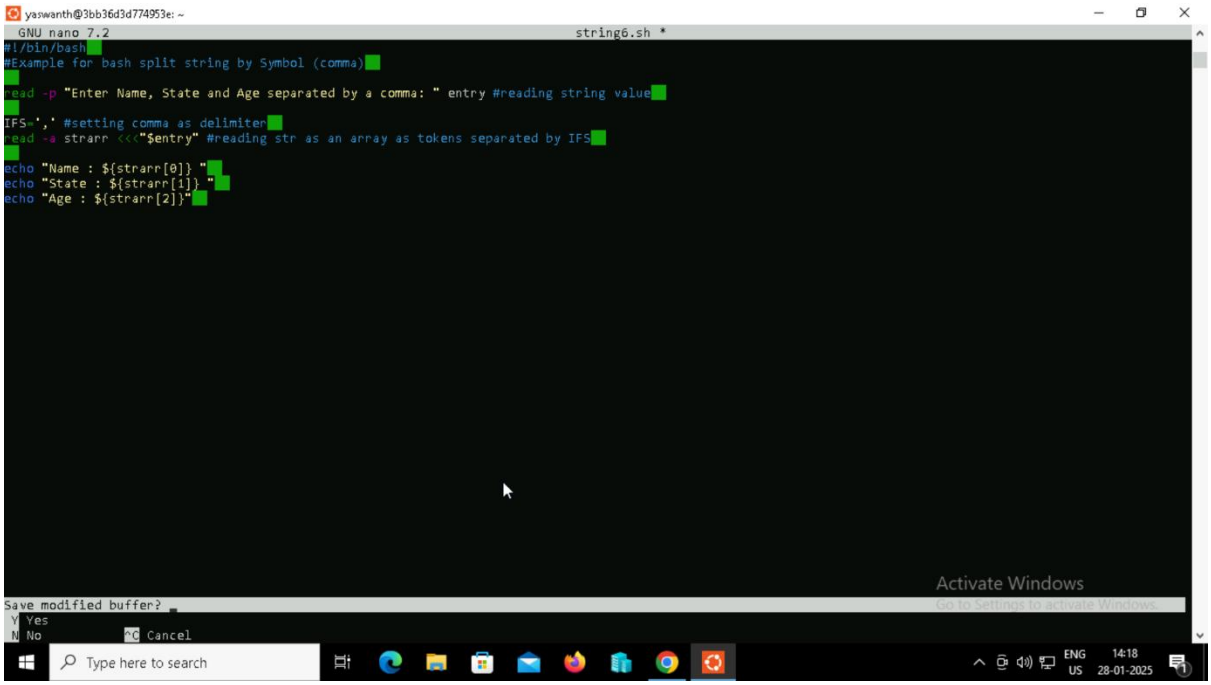
19. This example, a string is **split** using a space character delimiter.



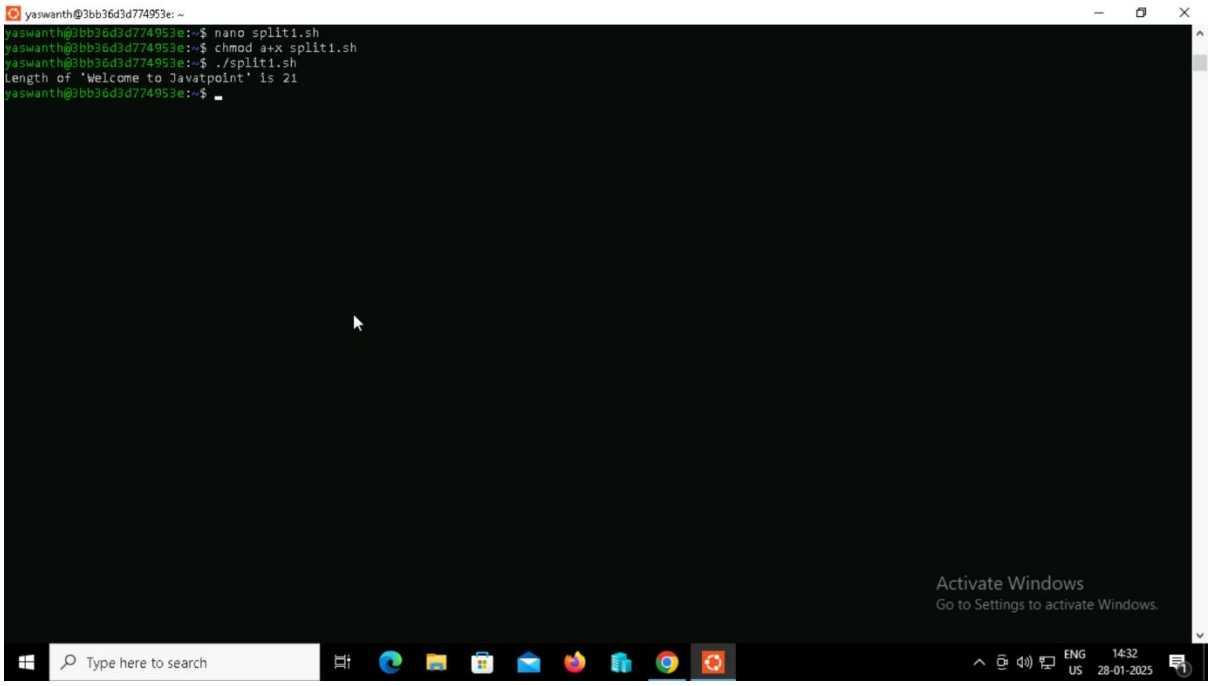


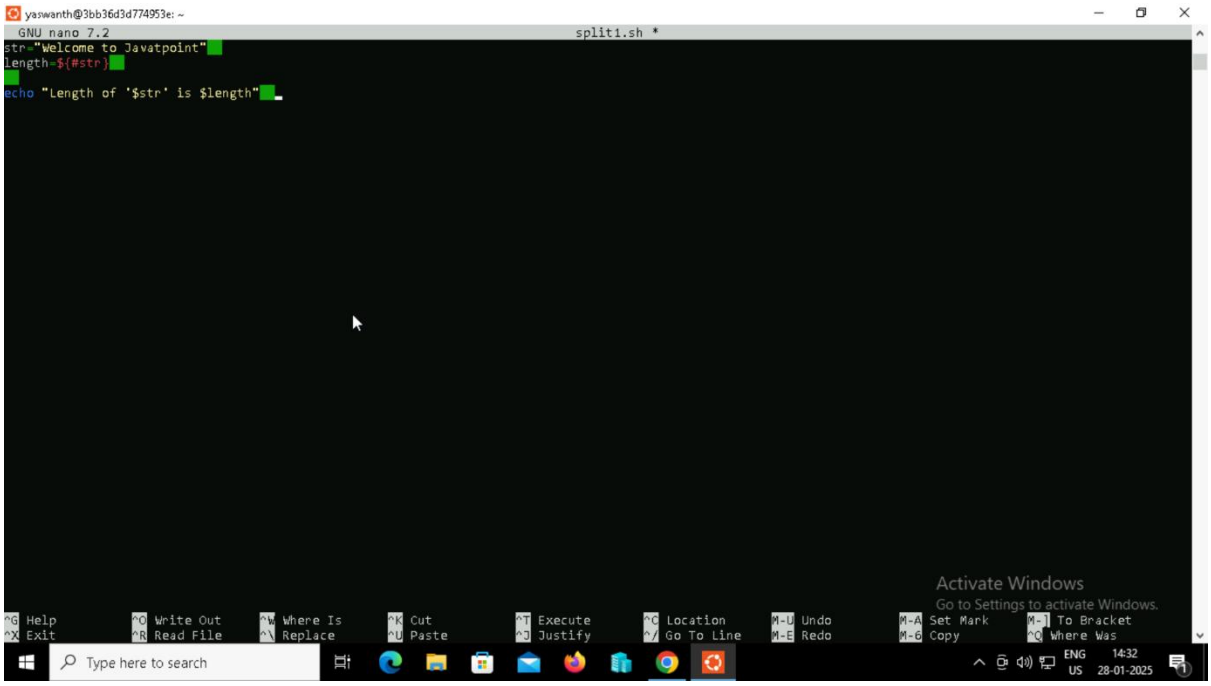
20 . This example, a string is **split** using a comma (,) symbol character as a delimiter.



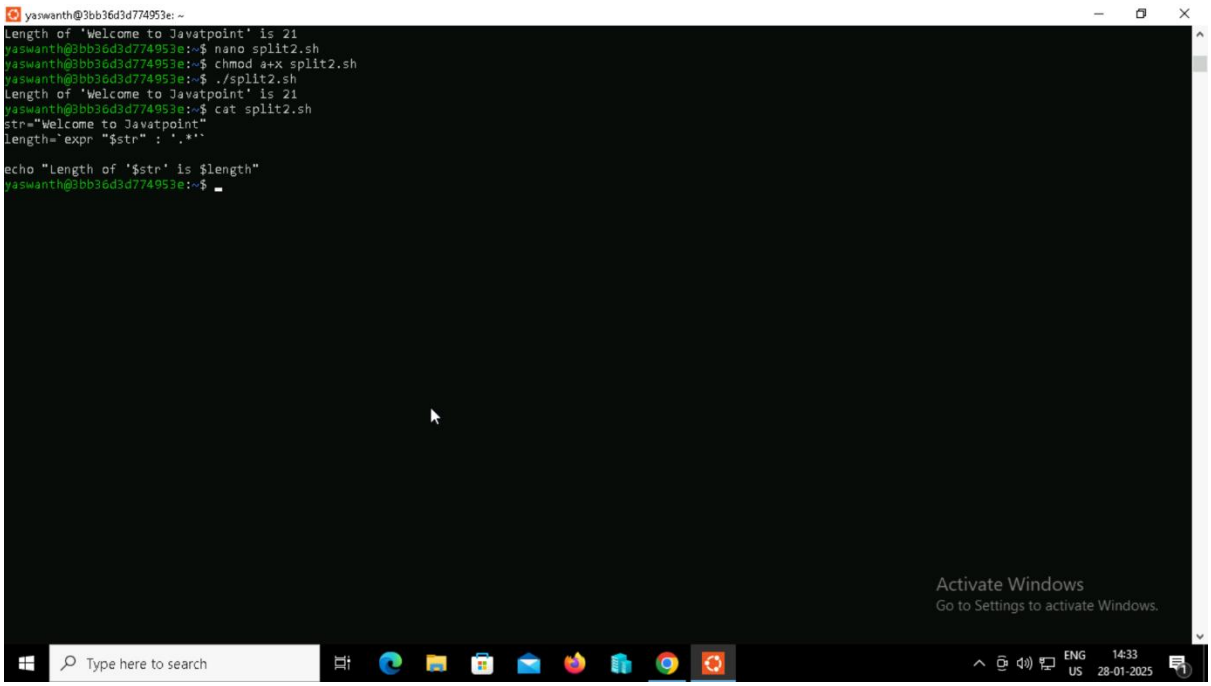


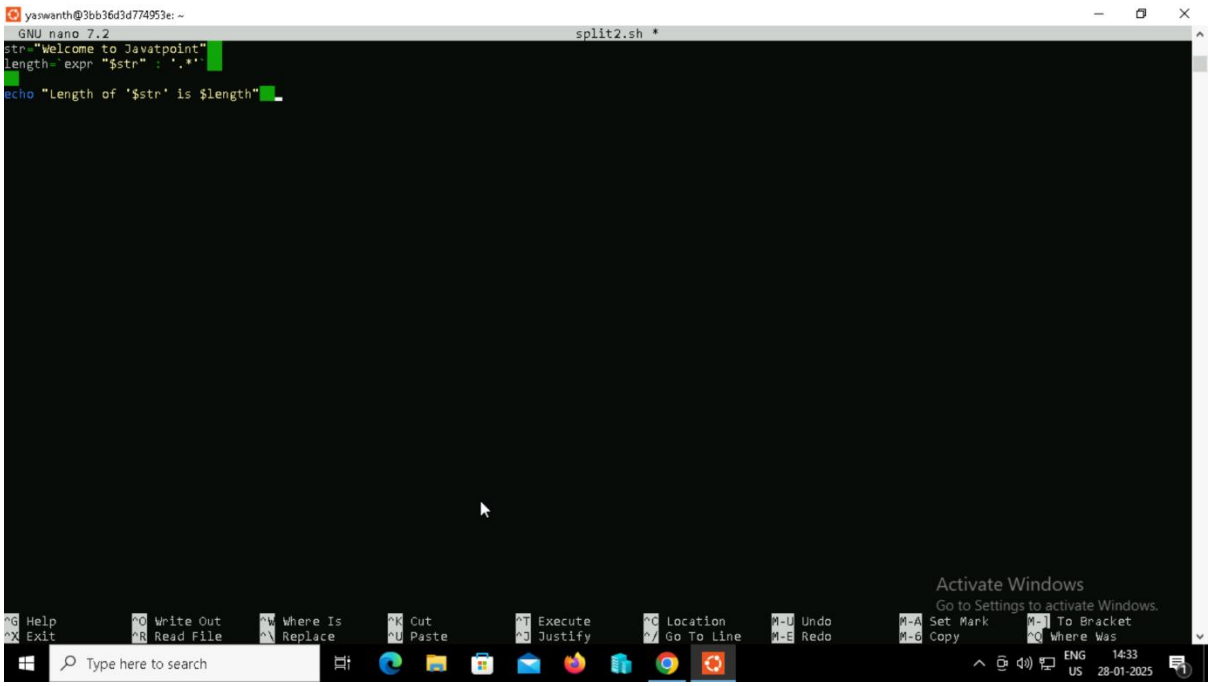
21. To calculate the length of a string is to use '#' symbol. In this example, we have used **$[#string_variable_name}** to **find** the length of a string.



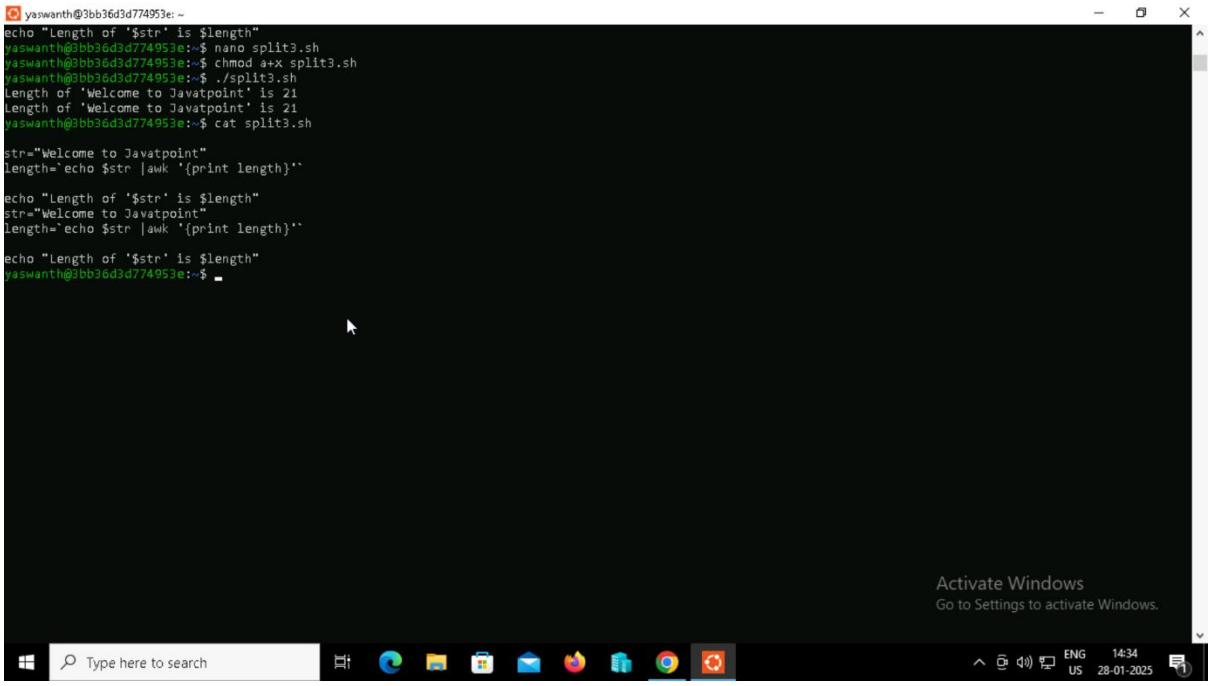


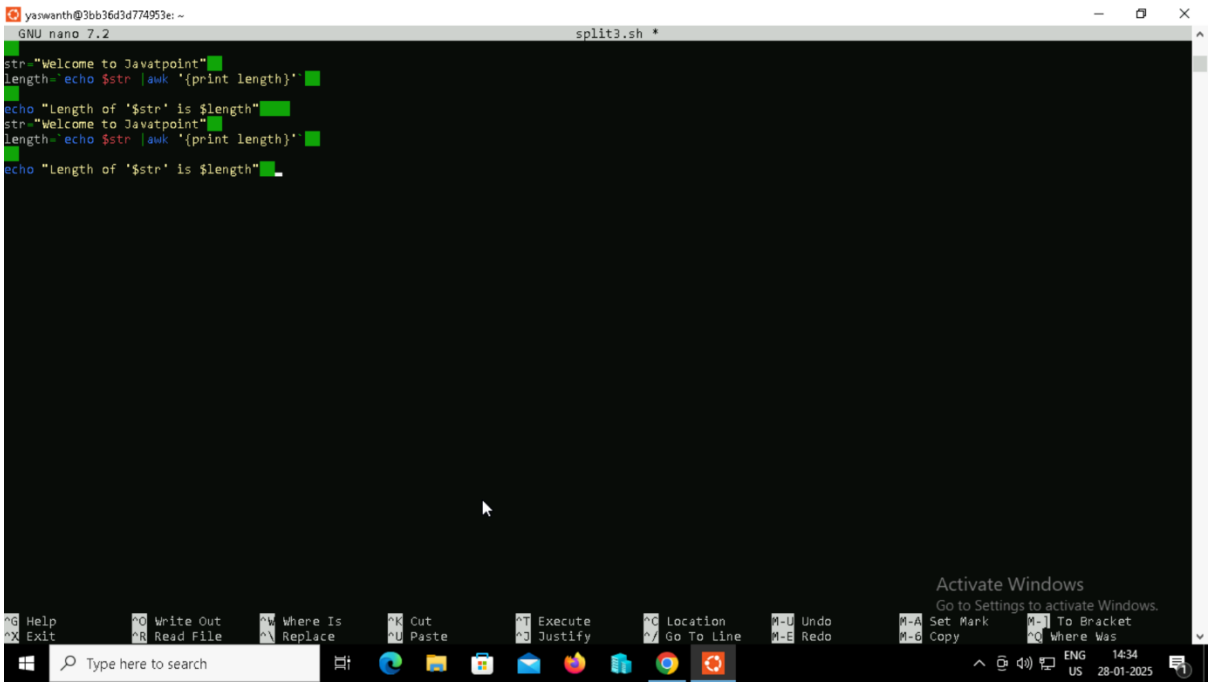
22. to calculate the length of a string is to use `expr` command with the 'length' keyword. In this example, we have used `**expr length "$str"**` to **find** the length of a string.



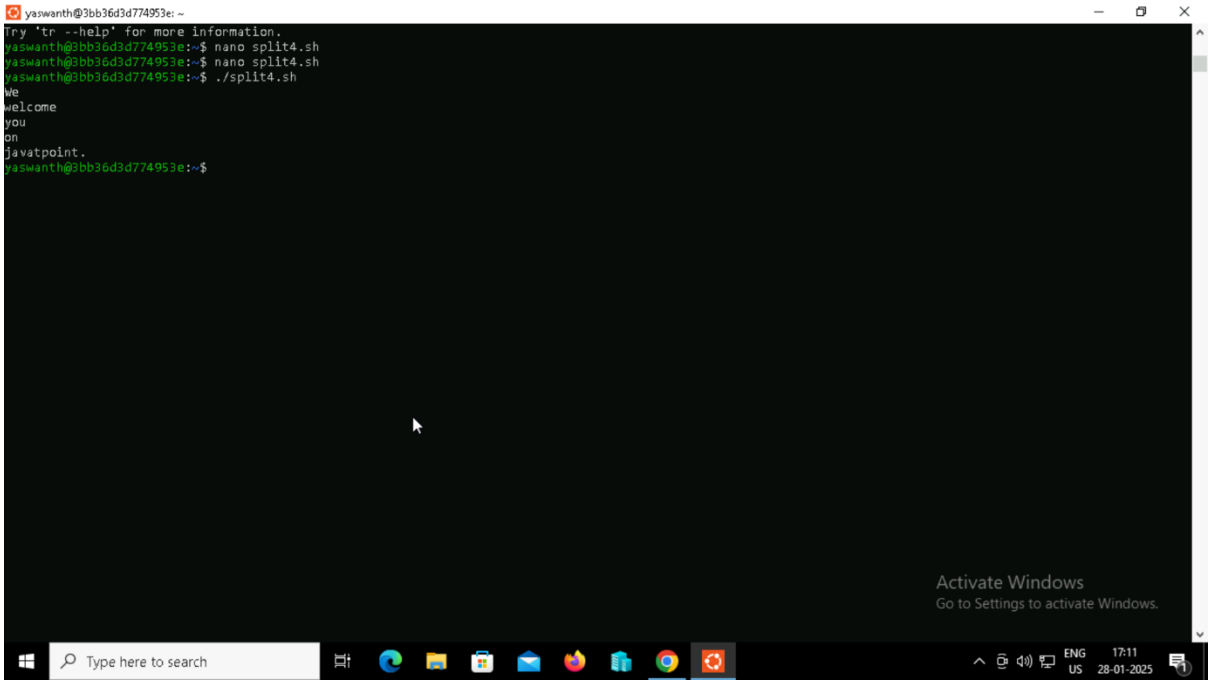


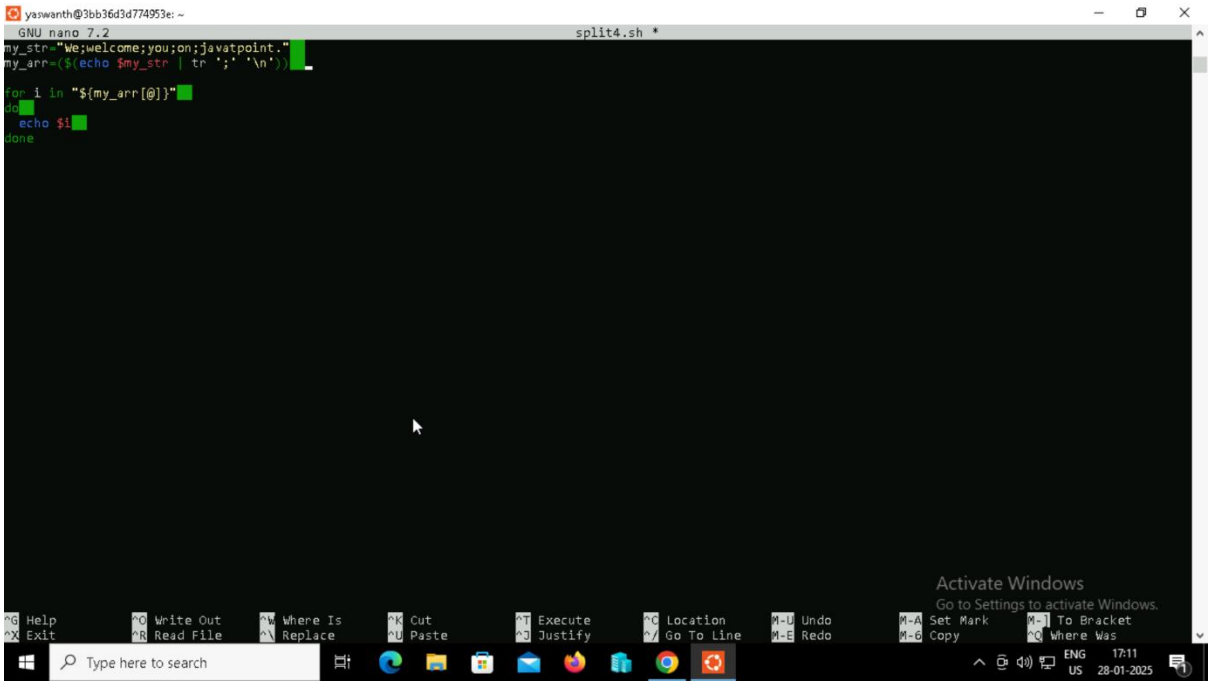
23. we have used `**awk**` command to find the length of a string.



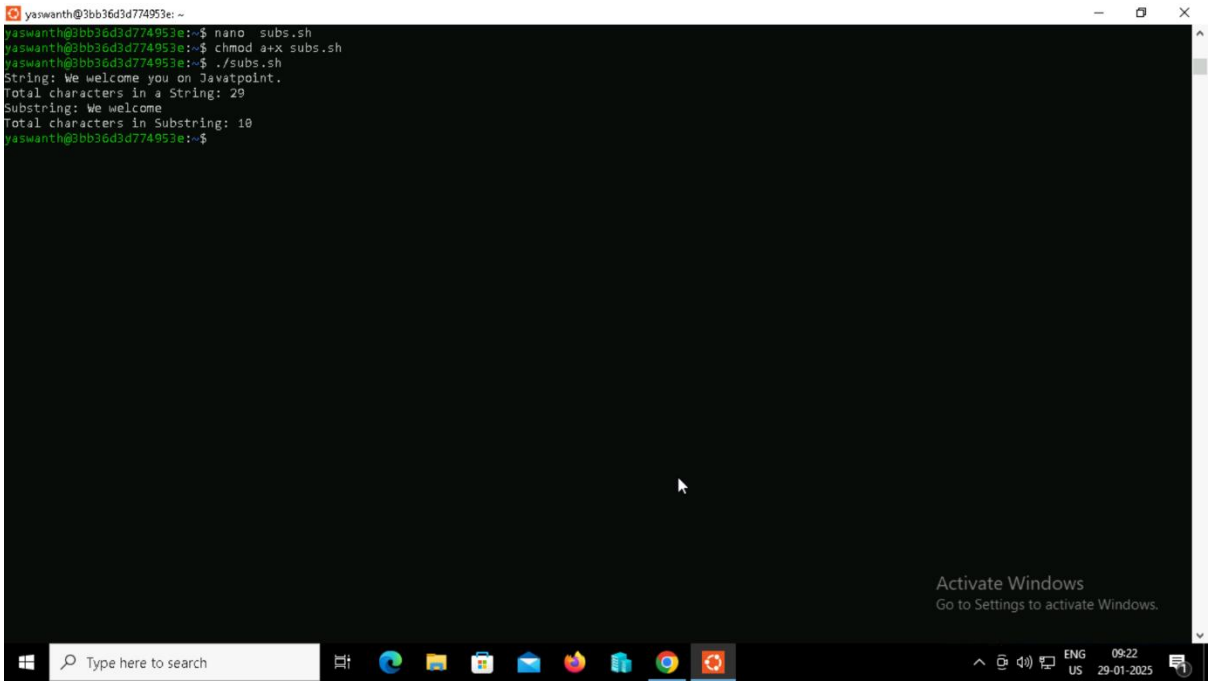


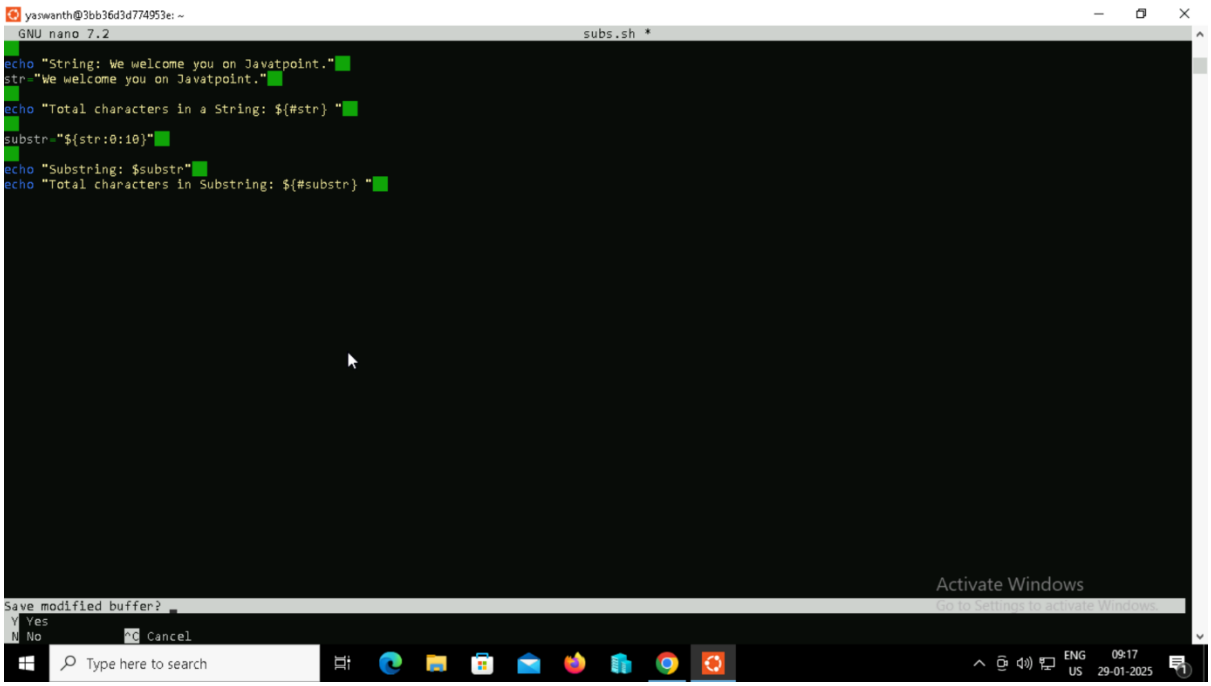
24. Bash **Split** String using Trim Command





25. To Extract till Specific Characters from Starting :**SUB STRING**

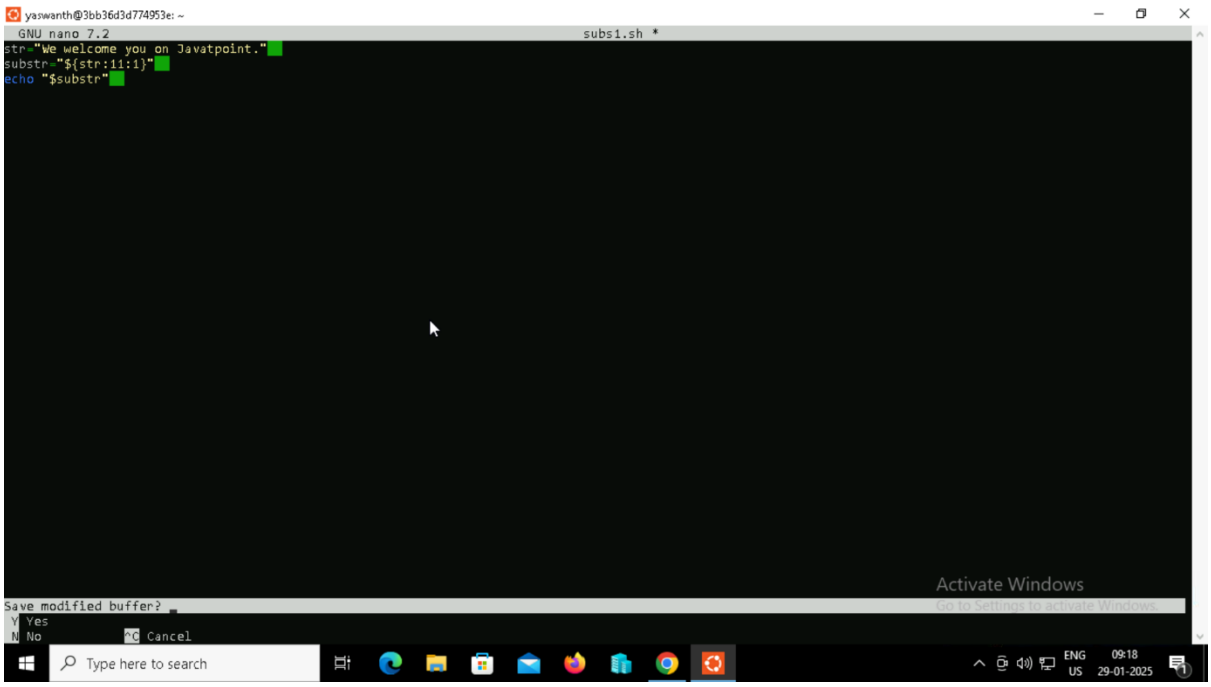




26 . To Extract a Single Character

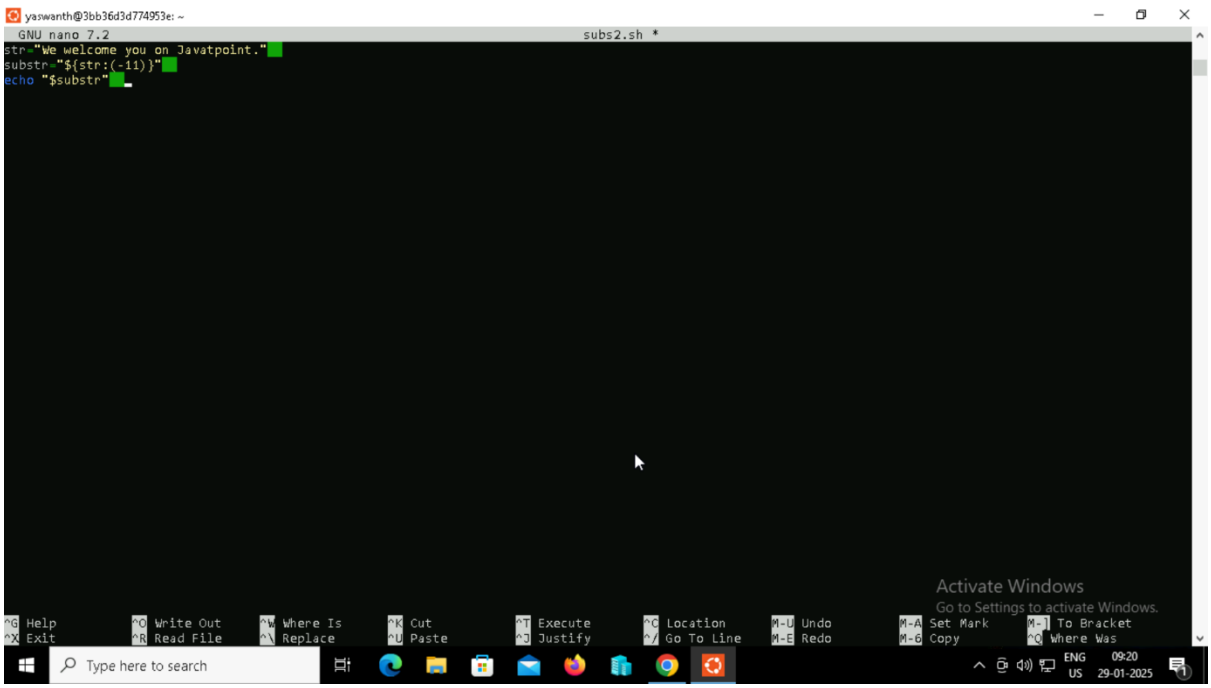
```
aswanth@3bb36d3d774953e:~$ nano  subs1.sh
aswanth@3bb36d3d774953e:~$ chmod a+x subs1.sh
aswanth@3bb36d3d774953e:~$ ./subs1.sh

aswanth@3bb36d3d774953e:~$ cat subs1.sh
tr="We welcome you on Javatpoint."
ubstr="${str:11:1}"
cho "$substr"
aswanth@3bb36d3d774953e:~$
```

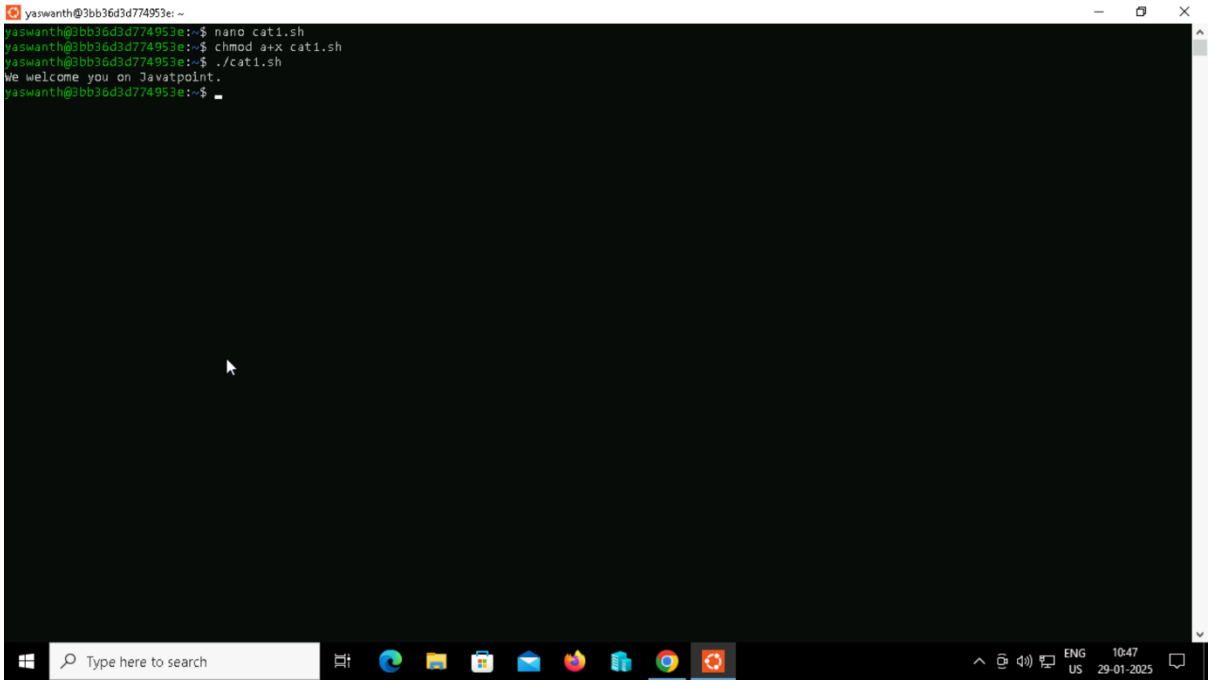


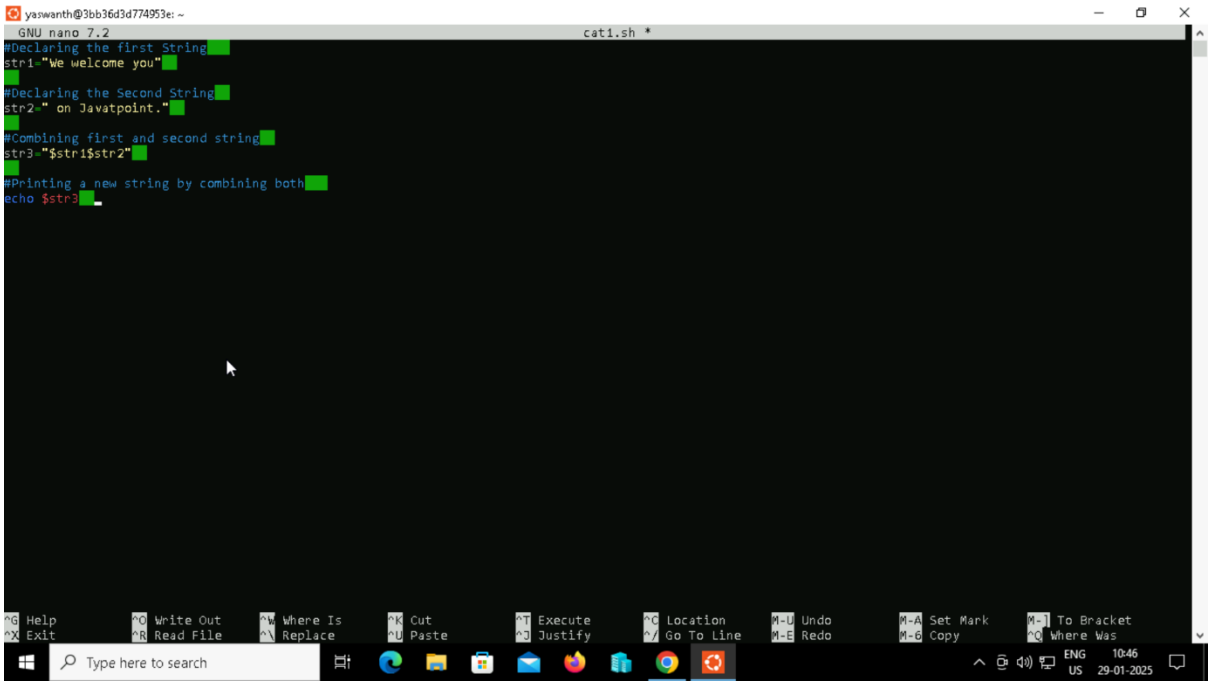
27. To Extract the specific characters from last

```
yaswanth@3bb36d3d774953e:~$ nano  subs2.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x subs2.sh
yaswanth@3bb36d3d774953e:~$ cat subs2.sh
str="We welcome you on Javatpoint."
substr="${str:(-11)}"
echo "$substr"
yaswanth@3bb36d3d774953e:~$ ./subs2.sh
Javatpoint.
yaswanth@3bb36d3d774953e:~$
```



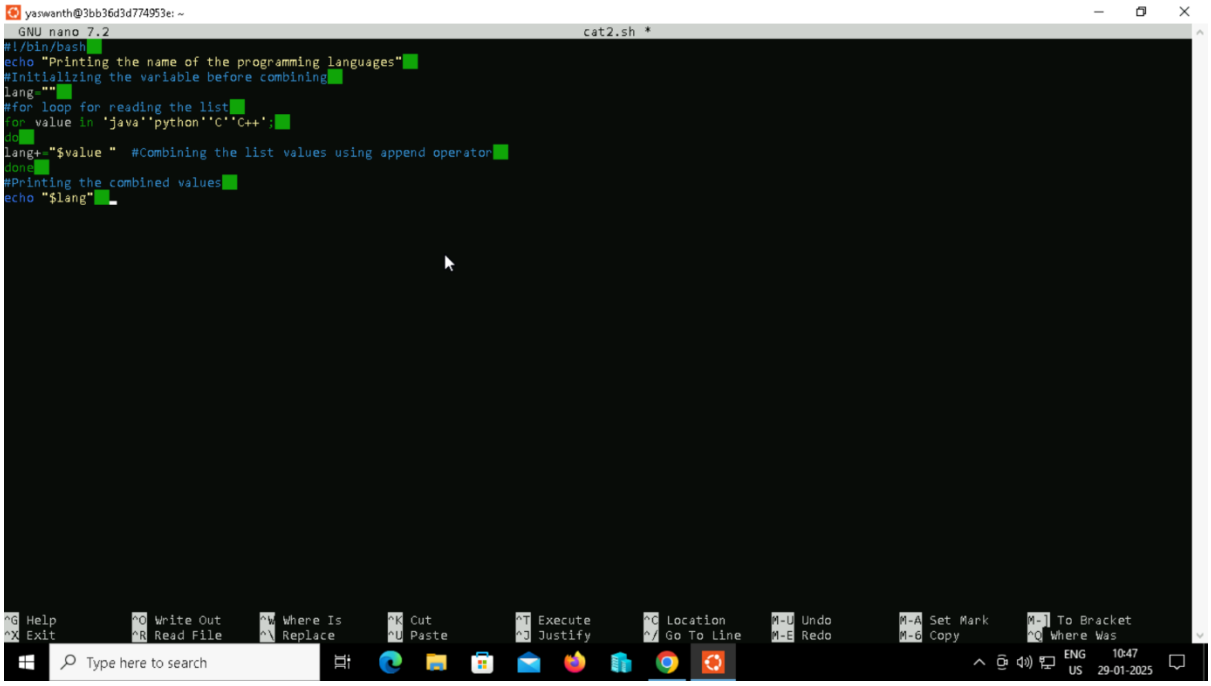
28. The basic example of String Concatenation, and we do not need any extra operator or function in this method.



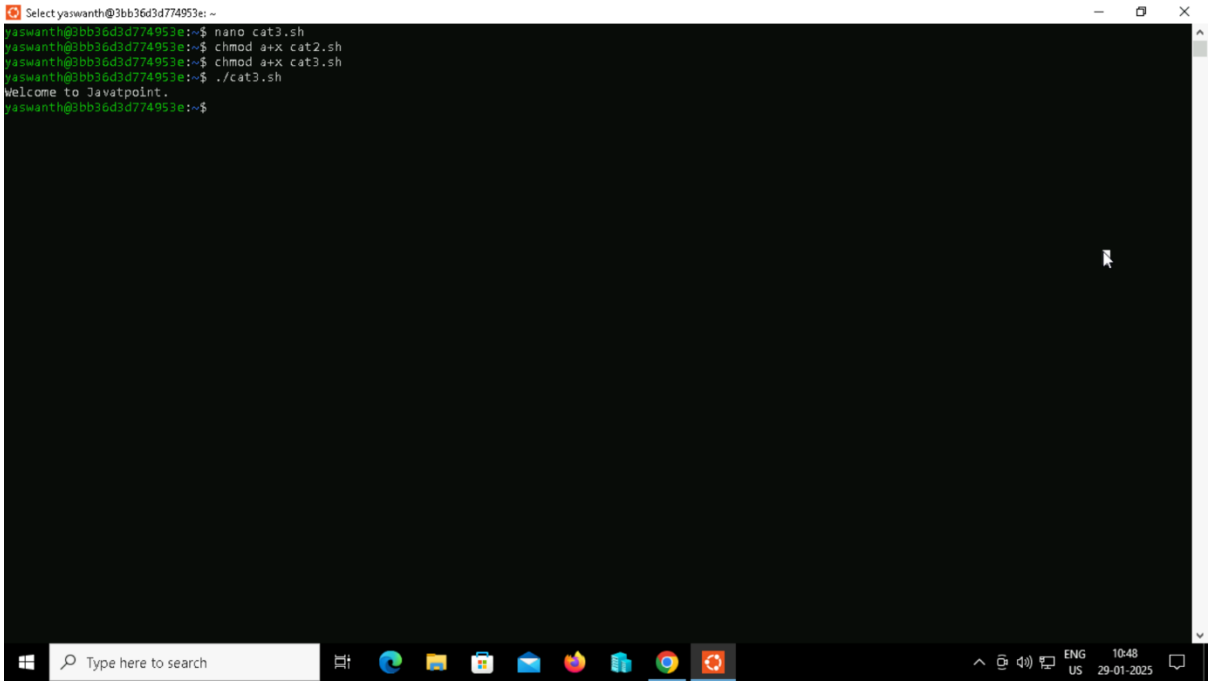


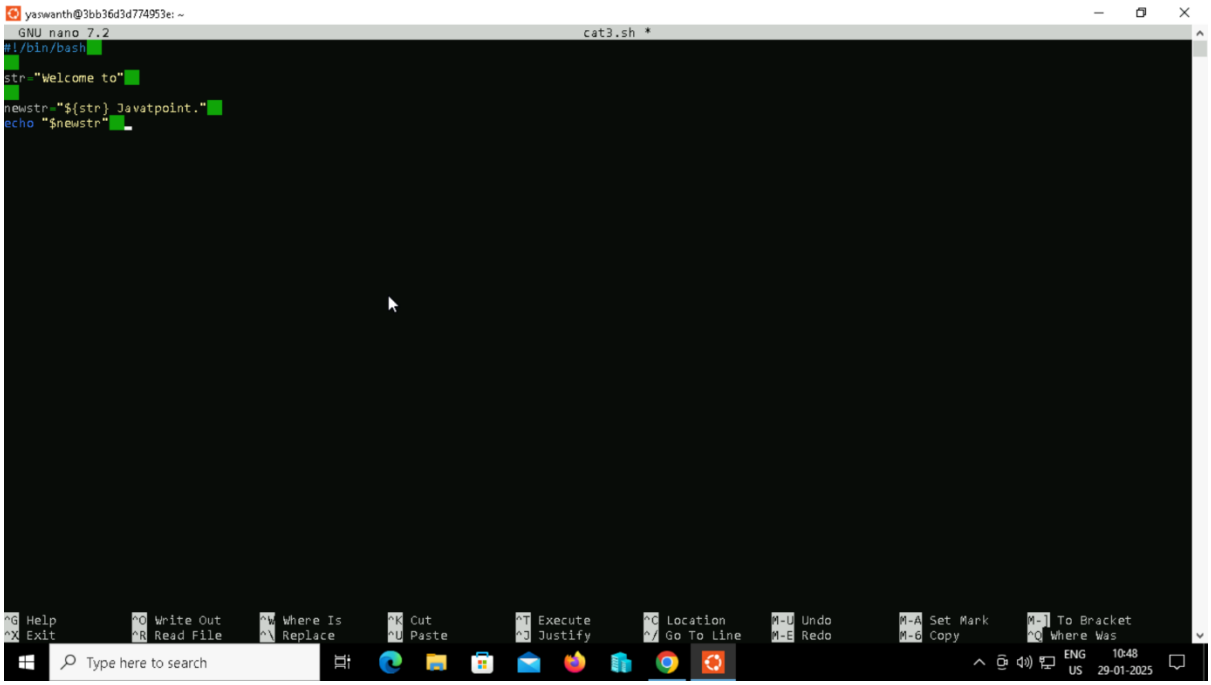
29. Using Append Operator with Loop

```
yaswanth@3bb36d3d774953e:~$ nano cat2.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x cat2.sh
yaswanth@3bb36d3d774953e:~$ ./cat2.sh
Printing the name of the programming languages
javapythonCC++
yaswanth@3bb36d3d774953e:~$
```

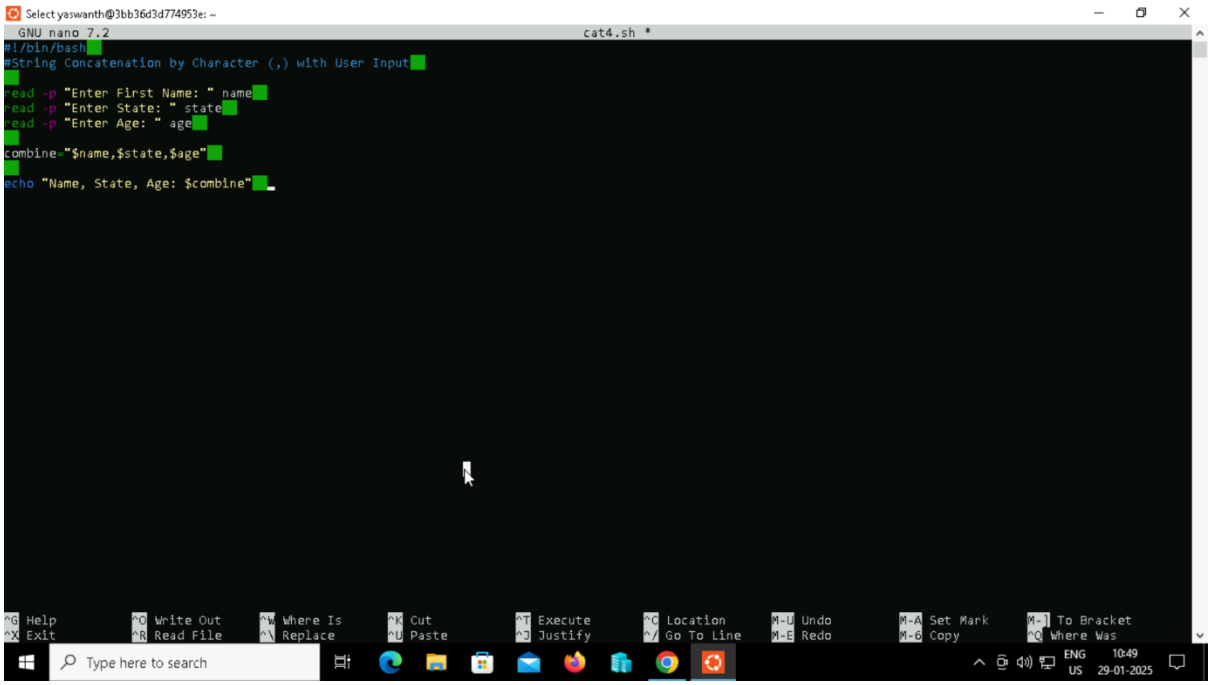


30. Using Literal Strings

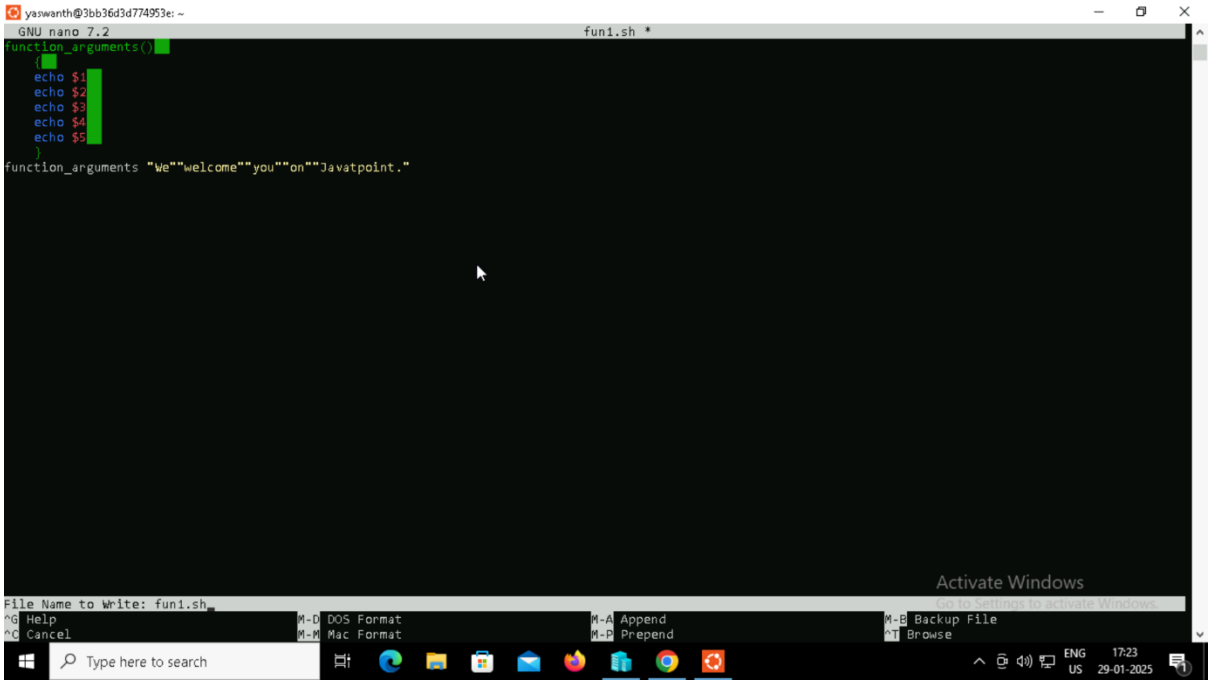


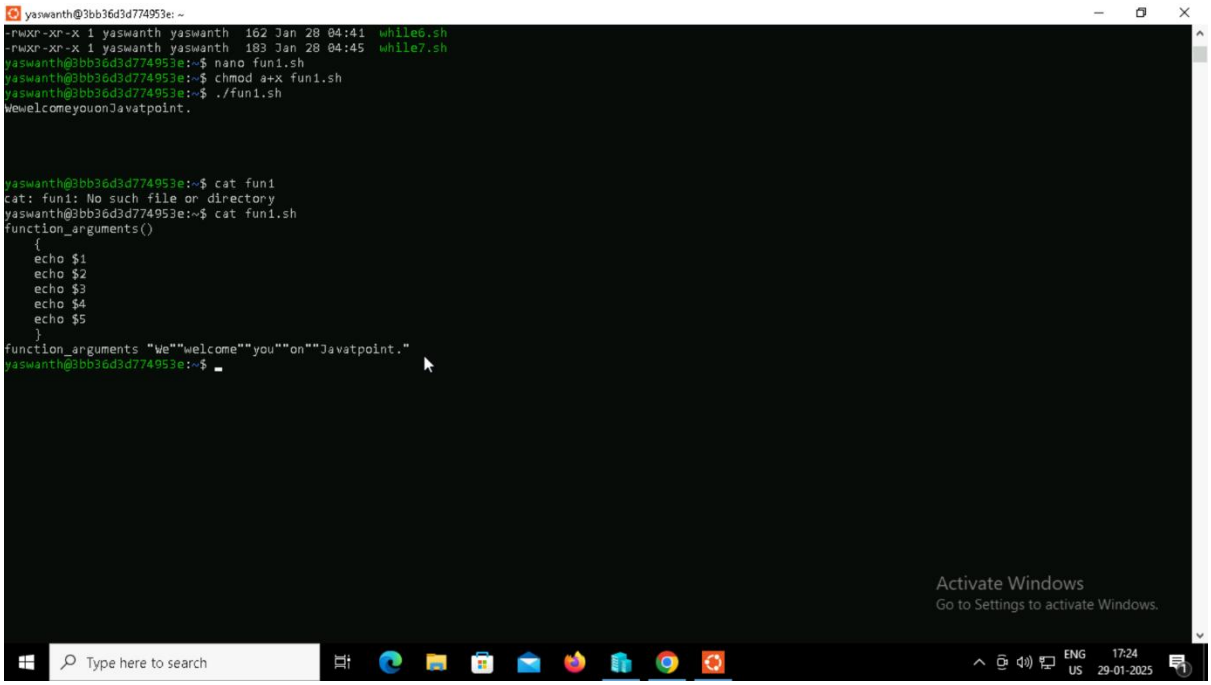


31. Using any Character

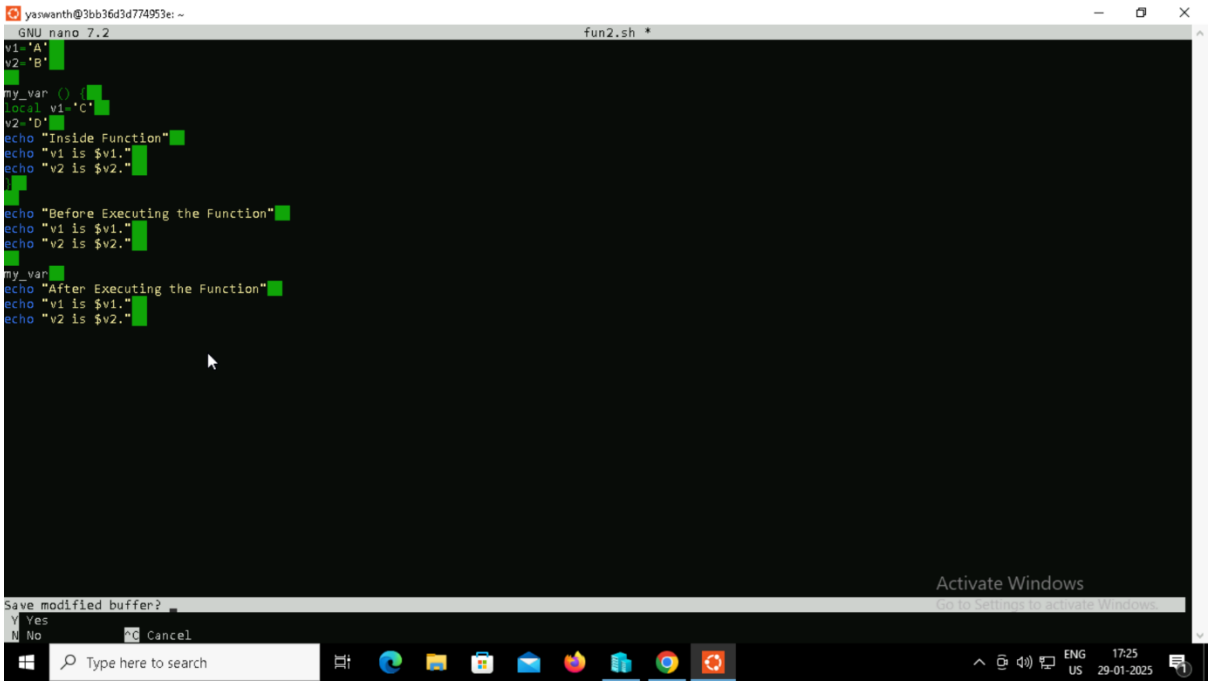


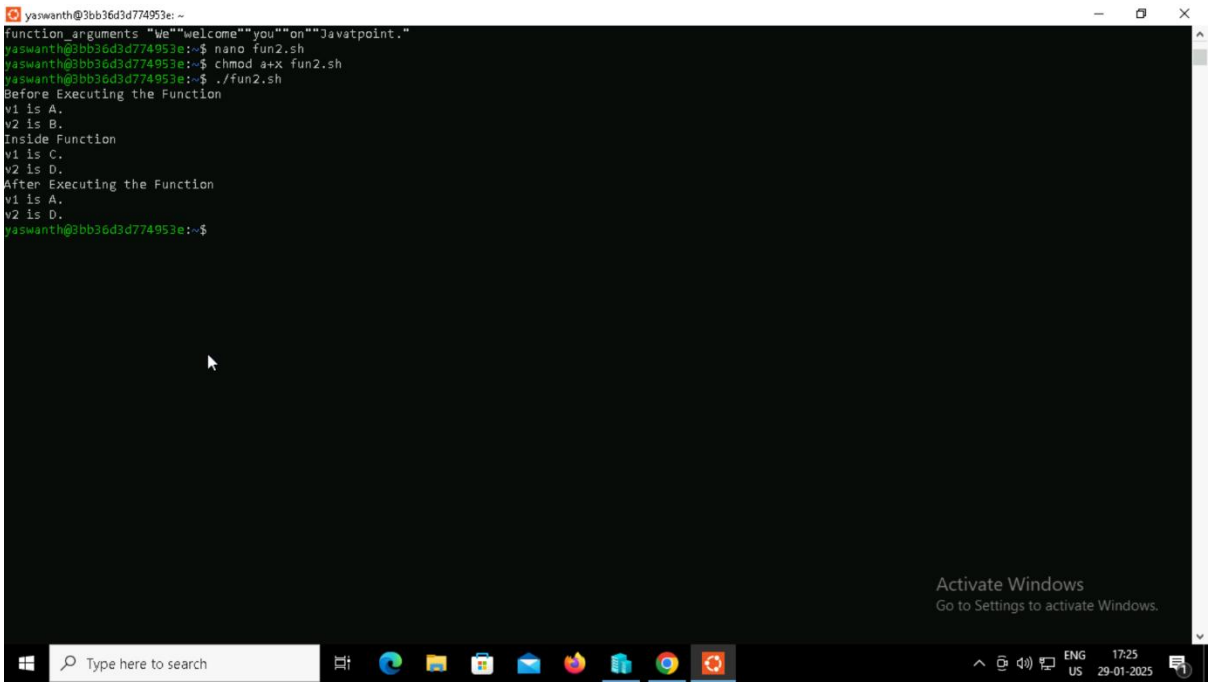
32. code that illustrates the procedure on how to pass arguments to functions, and access the arguments inside the function.



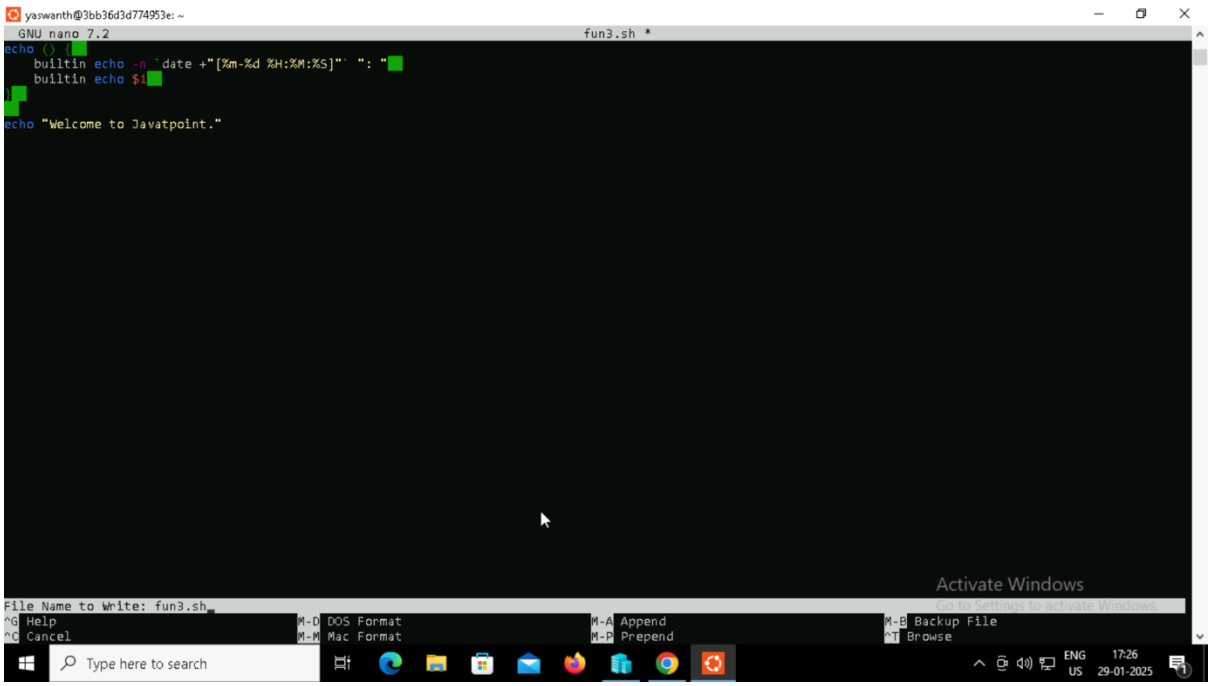


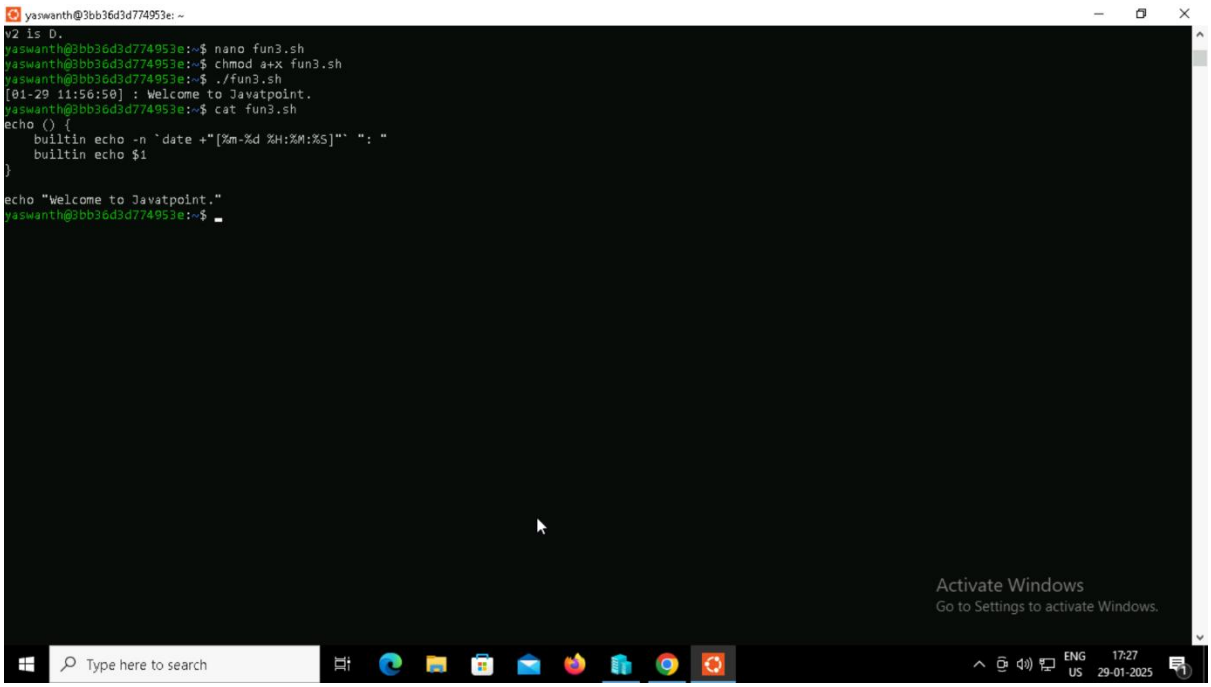
33. To understand how variables scope works in Bash Scripting



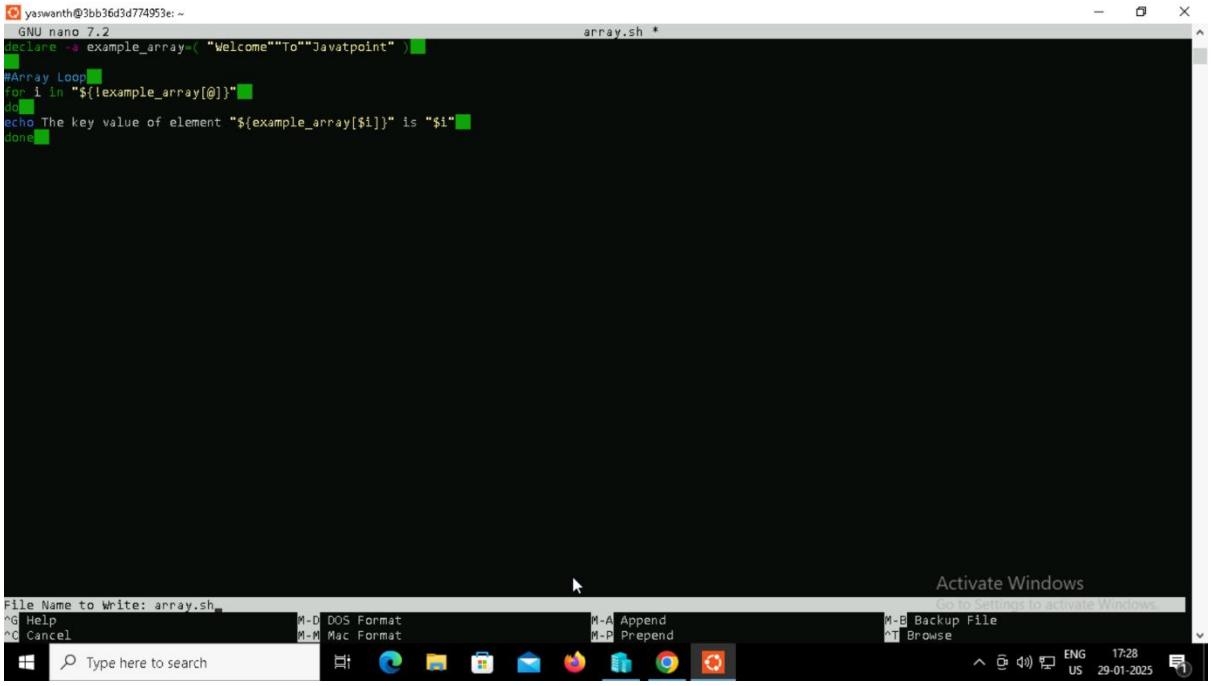


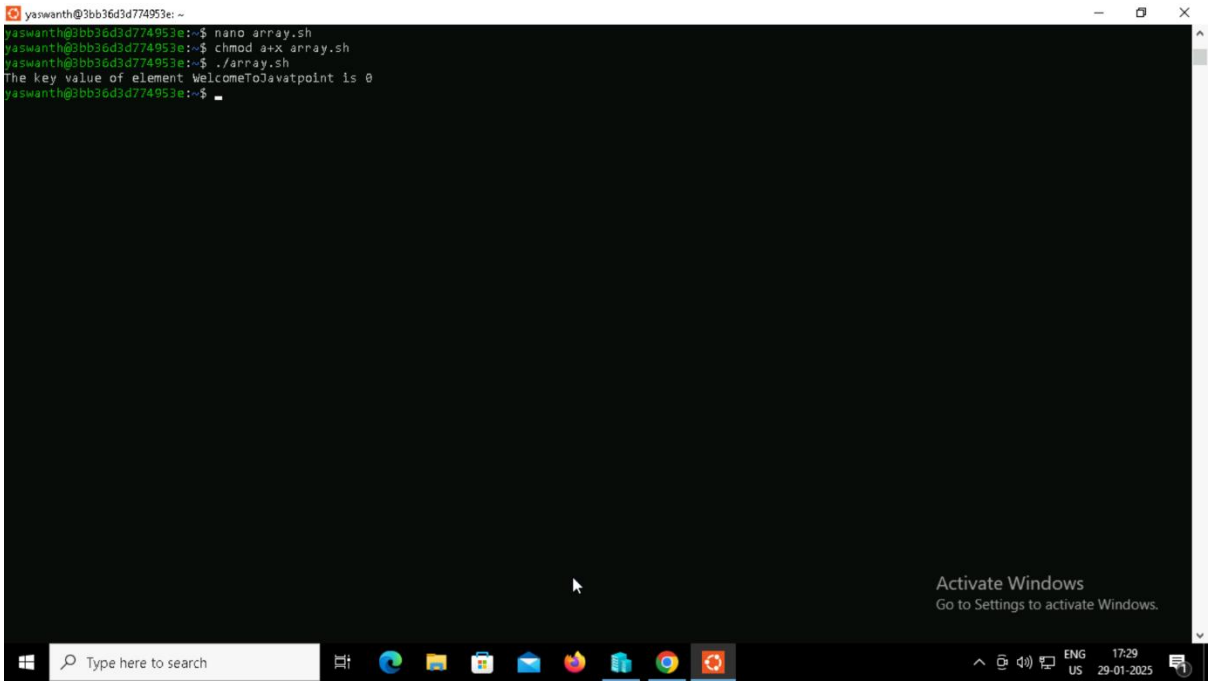
34. overridden the 'echo' command and added the time stamp in the form of the argument to the 'echo' command



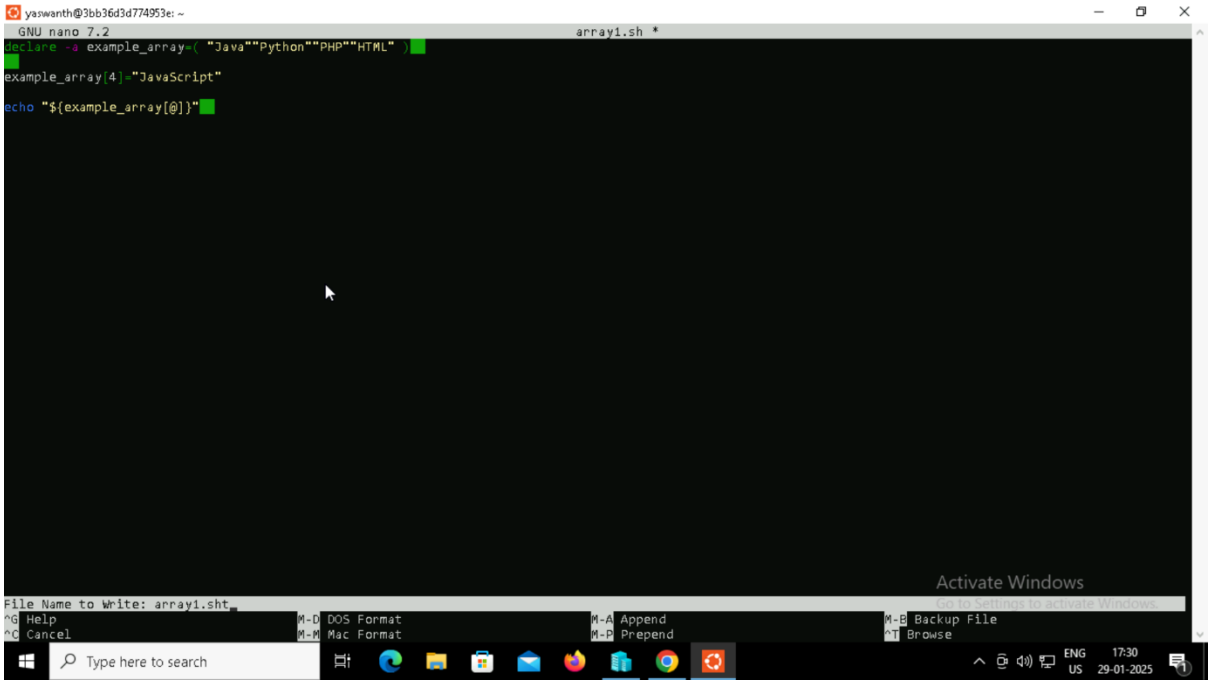


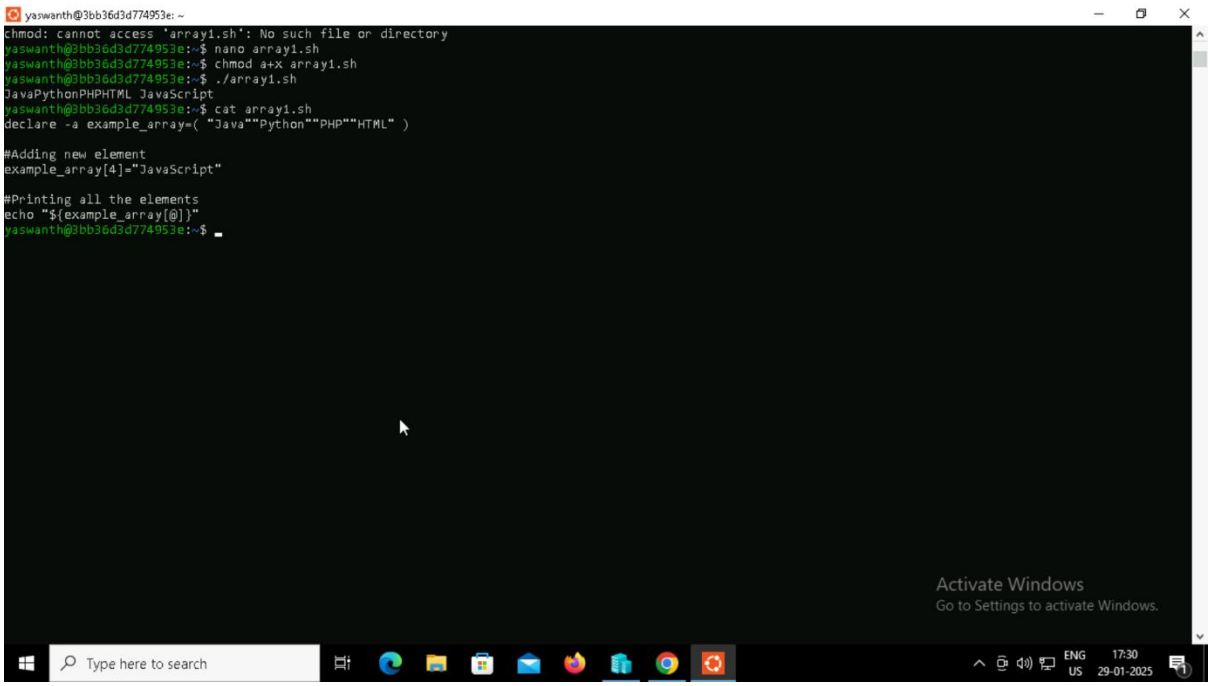
35. The general method to iterate over each item in an array is by using the 'for loop'.





36 . Adding Elements to an Array





37 . Deleting the Entire Array