**Bash Project**

**Bash Case**

The **Bash case** statement is the simplest form of IF-THEN-ELSE with many ELIF elements. Using the case statement makes our bash script more readable and easier to maintain. These are generally applied to simplify the complex conditions having multiple different choices.

The Bash case statement follows a similar logic as the Javascript or C switch statement. There is a slight difference, as follows:

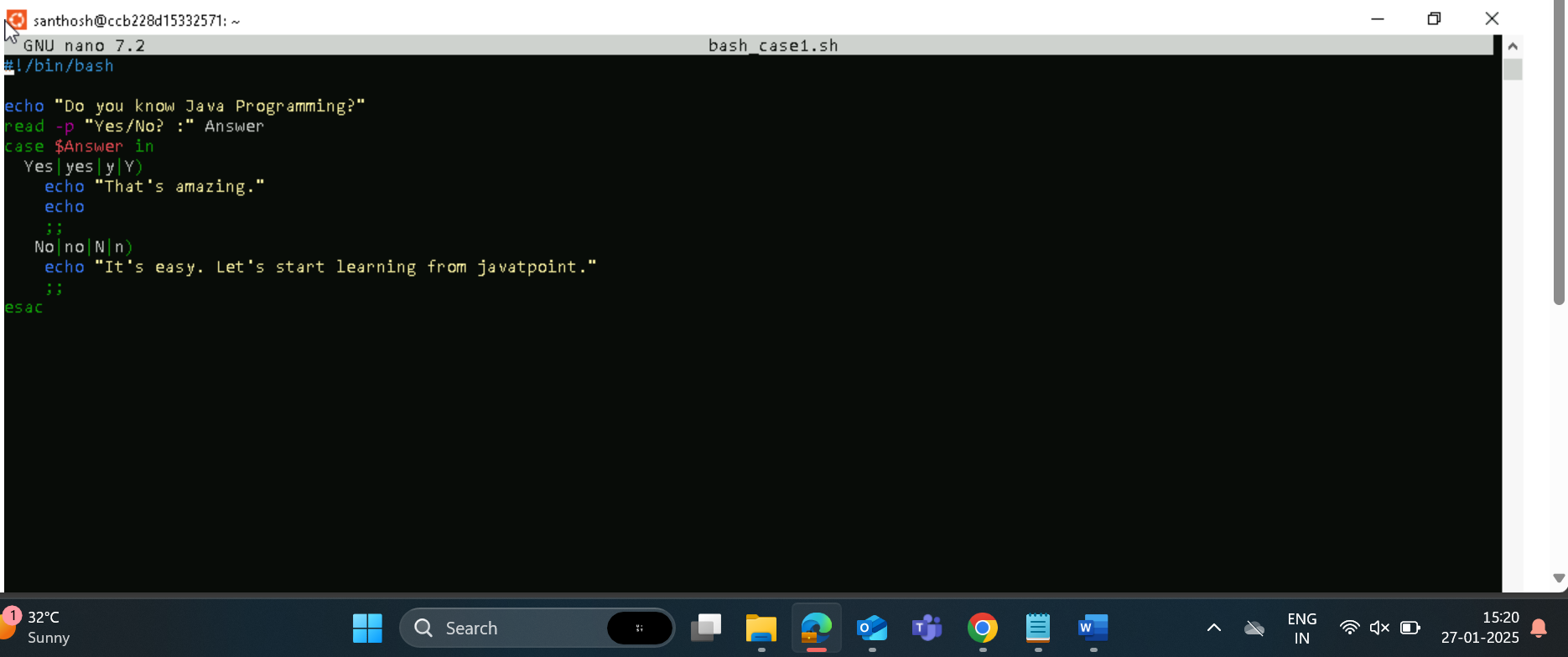
o The Bash case statement takes a value once and tests that value multiple times. It stops searching for a pattern once it has found it and executed the statement linked with it, which is almost opposite in case of the C switch statement.

**Example 1**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

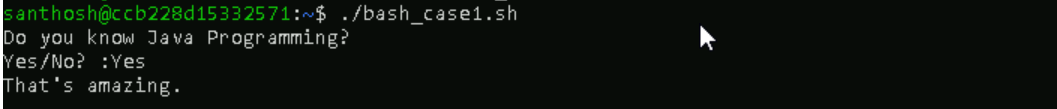
****

**Step 3**: Providing the necessary permissions for the base\_case1.sh script.

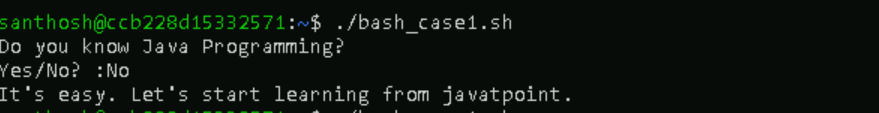
****

**Step 4:** Executing the output.

1. For Yes the output is.

****

1. For No the output is.

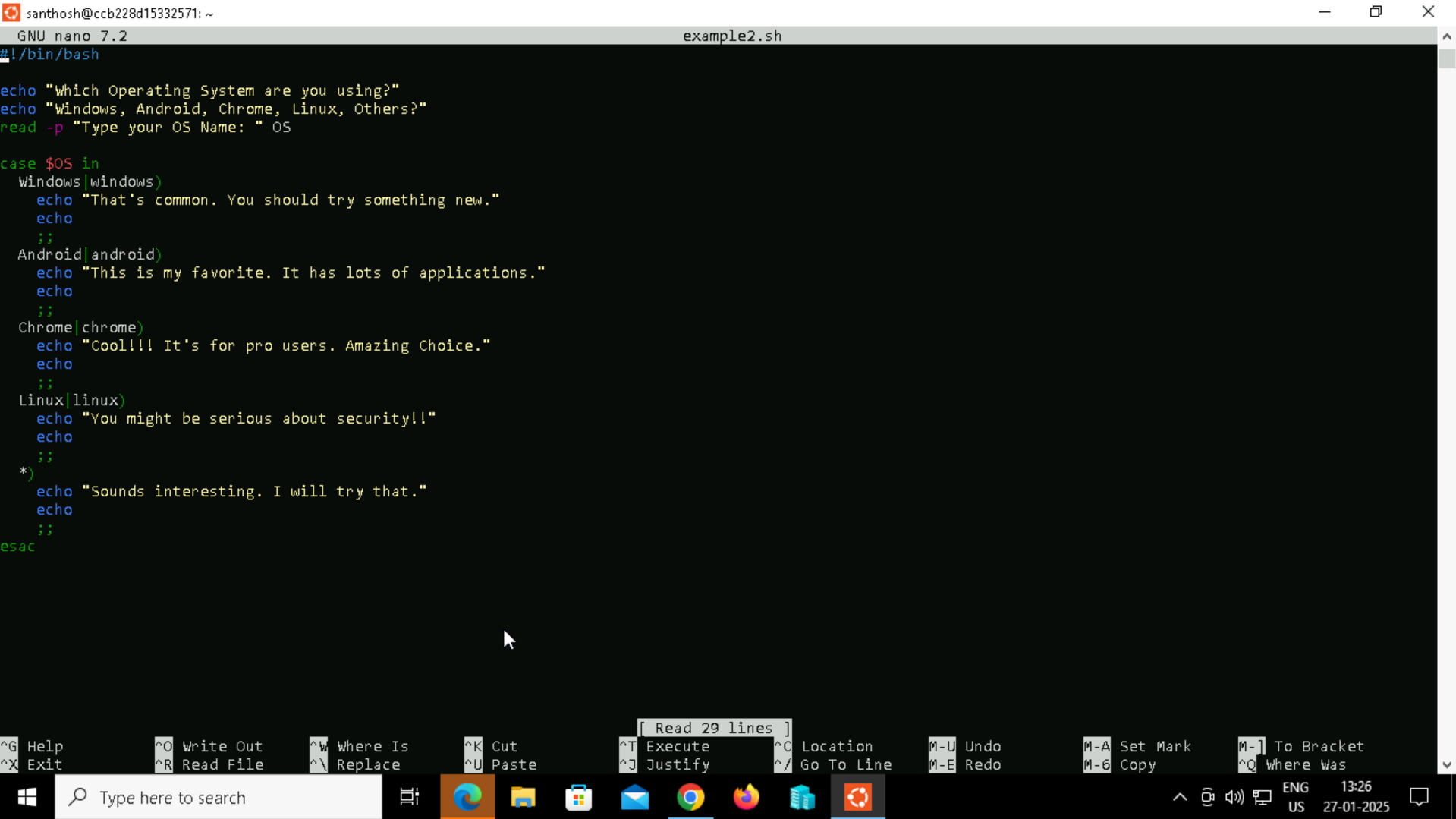
****

**Example 2**

A combined scenario where there is also a default case when no previous matched case is found.

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

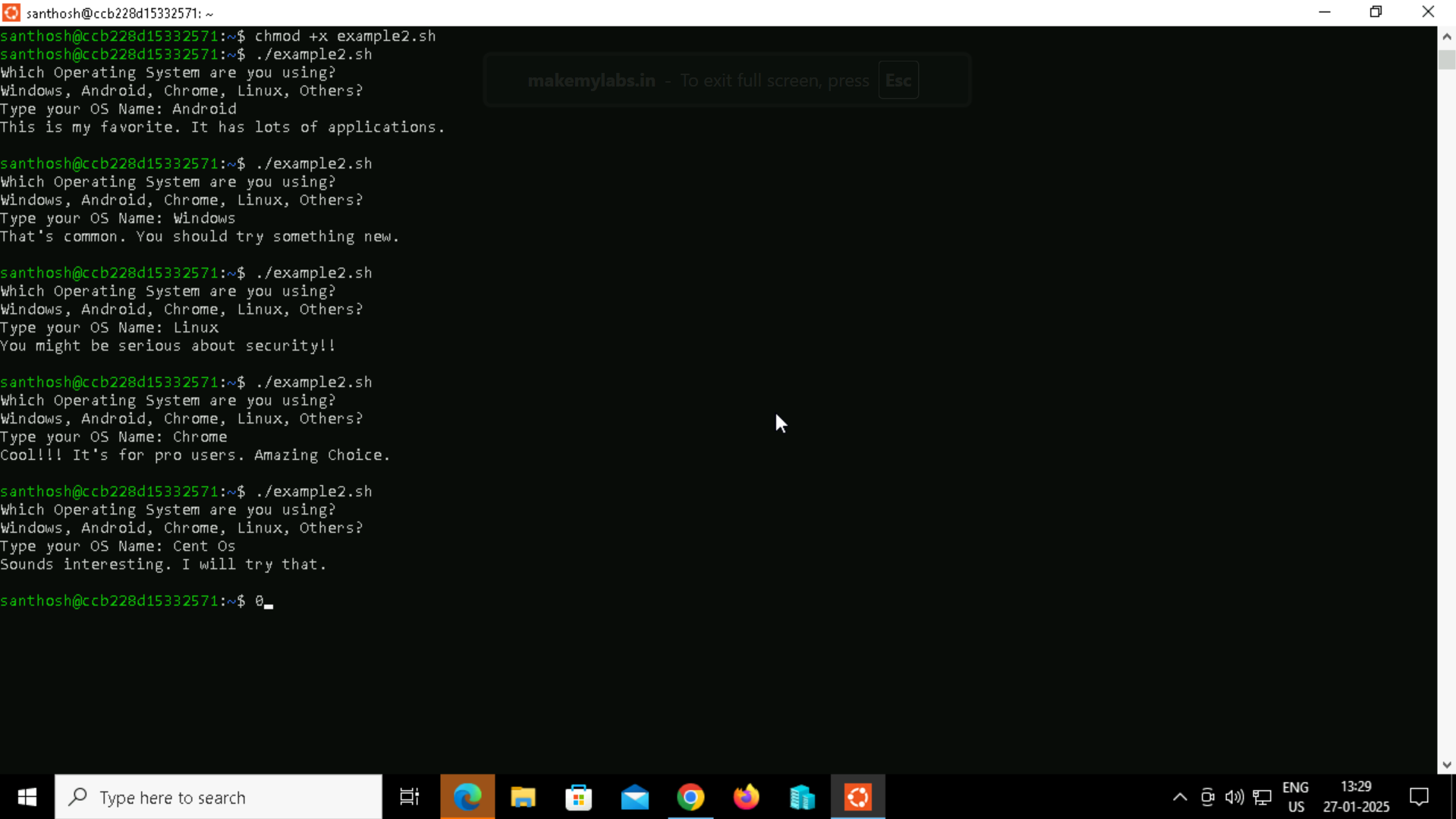
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3**: Providing the necessary permissions for the example2.sh script.



**Step 4:** Executing the output.

****

**Bash For Loop**

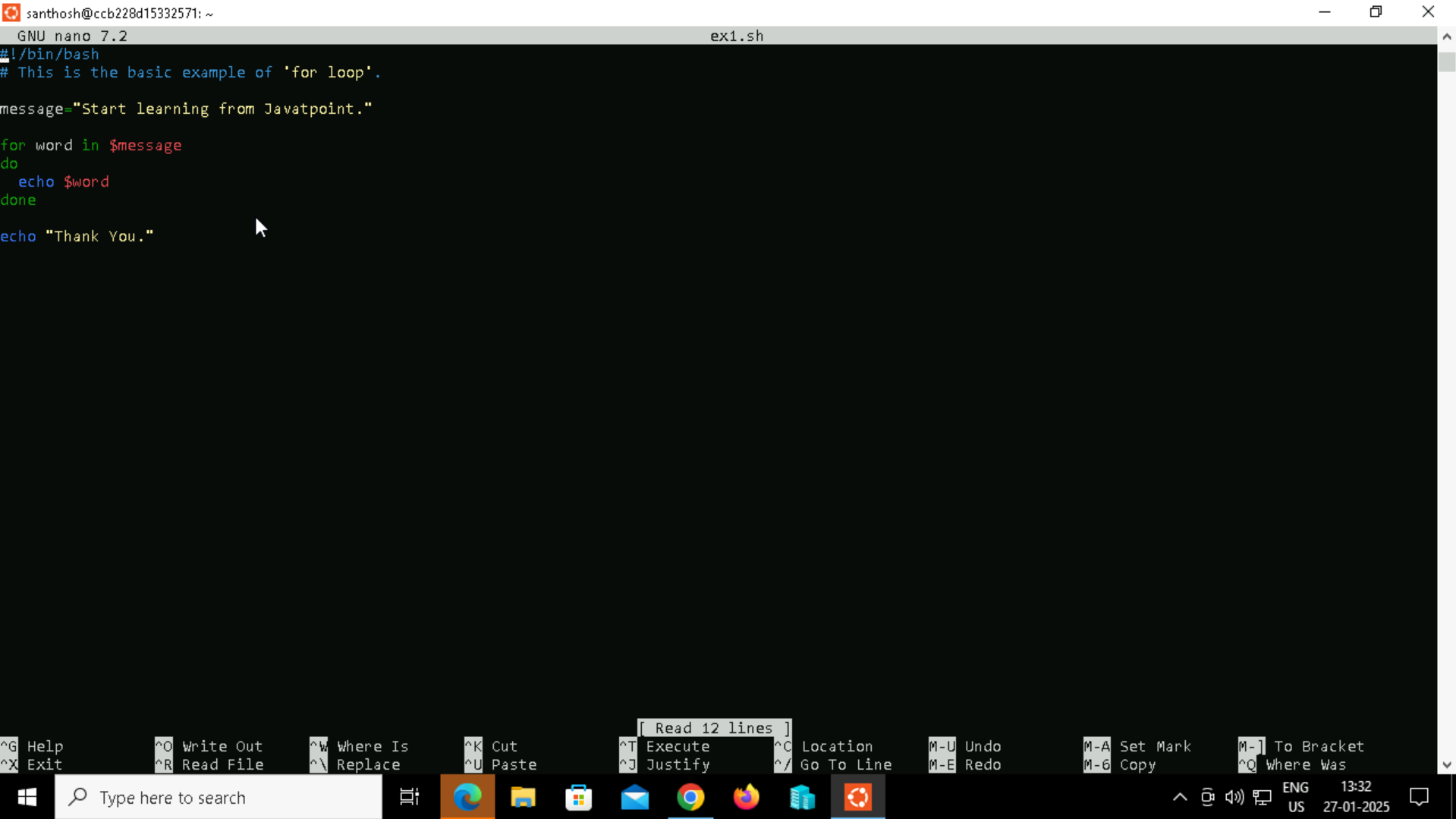
Like any other programming language, bash shell scripting also supports 'for loops' to perform repetitive tasks. It helps us to iterate a particular set of statements over a series of words in a string, or elements in an array. For example, you can either run UNIX command (or task) many times or just read and process the list of commands using a 'for loop'.

**Example 1**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



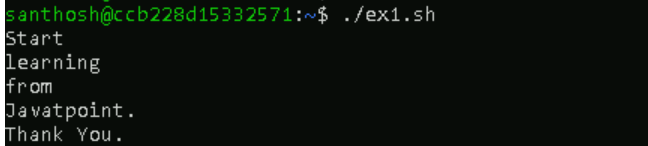
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3**: Providing the necessary permissions for the ex1.sh script.



**Step 4:** Executing the output.



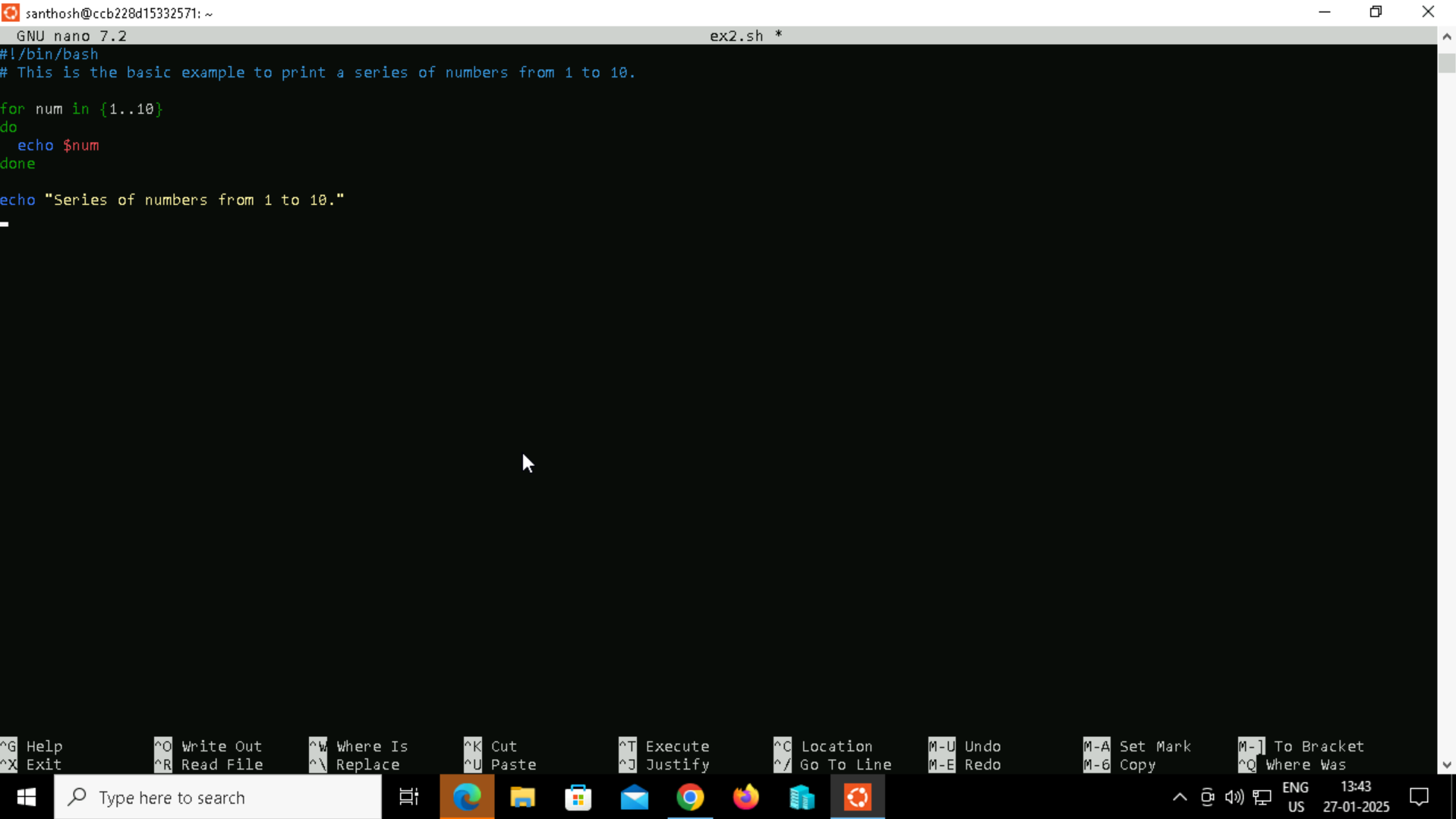
**Example 2**

**For Loop to Read a Range**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3**: Providing the necessary permissions for the ex2.sh script.

****

**Step 4:** Executing the output.



**Example 3**

**For Loop to Read a Range with Increment/Decrement**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex3.sh script.

****

**Step 4:** Executing the output.

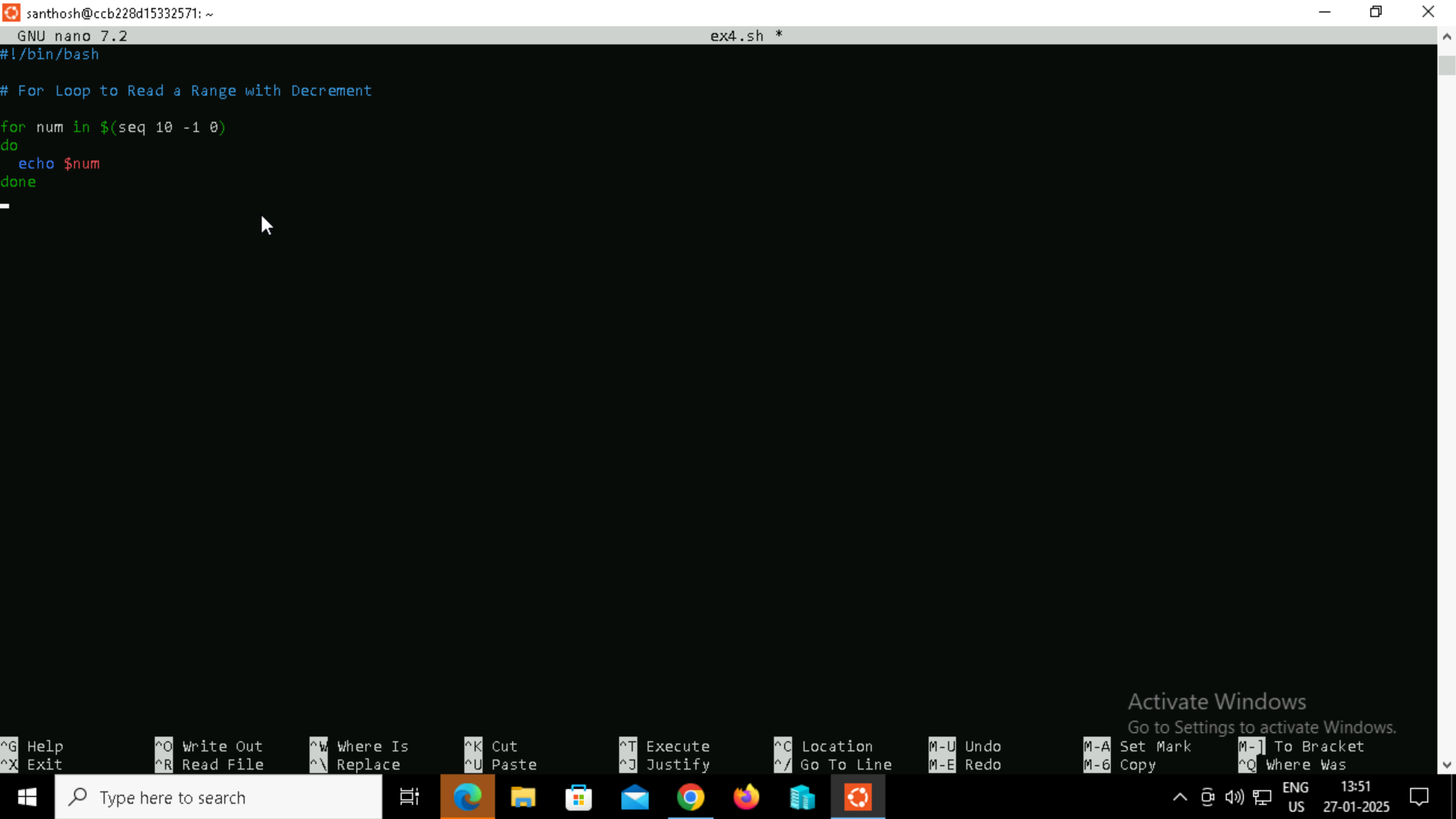
****

**Example 4**

**For Decrement**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

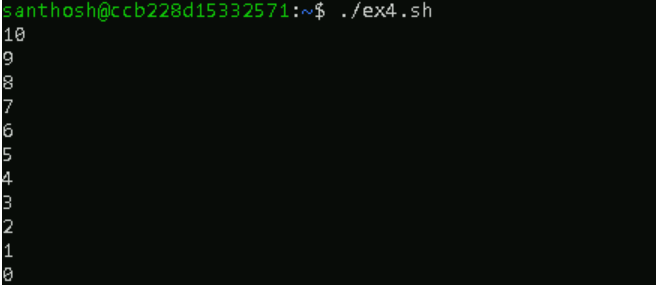
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex4.sh script.



**Step 4:** Executing the output.

****

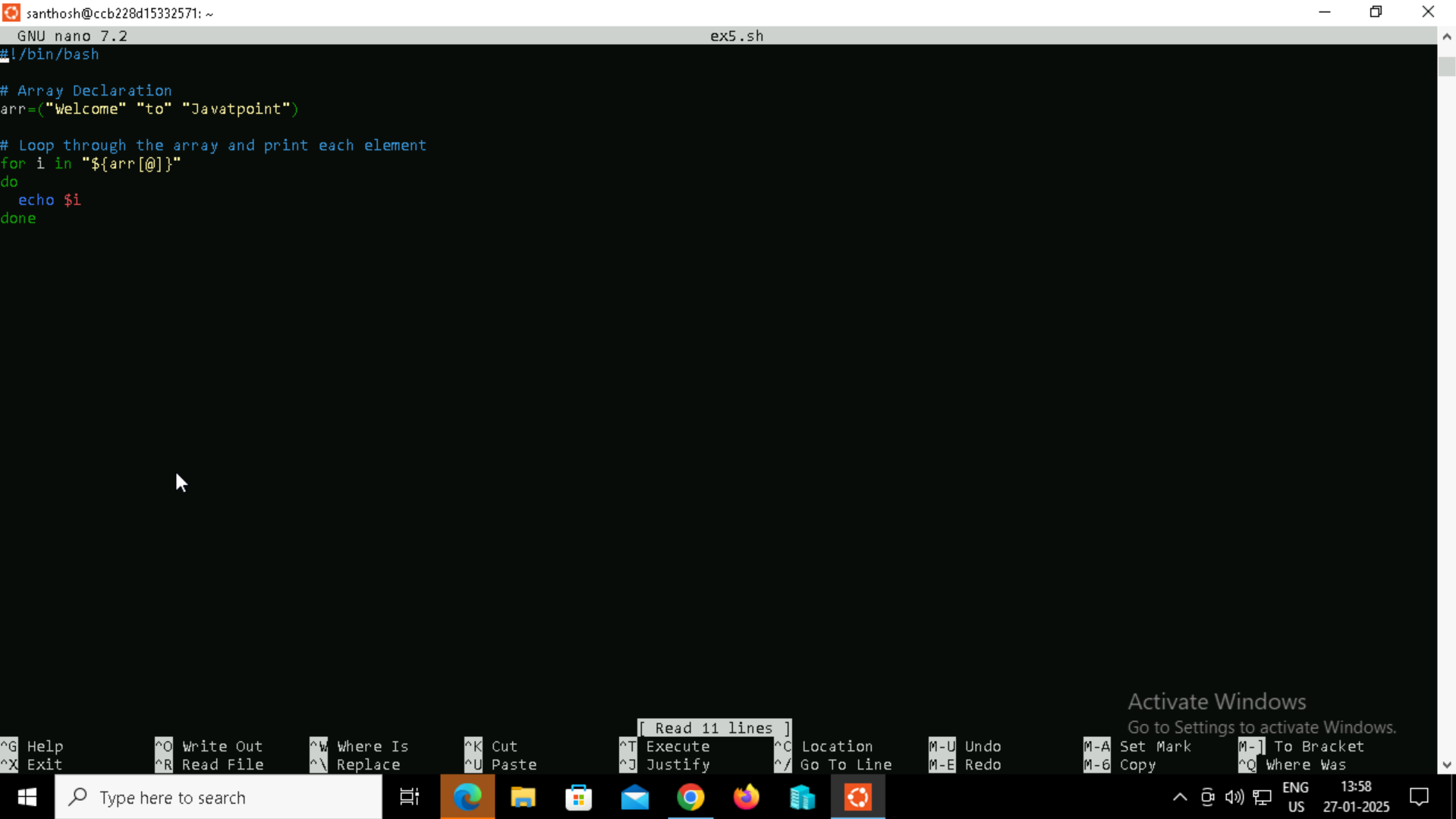
**Example 5**

**For Loop to Read Array Variables**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



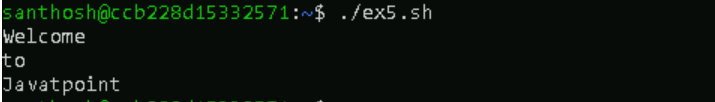
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement**.**

****

**Step 3:** Providing the necessary permissions for the ex5.sh script.

****

**Step 4:** Executing the output.

****

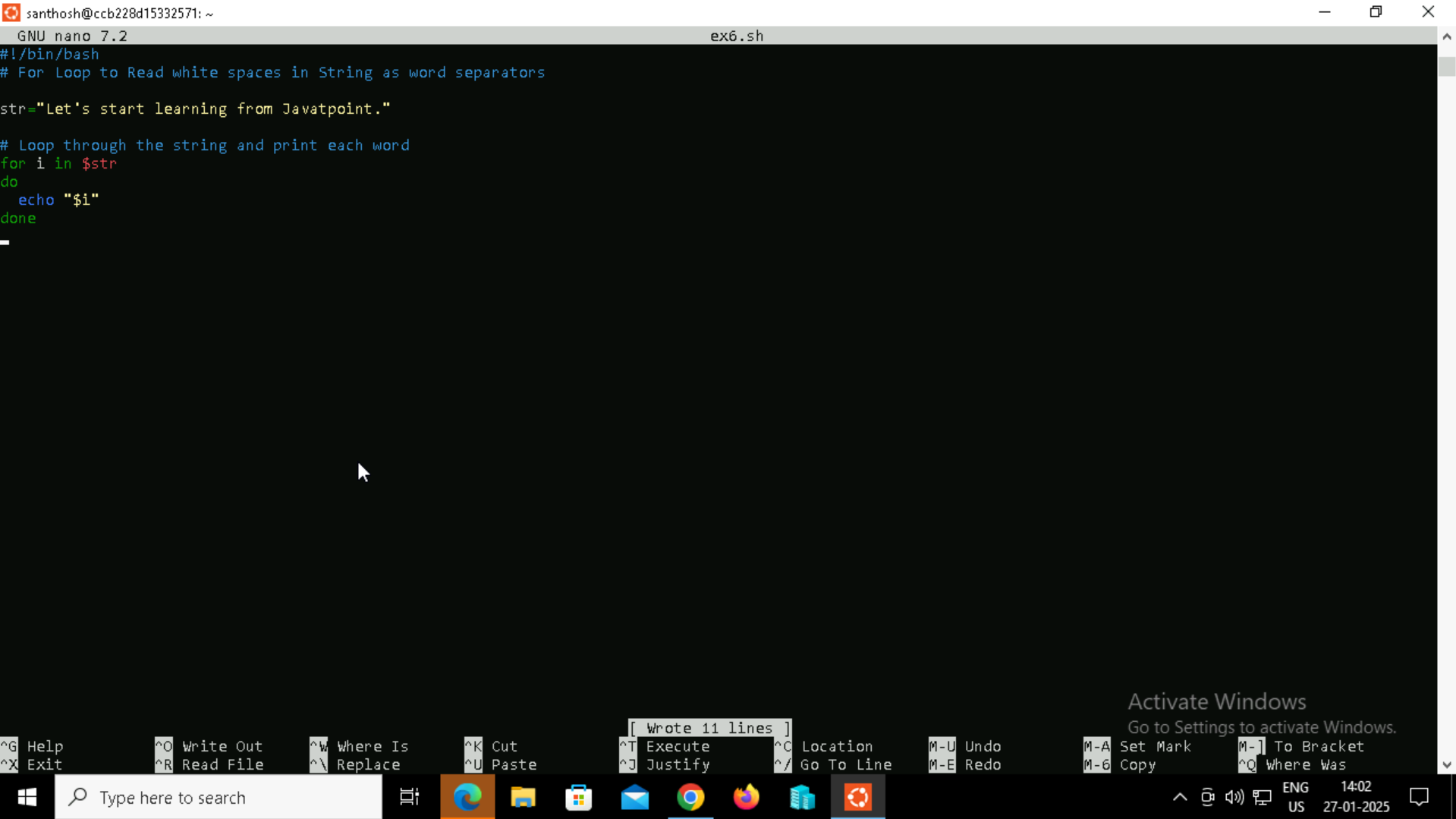
**Example 6**

**For Loop to Read white spaces in String as word separators.**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex6.sh script.



**Step 4:** Executing the output.



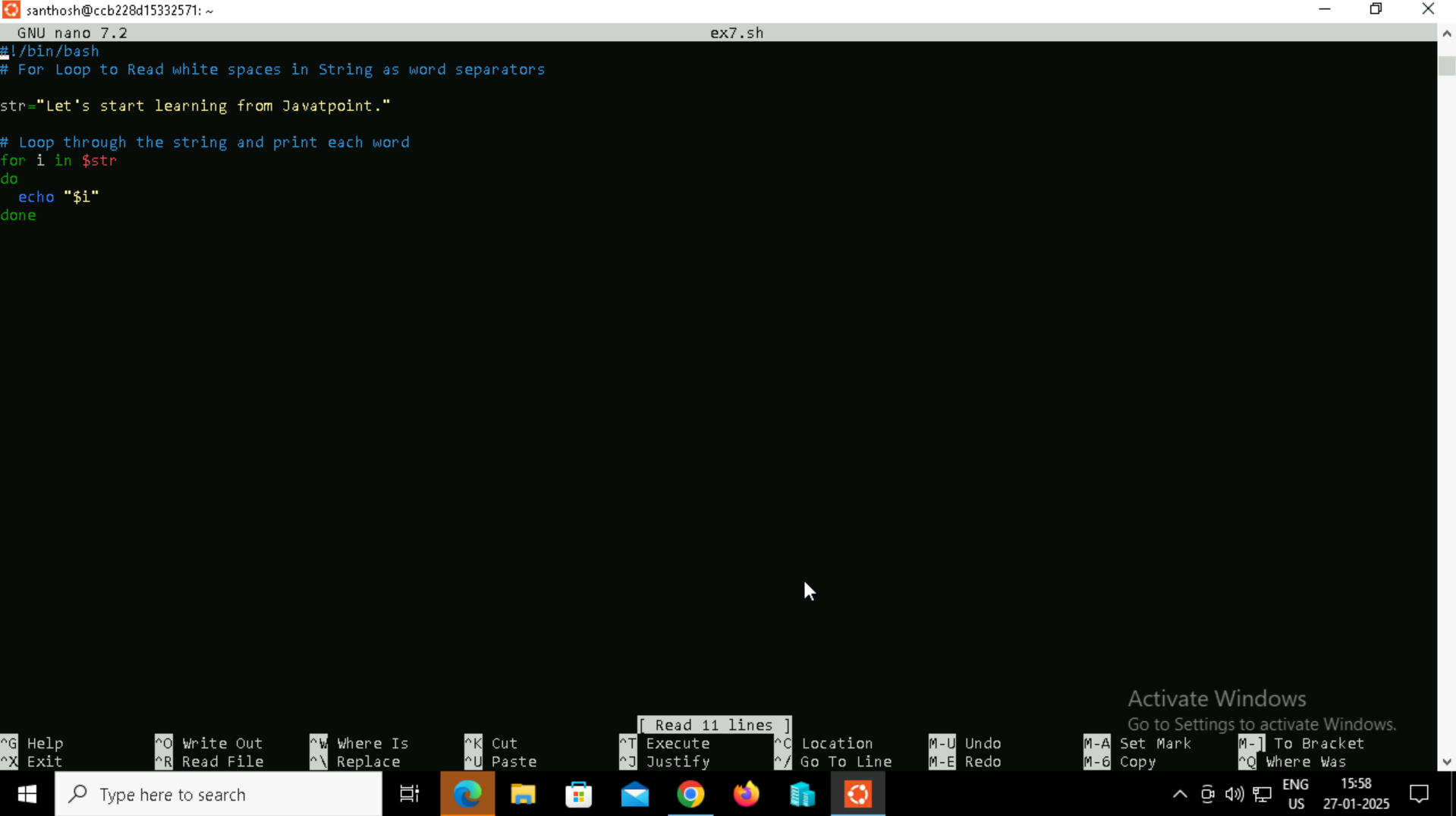
**Example 7**

**For Loop to Read each line in String as a word.**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



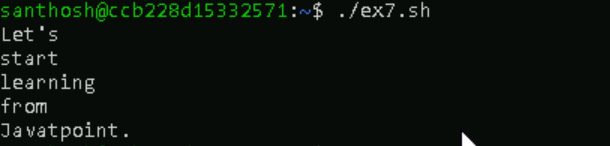
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex7.sh script.



**Step 4:** Executing the output.

****

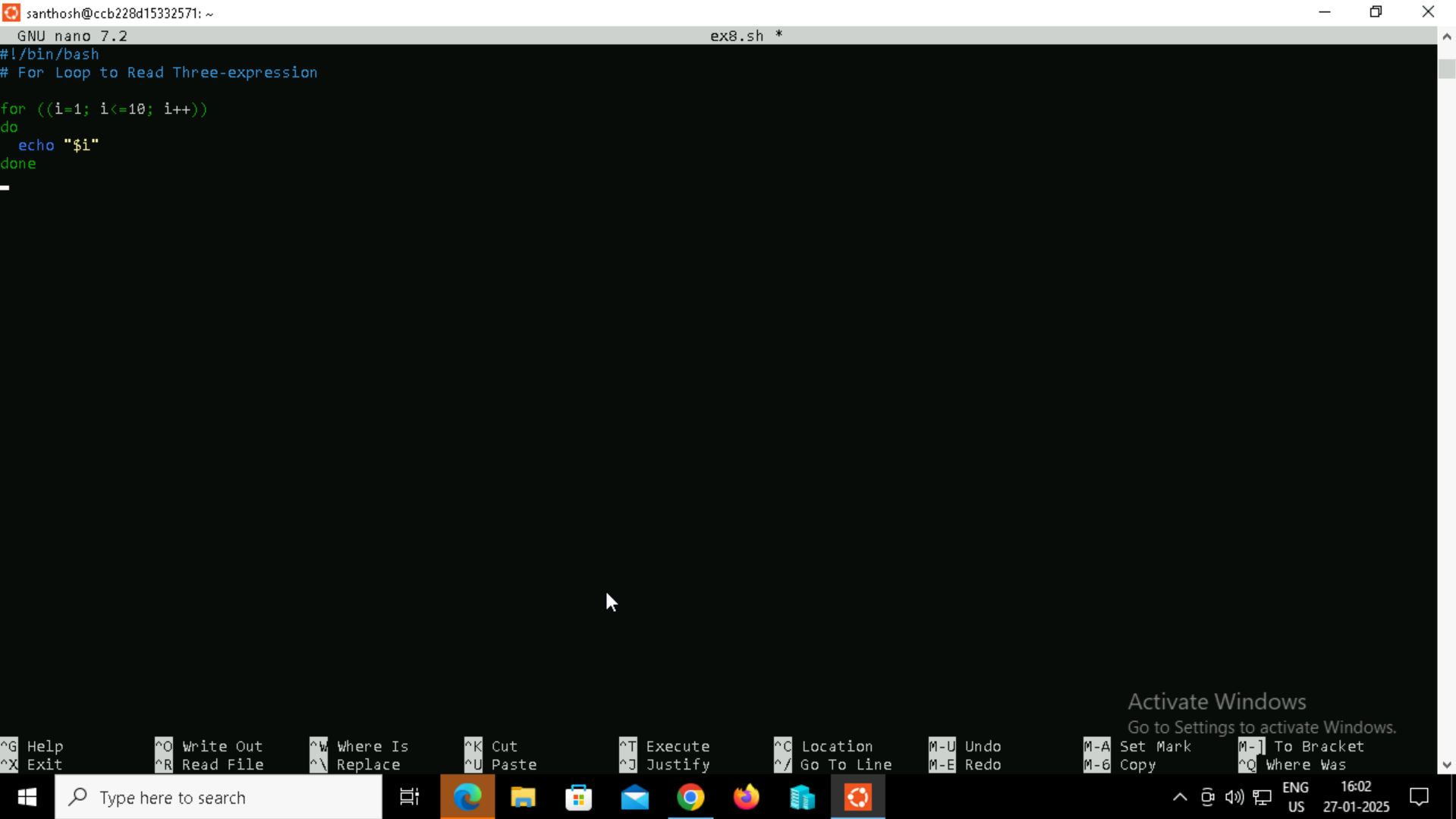
**Example 8**

**For Loop to Read Three-expression**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex8.sh script.



**Step 4:** Executing the output.

****

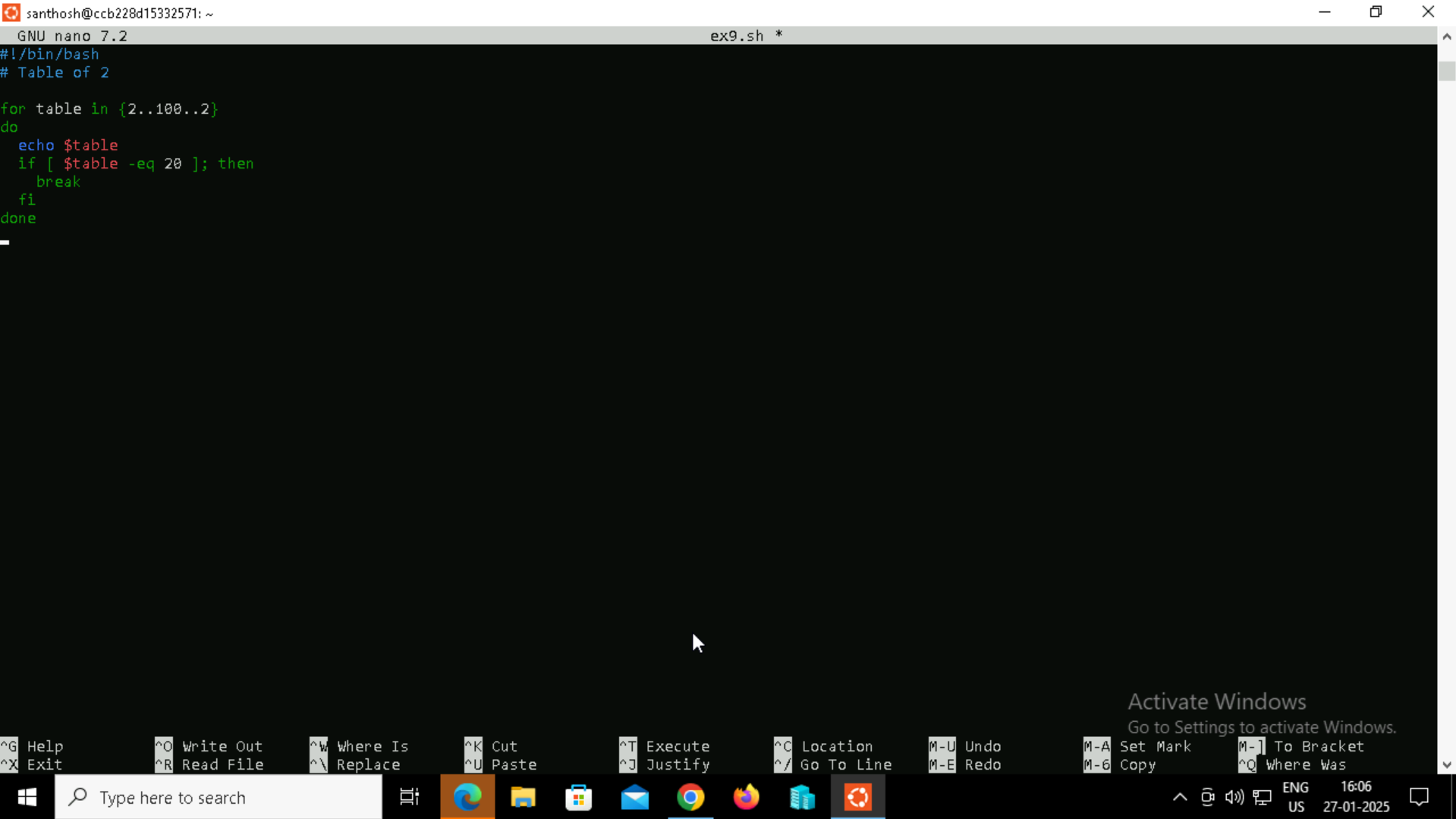
**Example 9**

**For Loop with a Break Statement**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

****

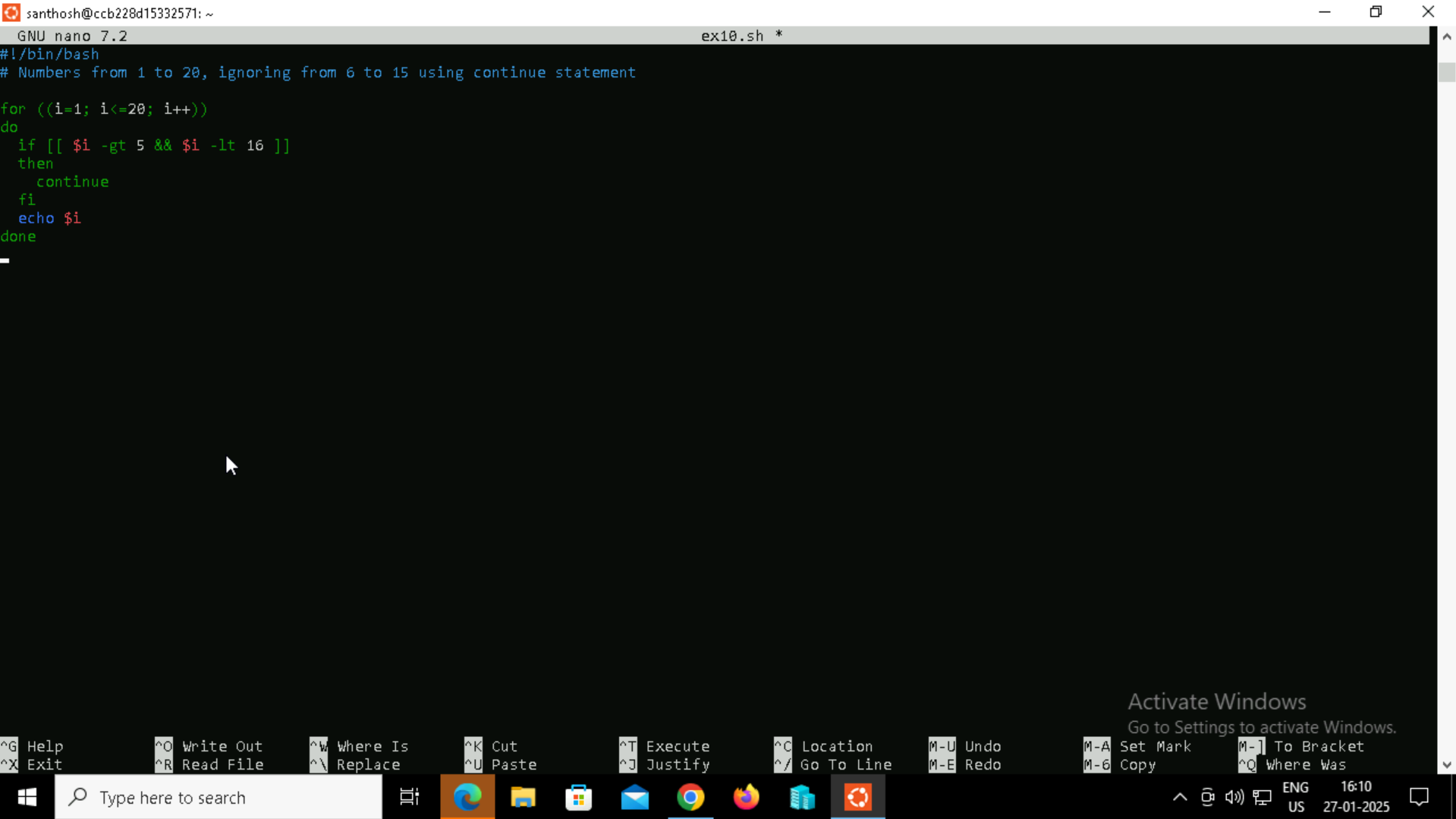
**Example 10**

**For Loop with a Continue Statement**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

****

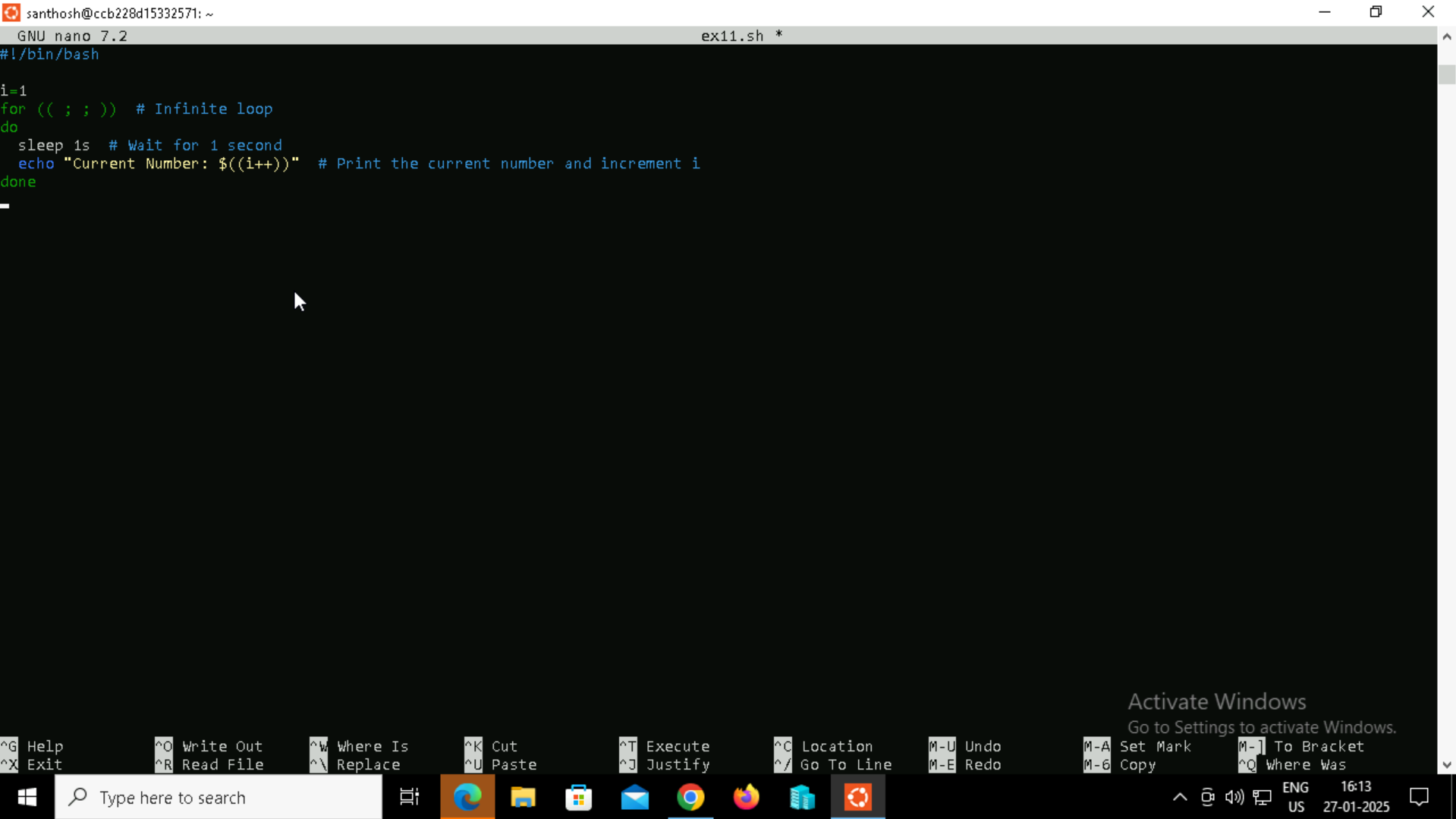
**Example 11**

**Infinite Bash for Loop**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



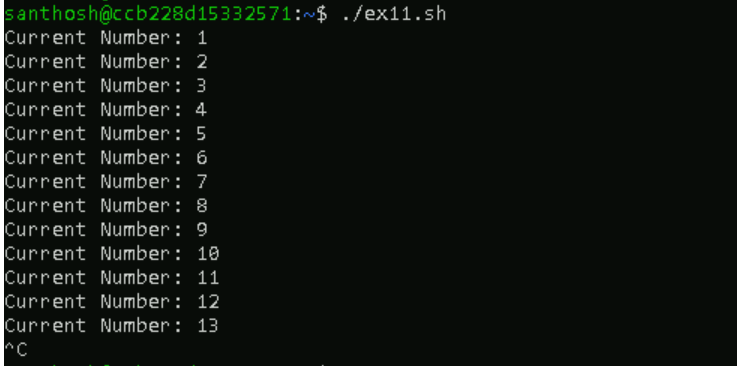
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**BASH While Loop**

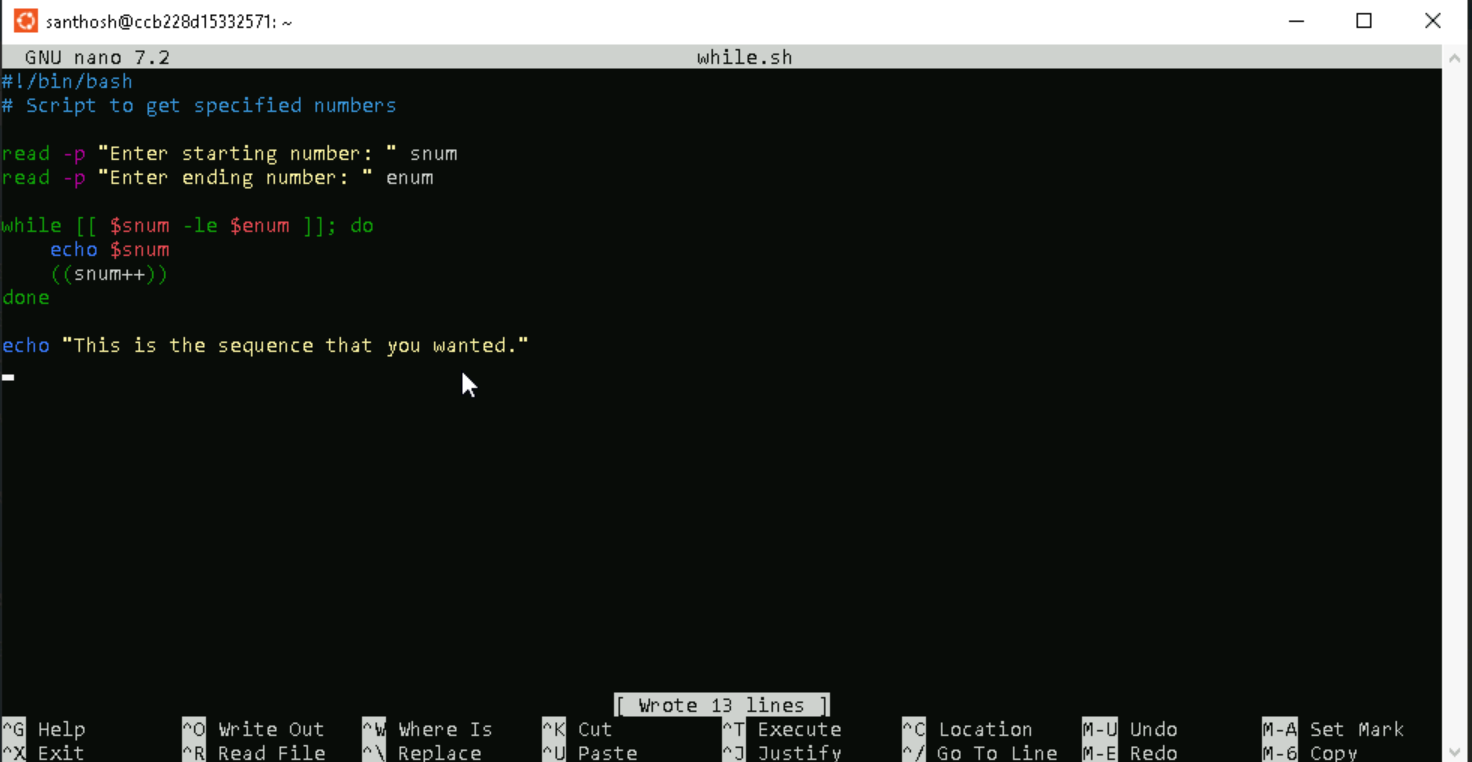
The bash while loop can be defined as a control flow statement which allows executing the given set of commands repeatedly as long as the applied condition evaluates to true. For example, we can either run echo command many times or just read a text file line by line and process the result by using while loop in Bash.

**While Loop with Single Condition**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

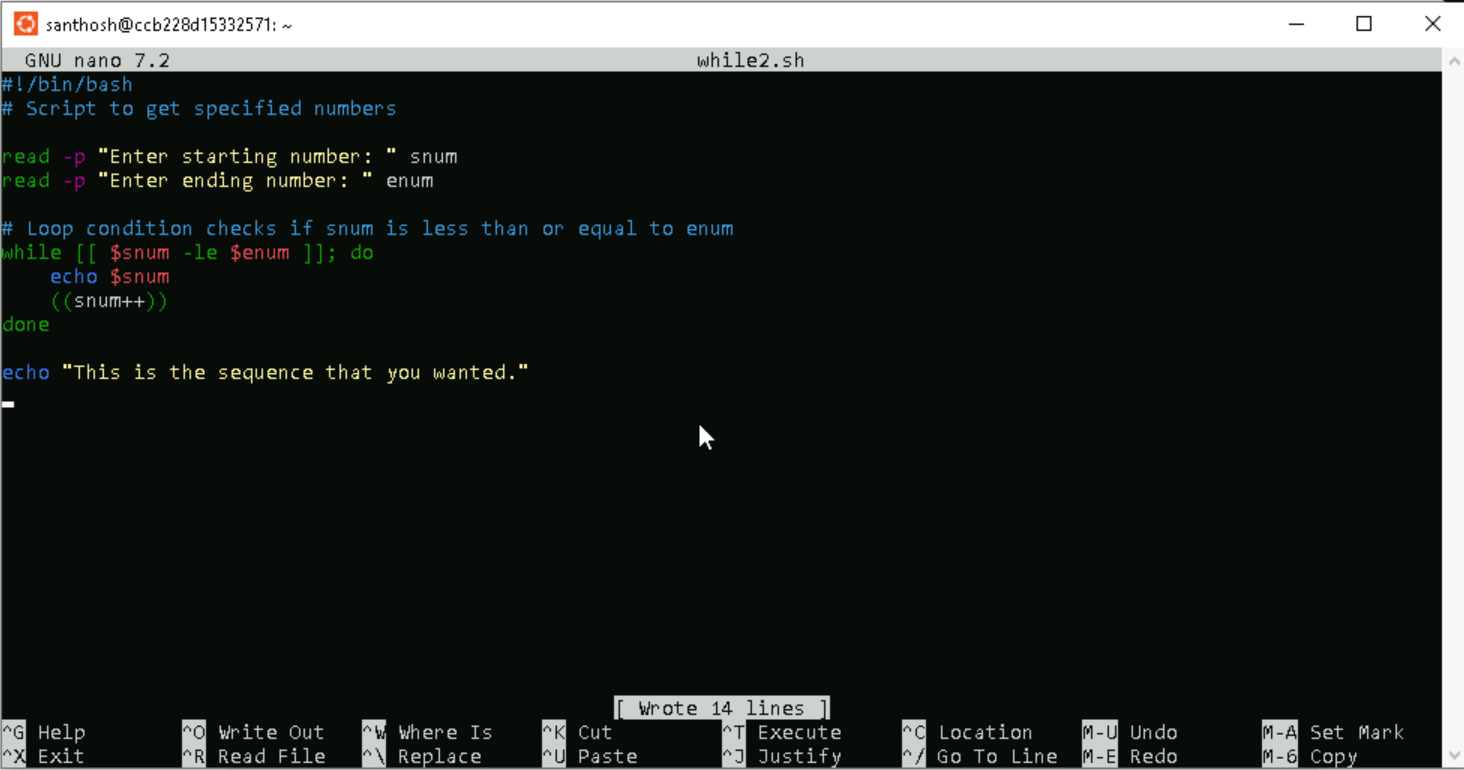


**While Loop with Multiple Conditions**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

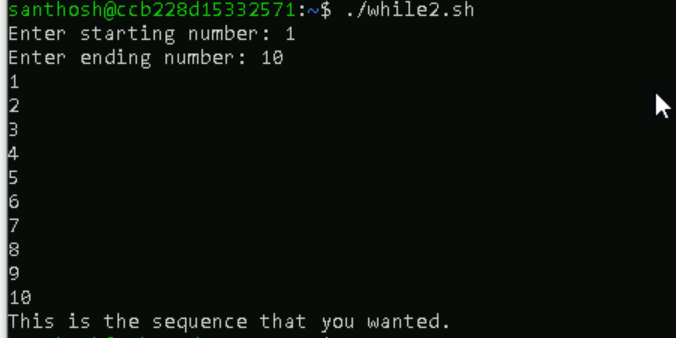
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3**: Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

****

**Infinite While Loop**

An infinite loop is a loop that has no ending or termination. If the condition always evaluates to true, it creates an infinite loop. The loop will execute continuously until it is forcefully stopped using CTRL+C :

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



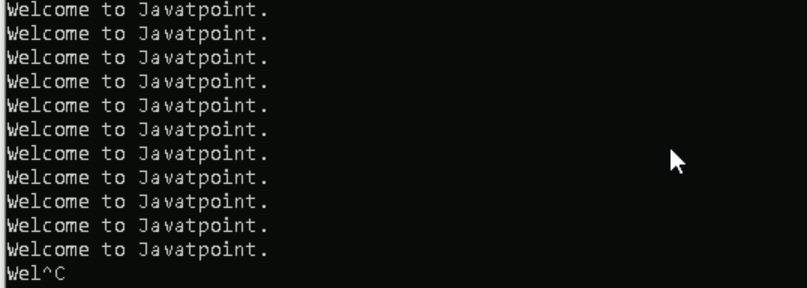
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



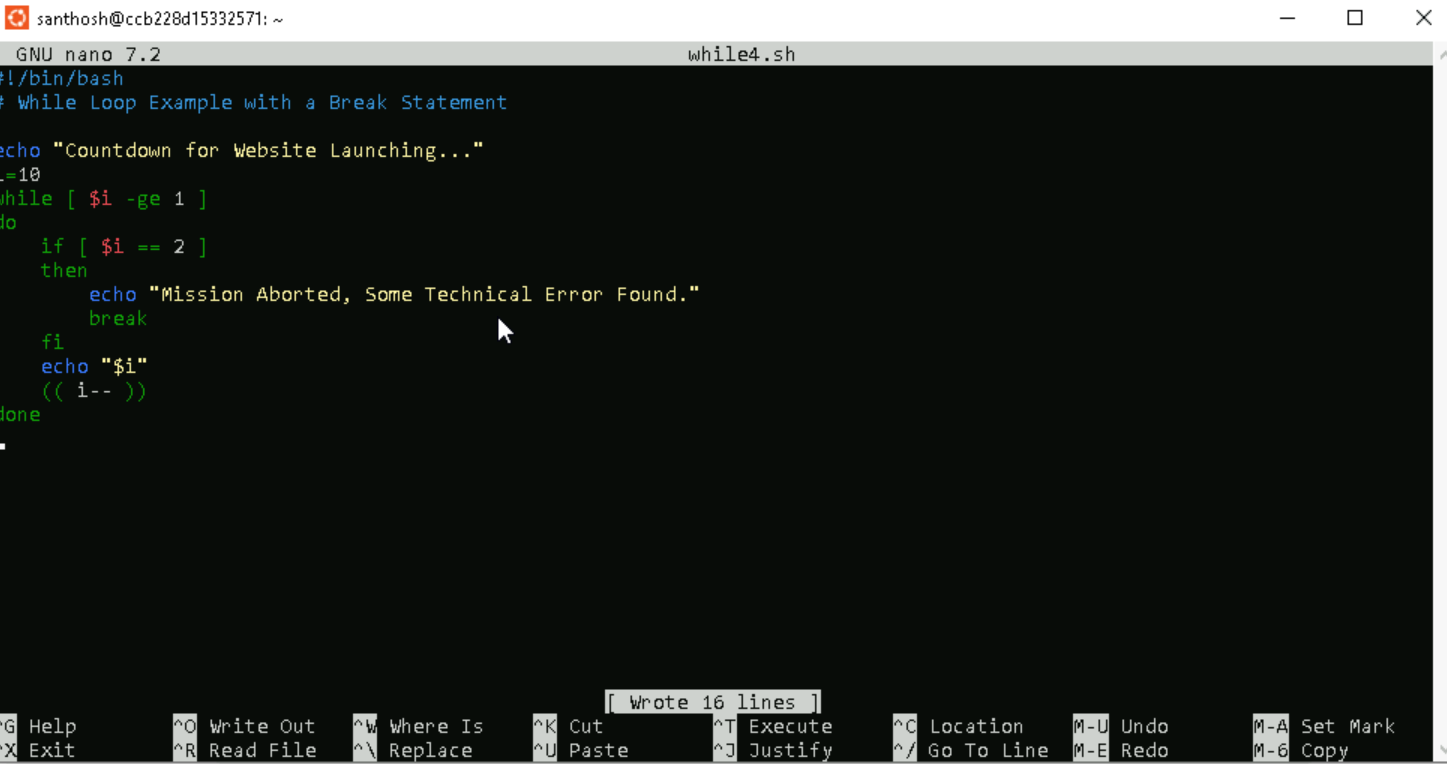
**Step 4:** Executing the output.



**While Loop with a Break Statement**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

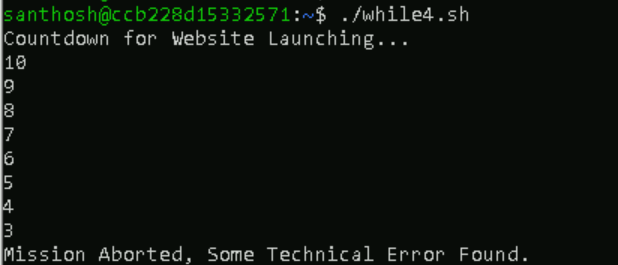
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

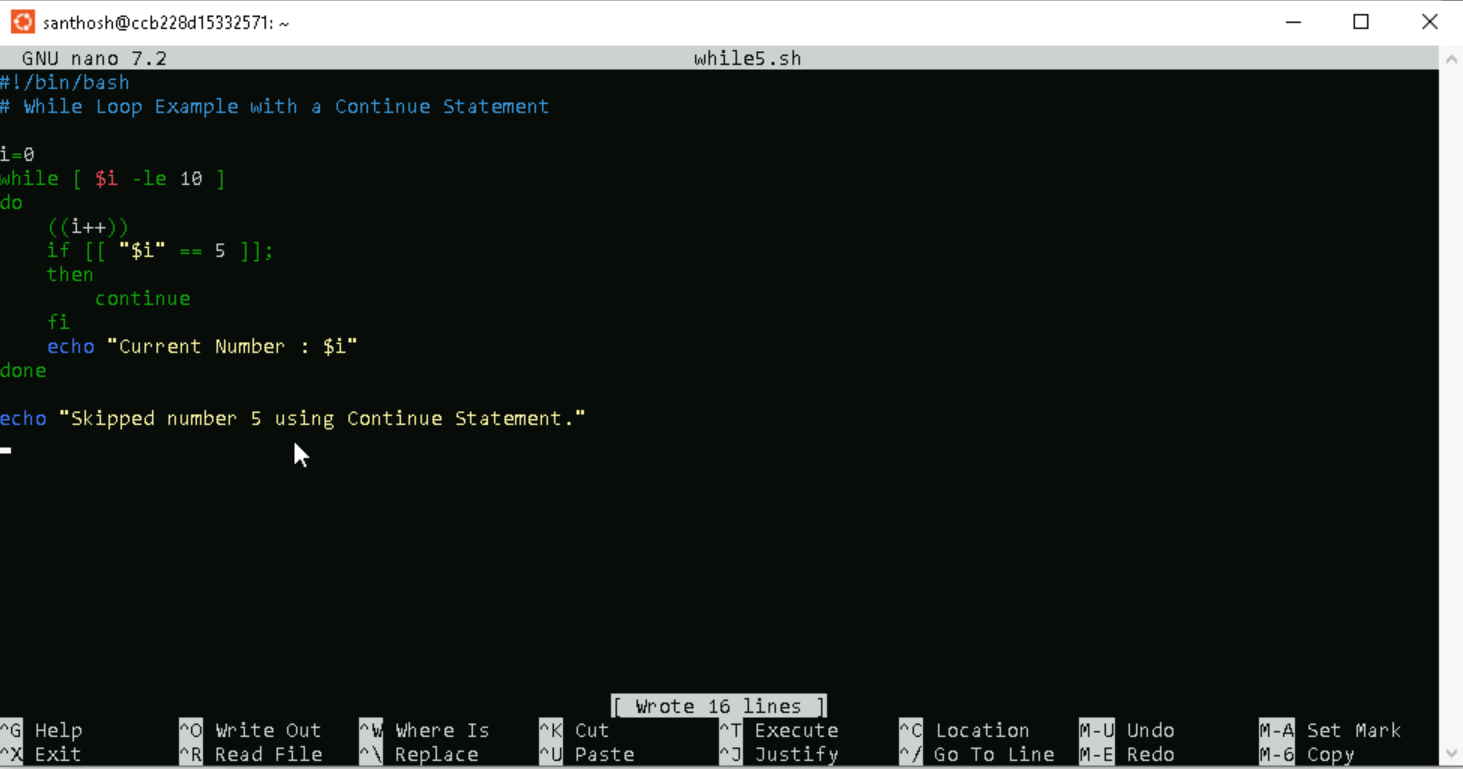


**While Loop with a Continue Statement**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



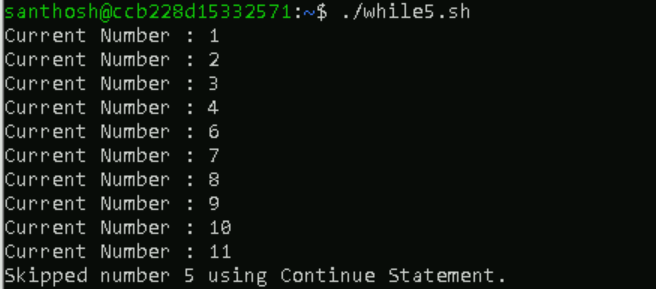
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

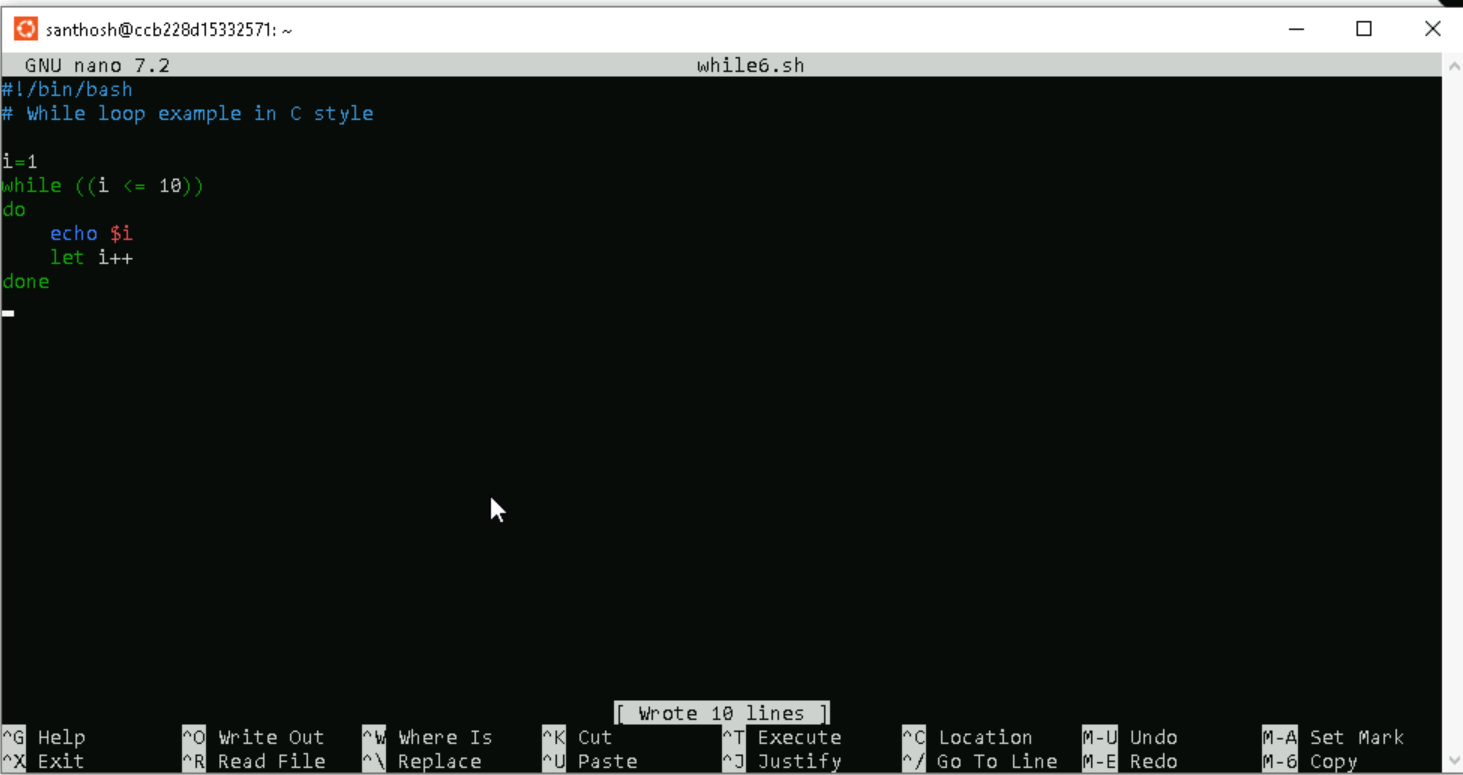


**While Loop with C-Style**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



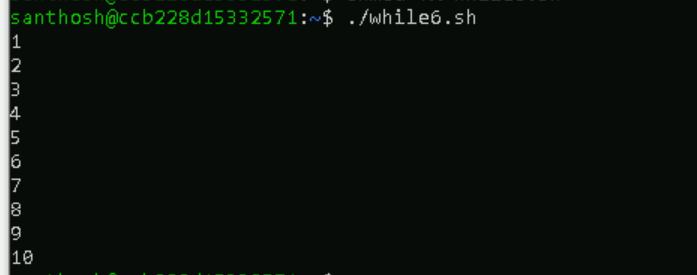
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Bash Until Loop**

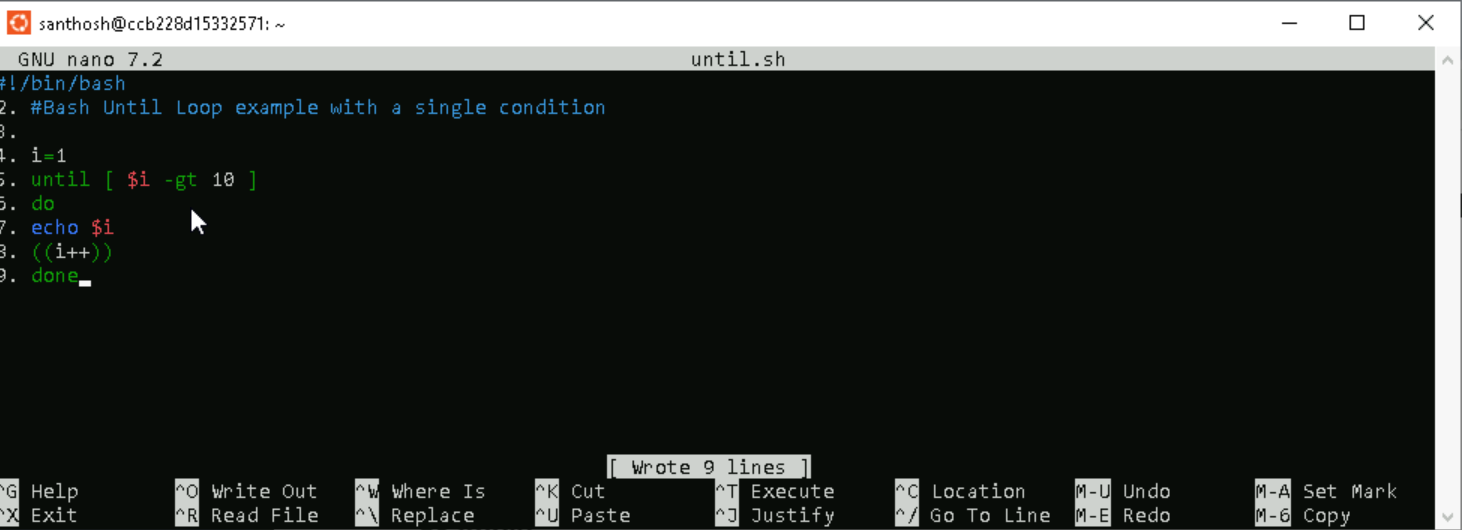
Bash Until Loop in a bash scripting is used to execute a set of commands repeatedly based on the boolean result of an expression. The set of commands are executed only until the expression evaluates to true. It means that when the expression evaluates to false, a set of commands are executed iteratively. The loop is terminated as soon as the expression evaluates to true for the first time.

**Until Loop with Single Condition**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.



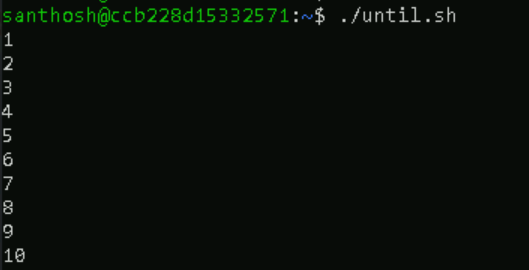
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

****

**Until Loop with Multiple Conditions**

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

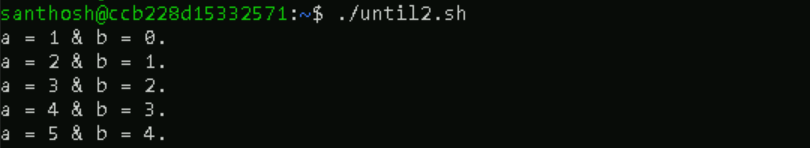
**Step 2**: Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

****

**Bash String**

Bash String is a data type such as as an integer or floating-point unit. It is used to represent text rather than numbers. It is a combination of a set of characters that may also contain numbers.

**Equal Operator**

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

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Not Equal Operator**

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

**Step 1:** Creating a bash script using touch command and adding the script bby editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

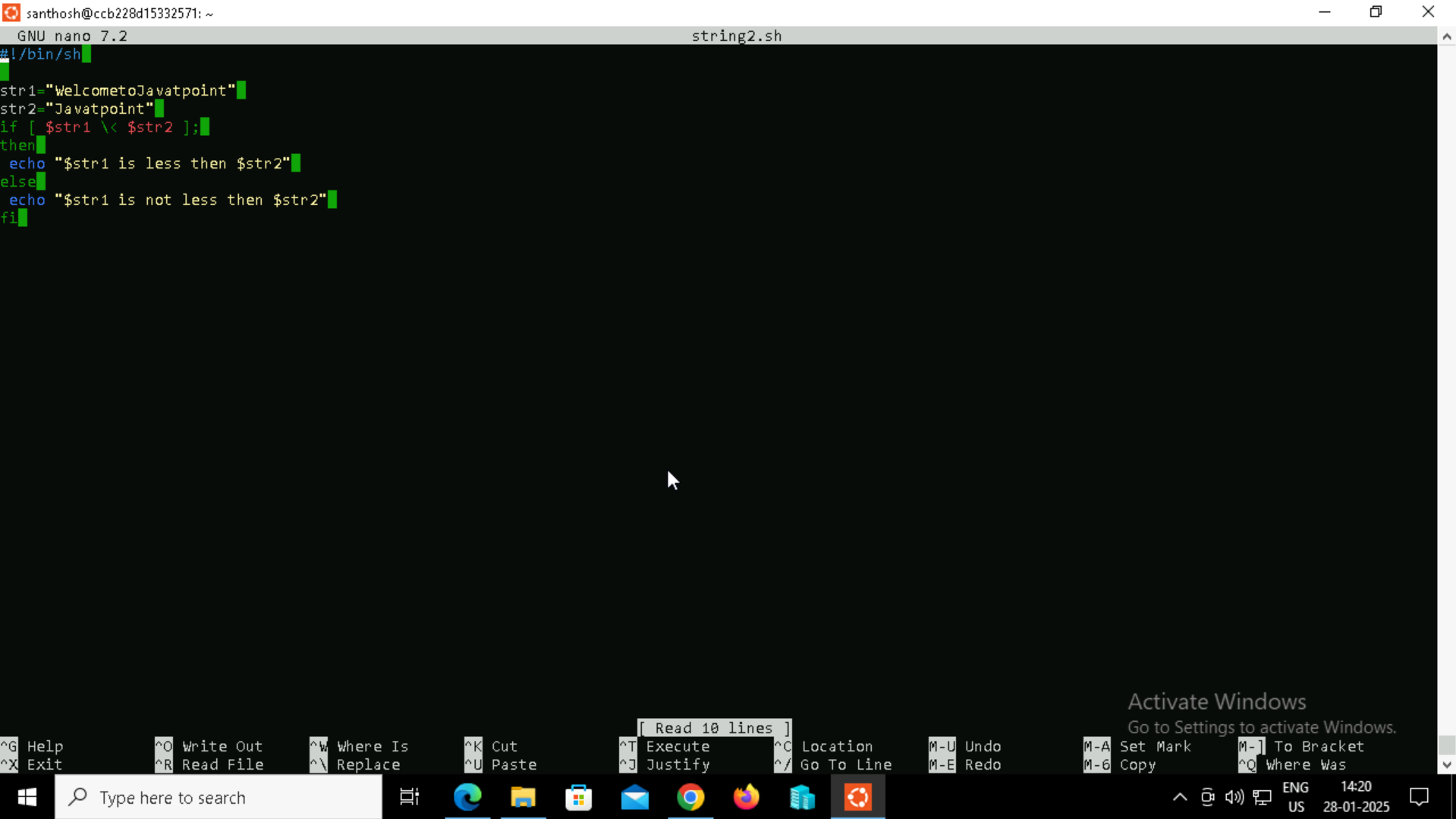
****

**Less than Operator**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

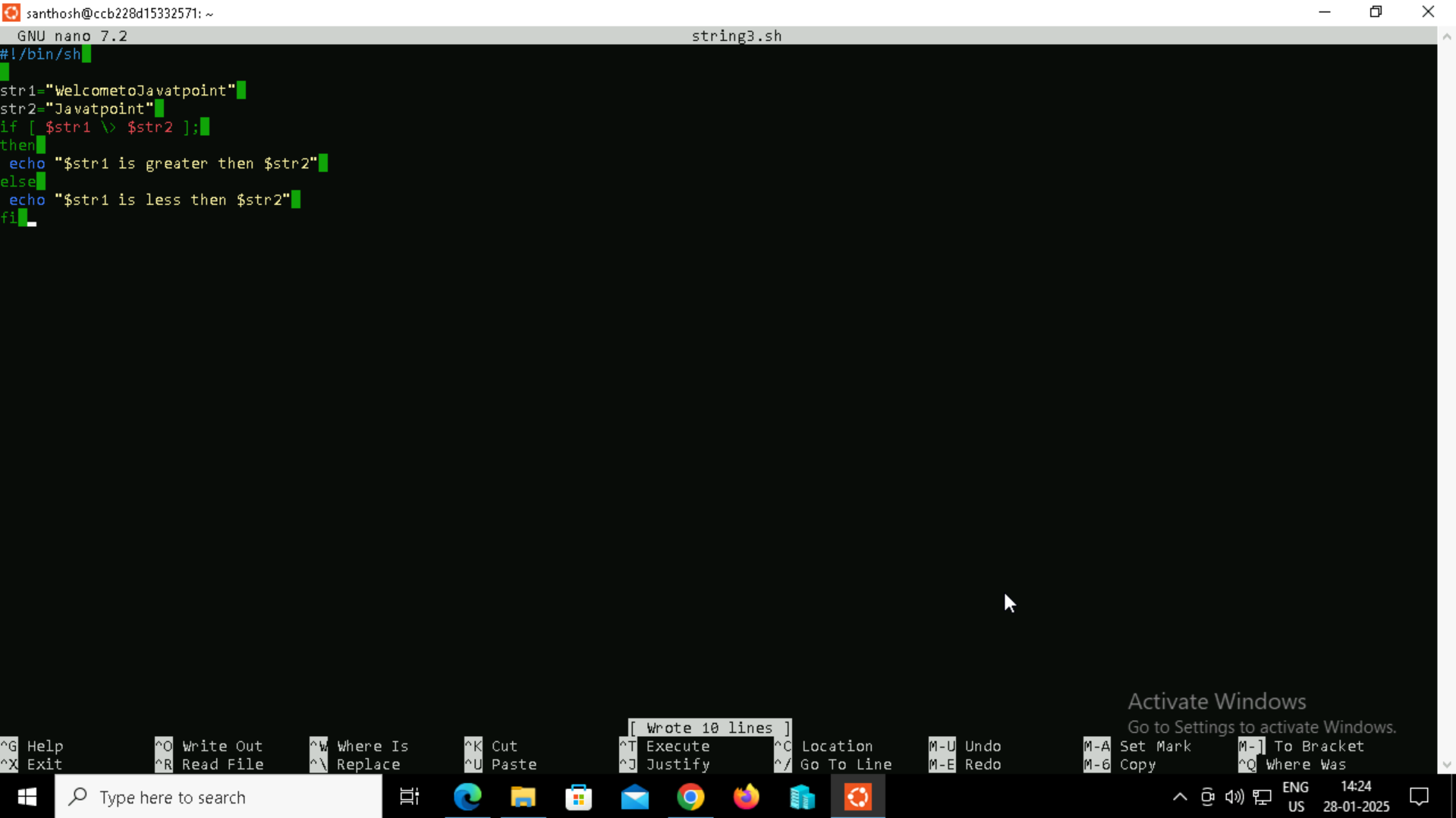
****

**Greater than Operator**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

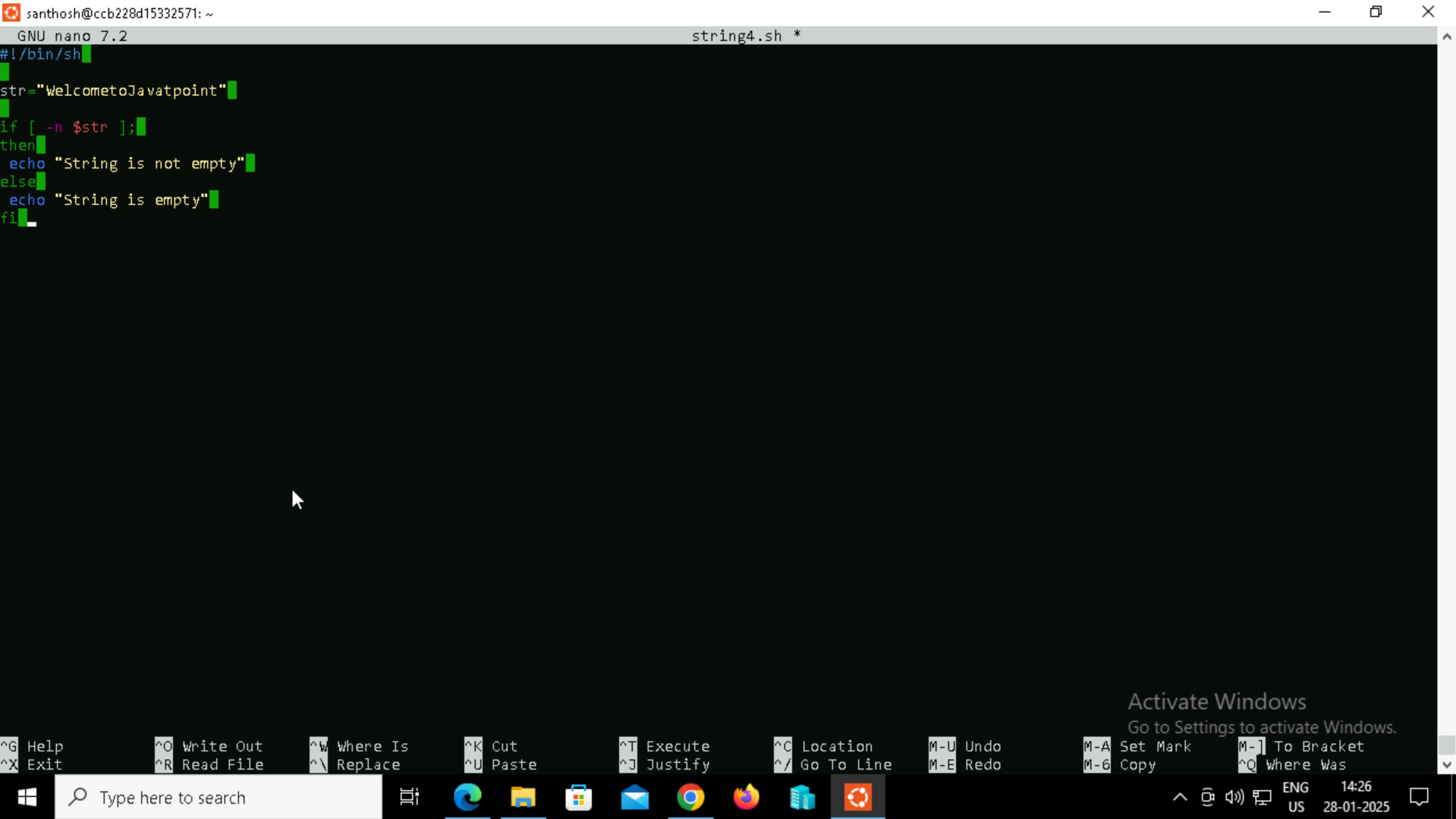


**To check if the string length is greater than Zero:**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

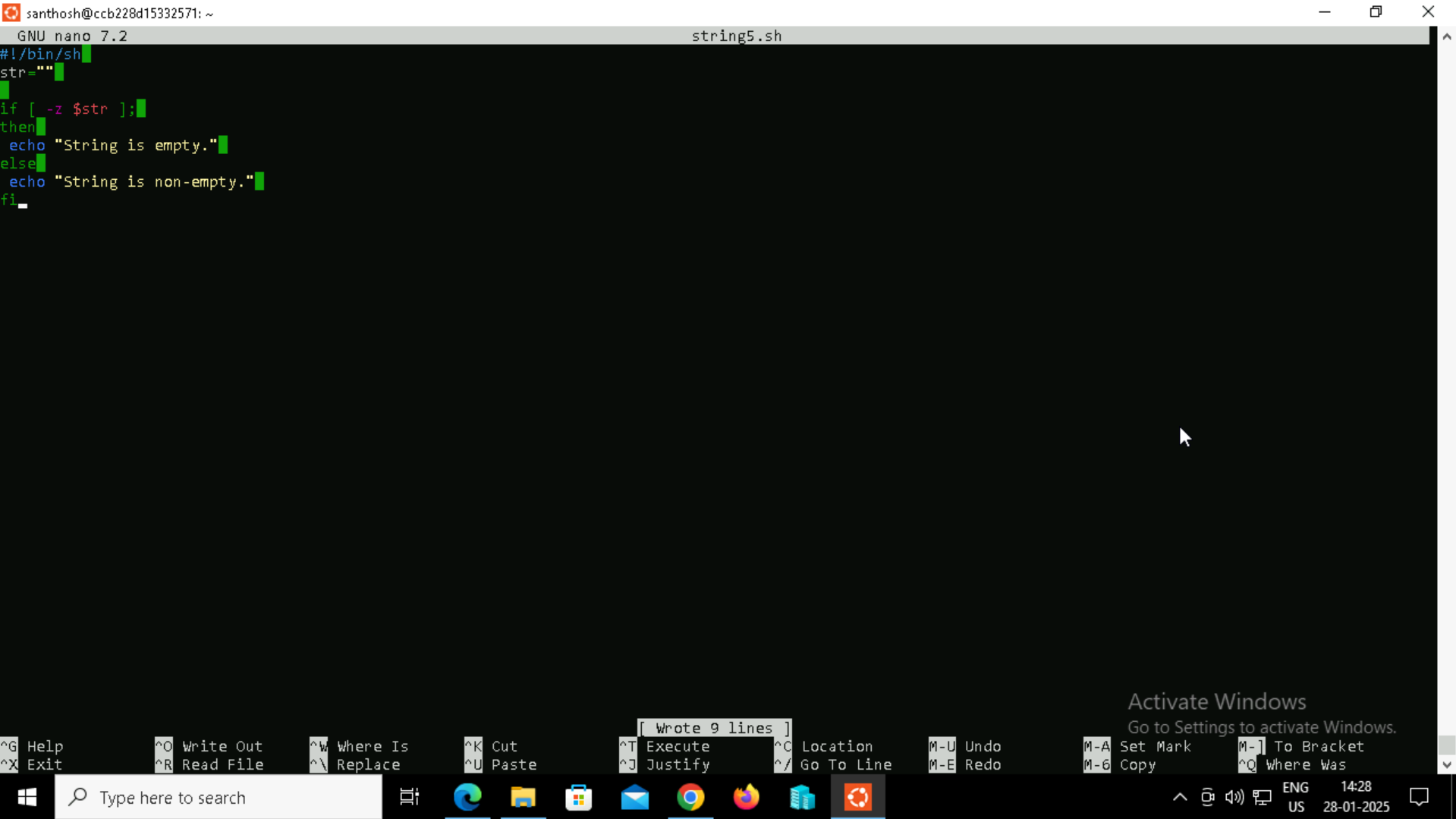


**To check if the string length is equal to Zero**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Bash Find**

The total number of characters in any string indicates the length of a string. In some cases, we might need to know about the length of a string to perform some specific tasks. Most of the programming languages have their own built-in functions to calculate the number of characters. However, Bash does not contain such type of built-in functions. But there are several ways that we can use to find the length of a string in Bash Scripting.

**Bash StringLength**

**Examples to find String Length in Bash**

**Example 1**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.



**Example 2**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

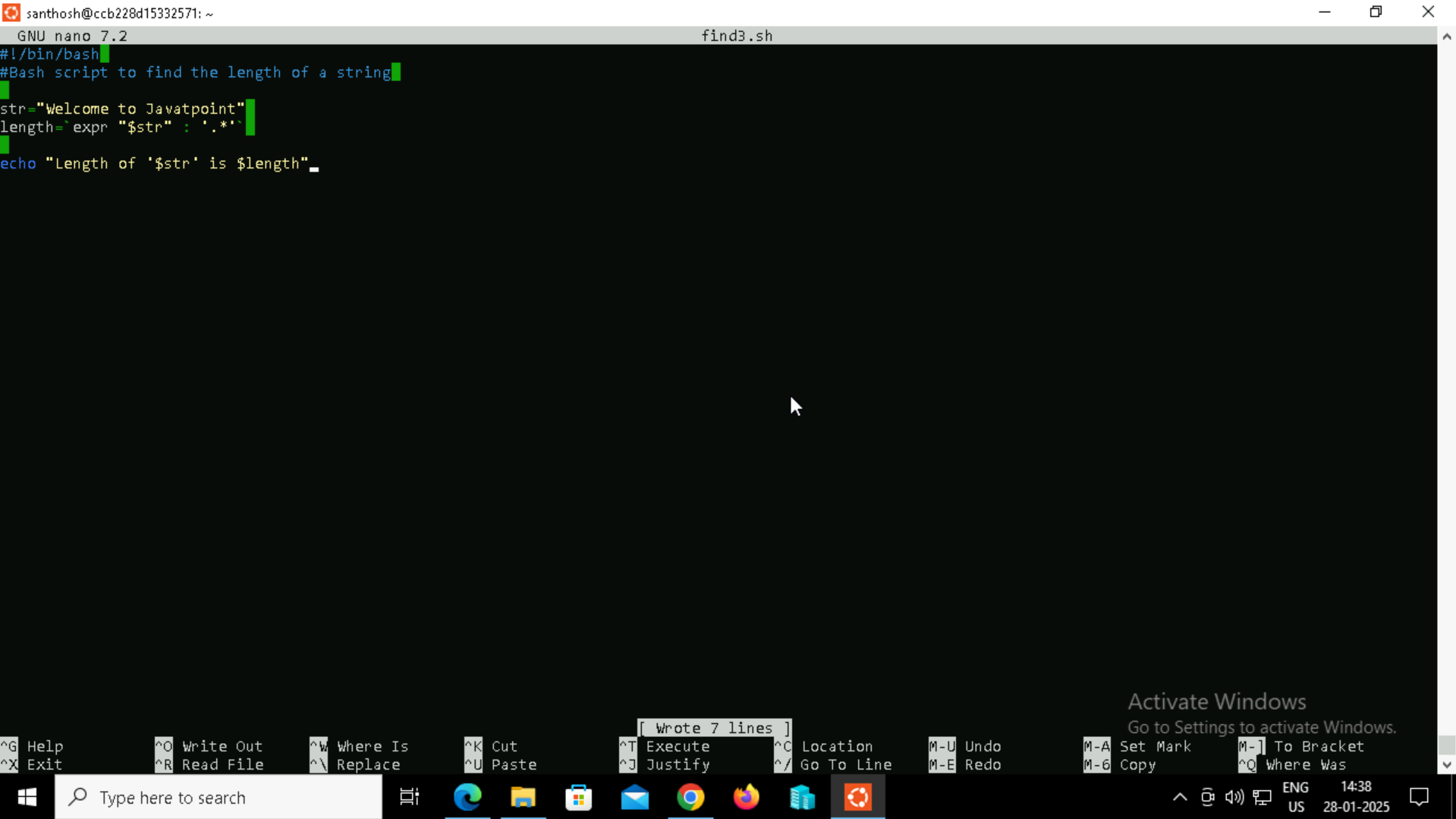
****

**Example 3**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

**Step 4:** Executing the output.

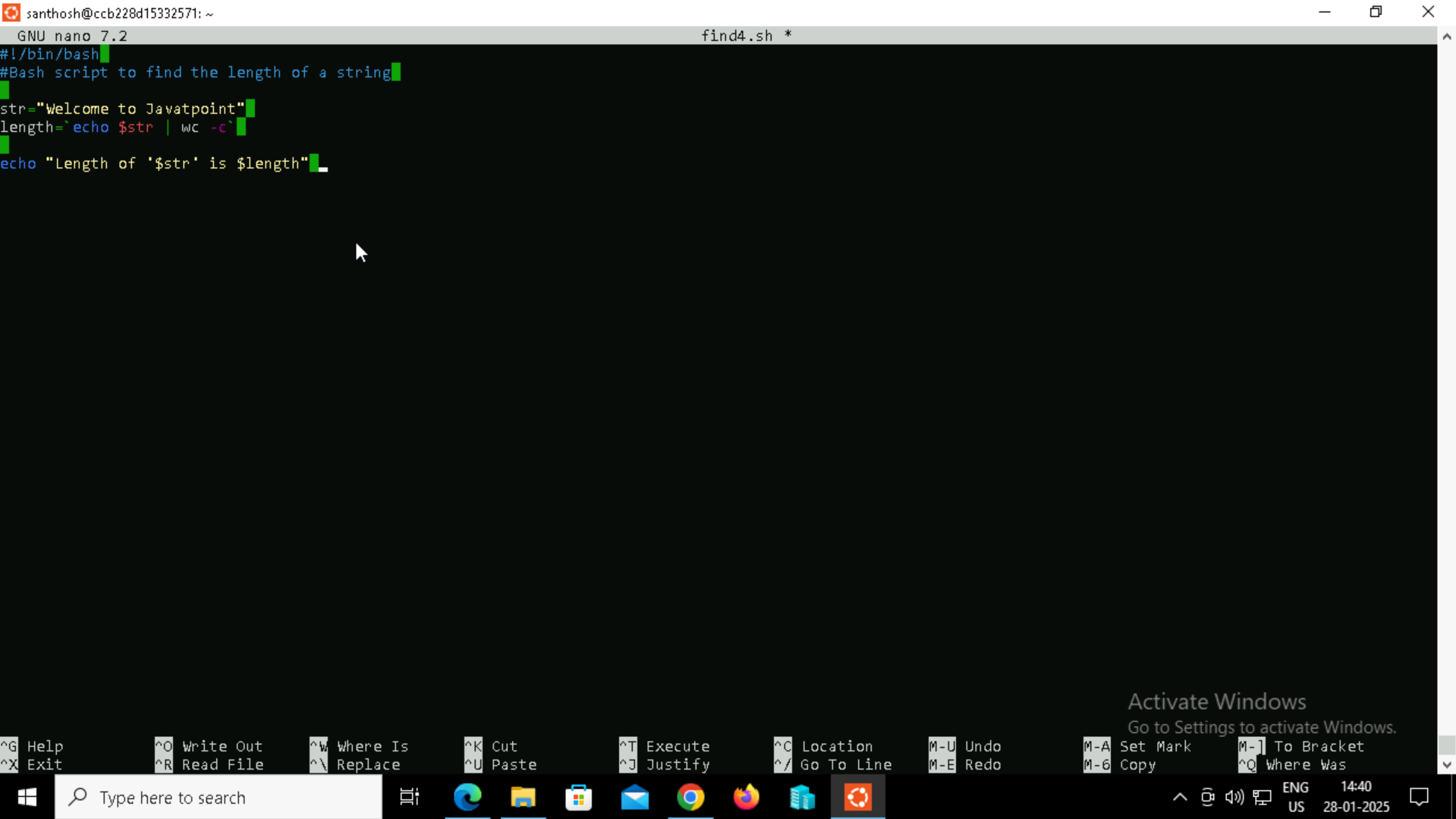
****

**Example 4**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.



**Example 5**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.



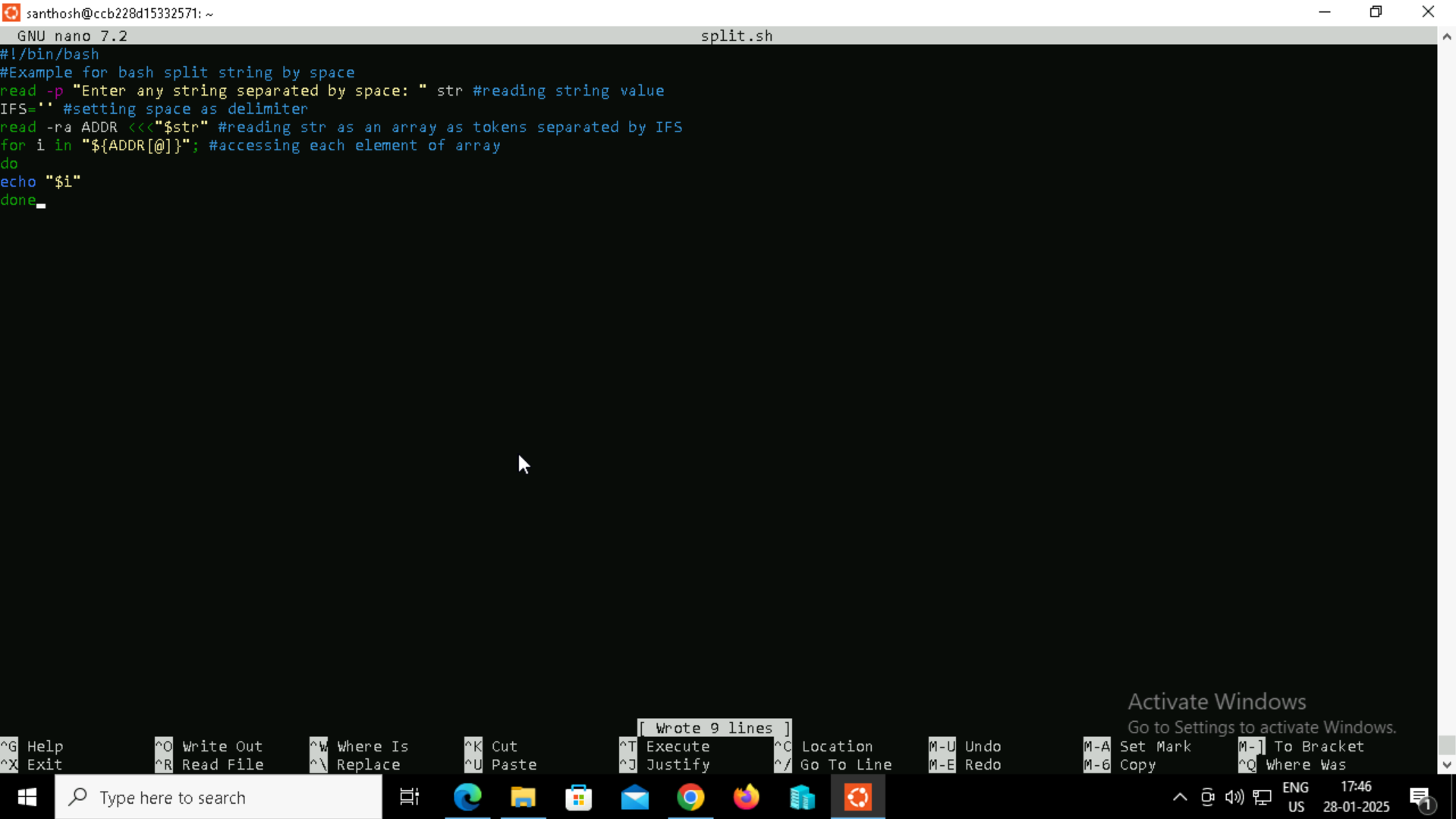
**Bash Split String**

**Example 1: Bash Split String by Space**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

****

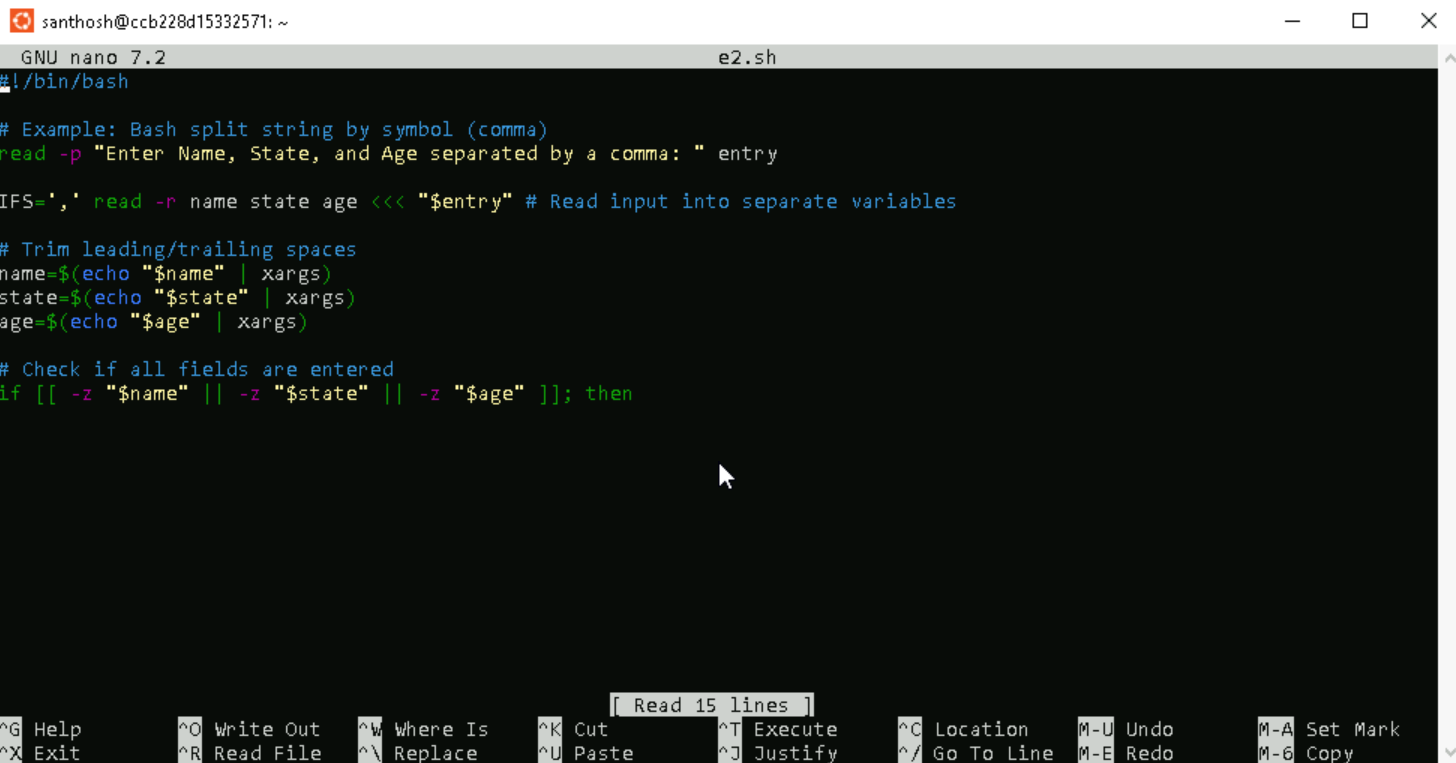
**Example 2: Bash Split String by Symbol**

In some cases, we may have a requirement to split a string by other delimiters such as a symbol or specific character. In this example, a string is split using a comma (,) symbol character as a delimiter.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

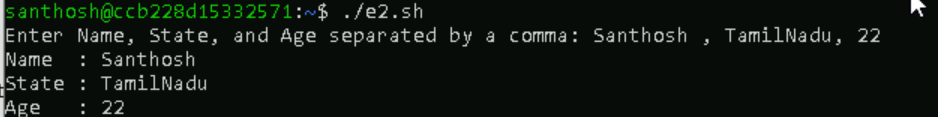
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

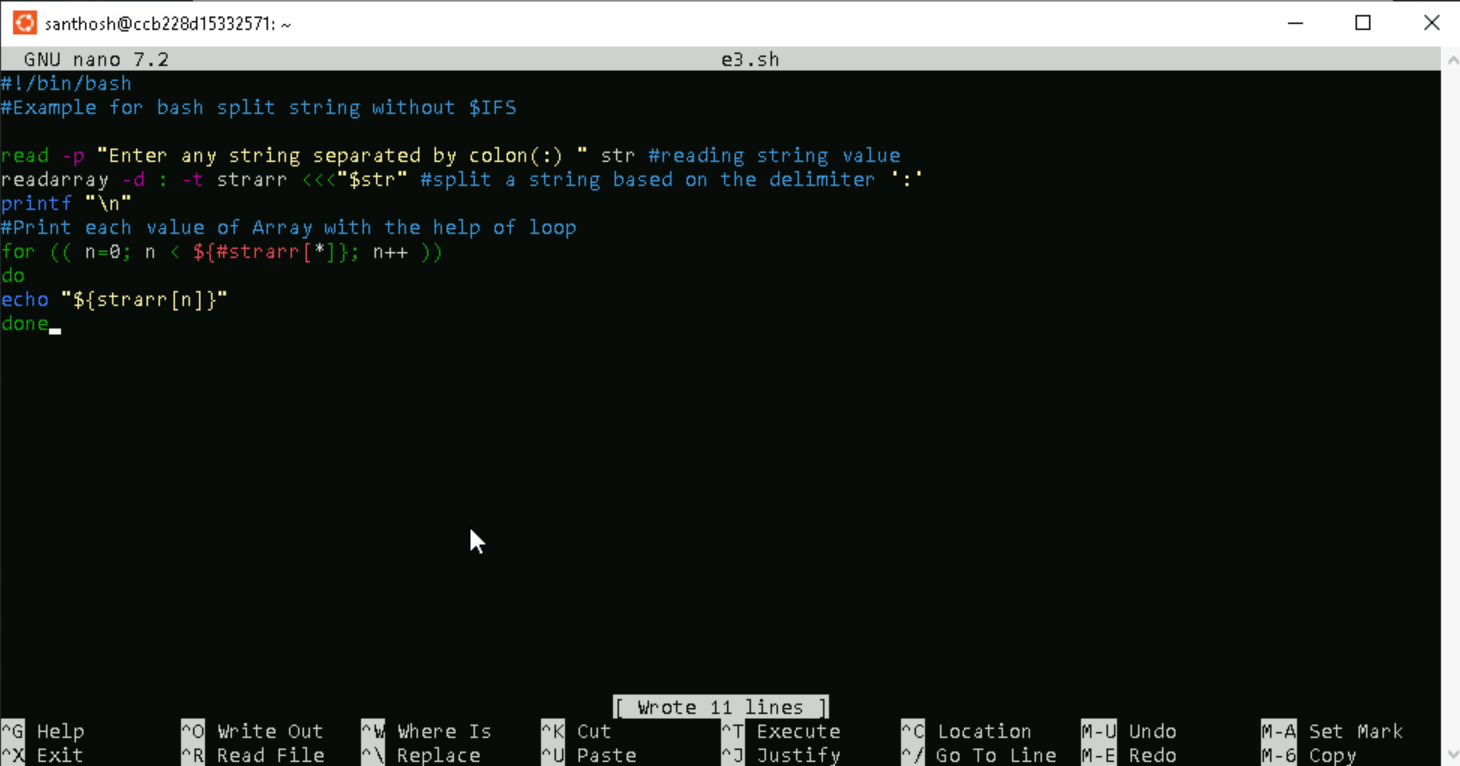


**Split without $IFS variable**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

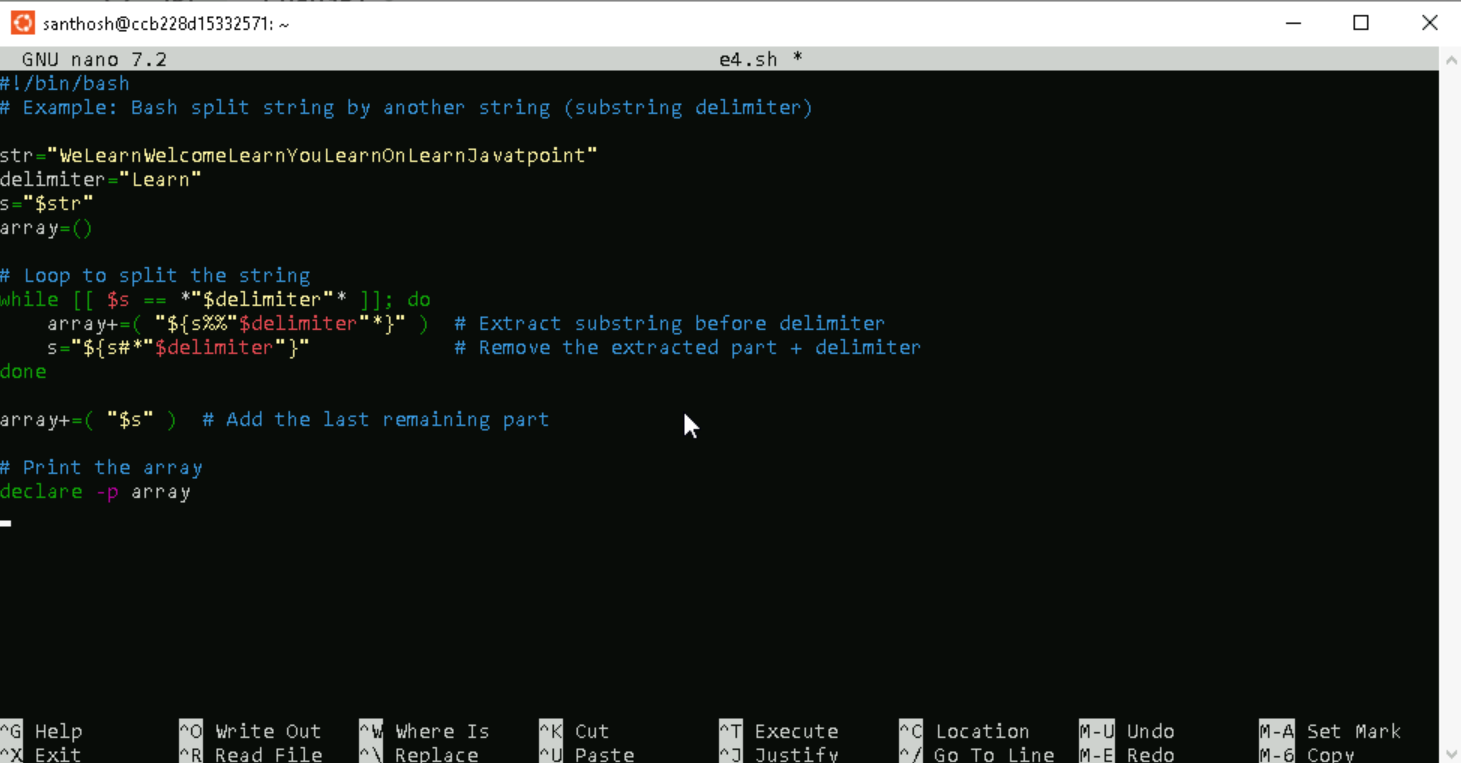


**Example 2: Bash Split String by another string.**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

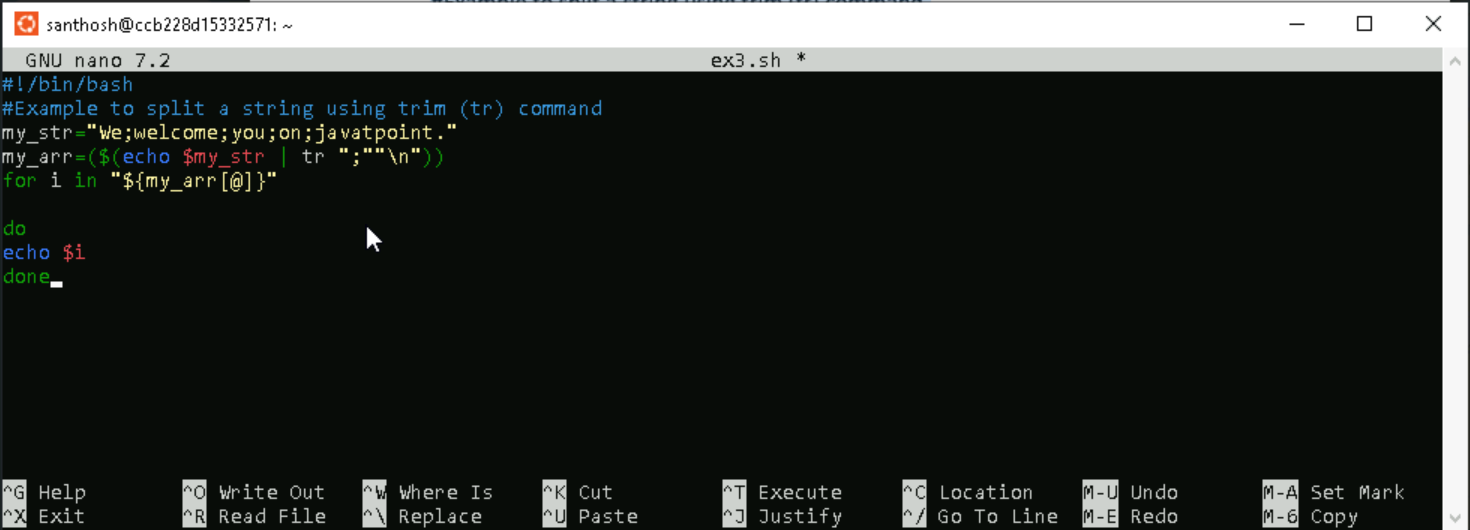


**Example 3: Bash Split String using Trim Command**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



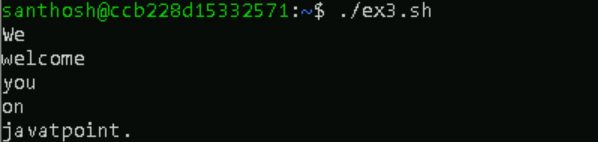
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Bash Substring**

A substring is a sequence of characters within a string. Bash provides an option to extract the information from a string itself. You can extract the digits or a given string using several methods.

**Example 1: To Extract till Specific Characters from Starting**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



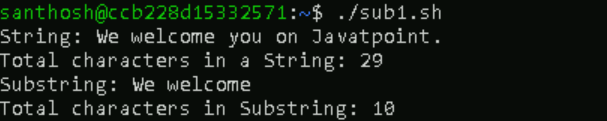
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

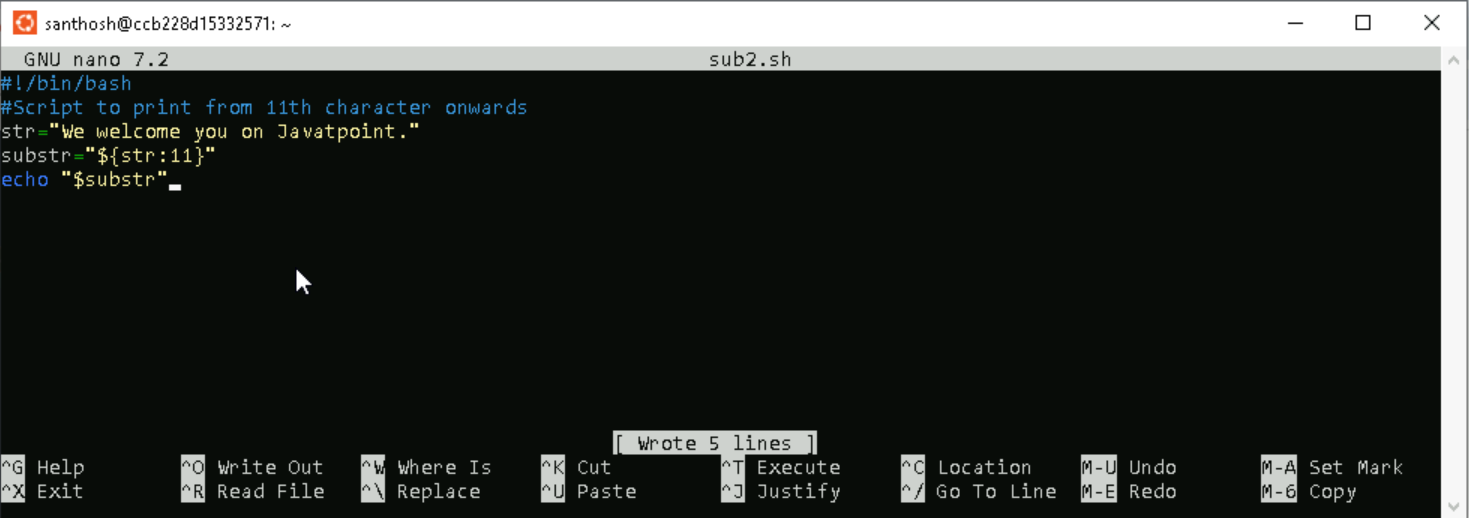
****

**Example 2: To Extract from Specific Character onwards**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

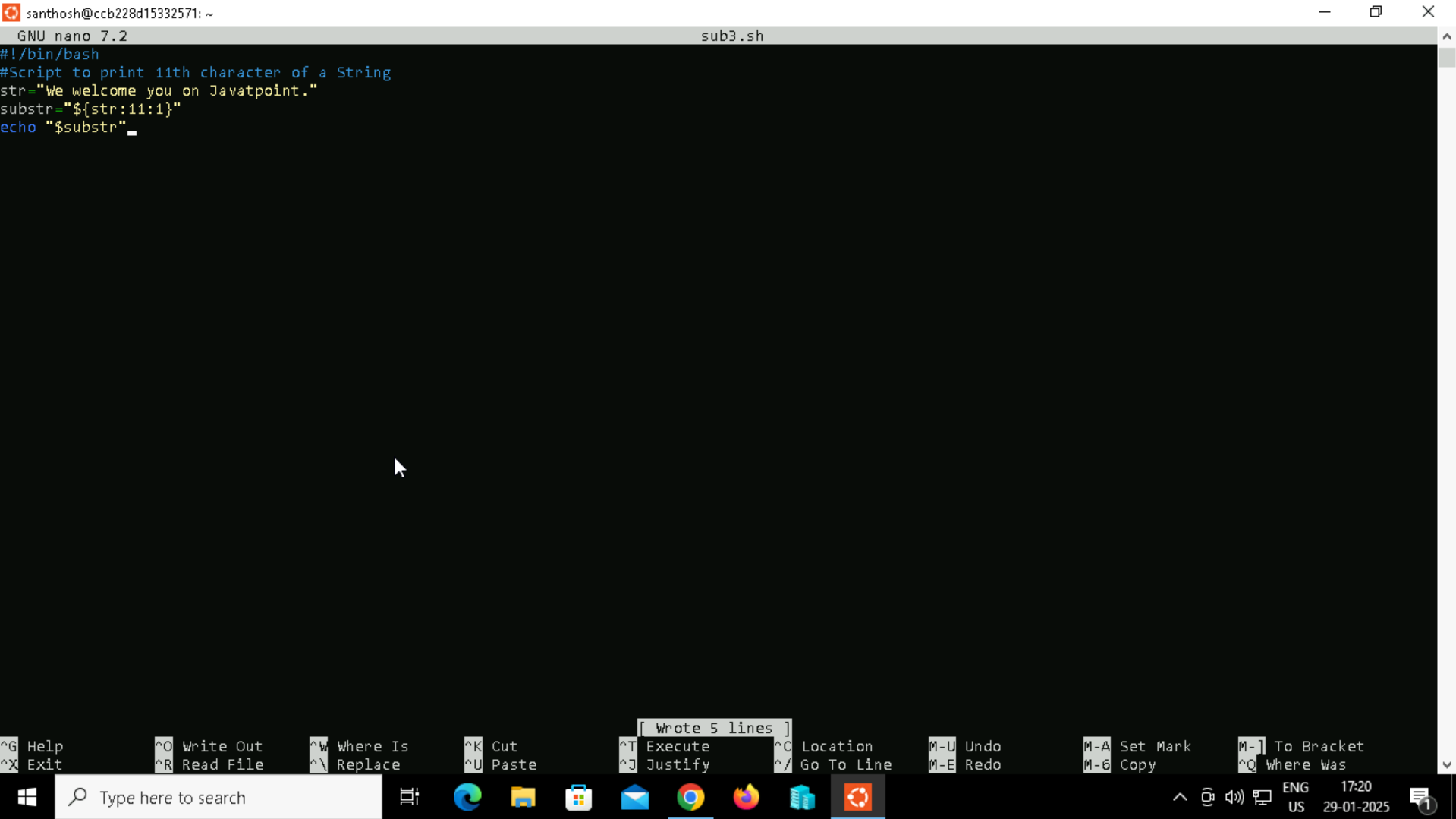
****

**Example 3: To Extract a Single Character**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

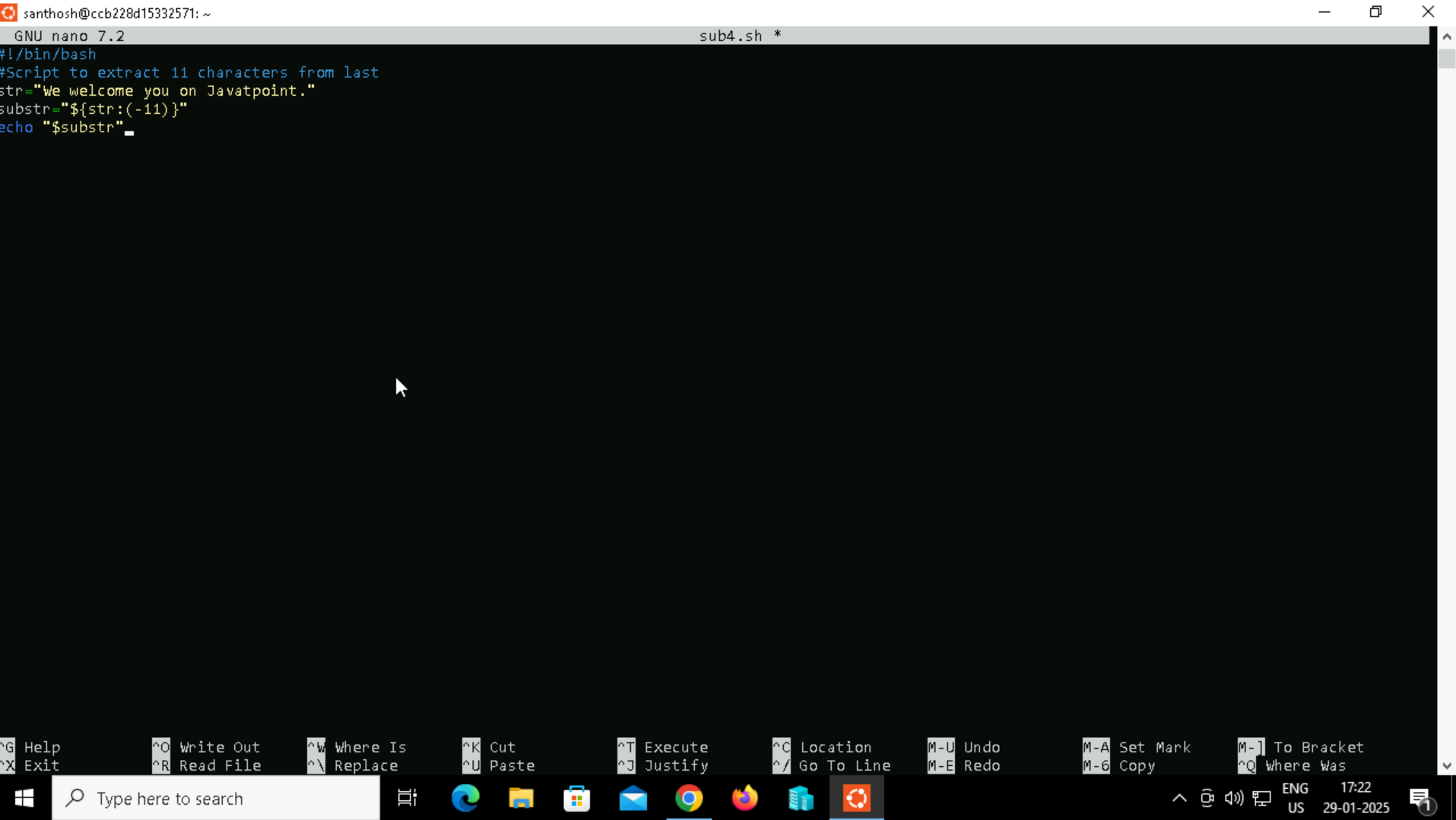
****

**Example 4: To Extract the specific characters from last**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

****

**Bash Concatenate String**

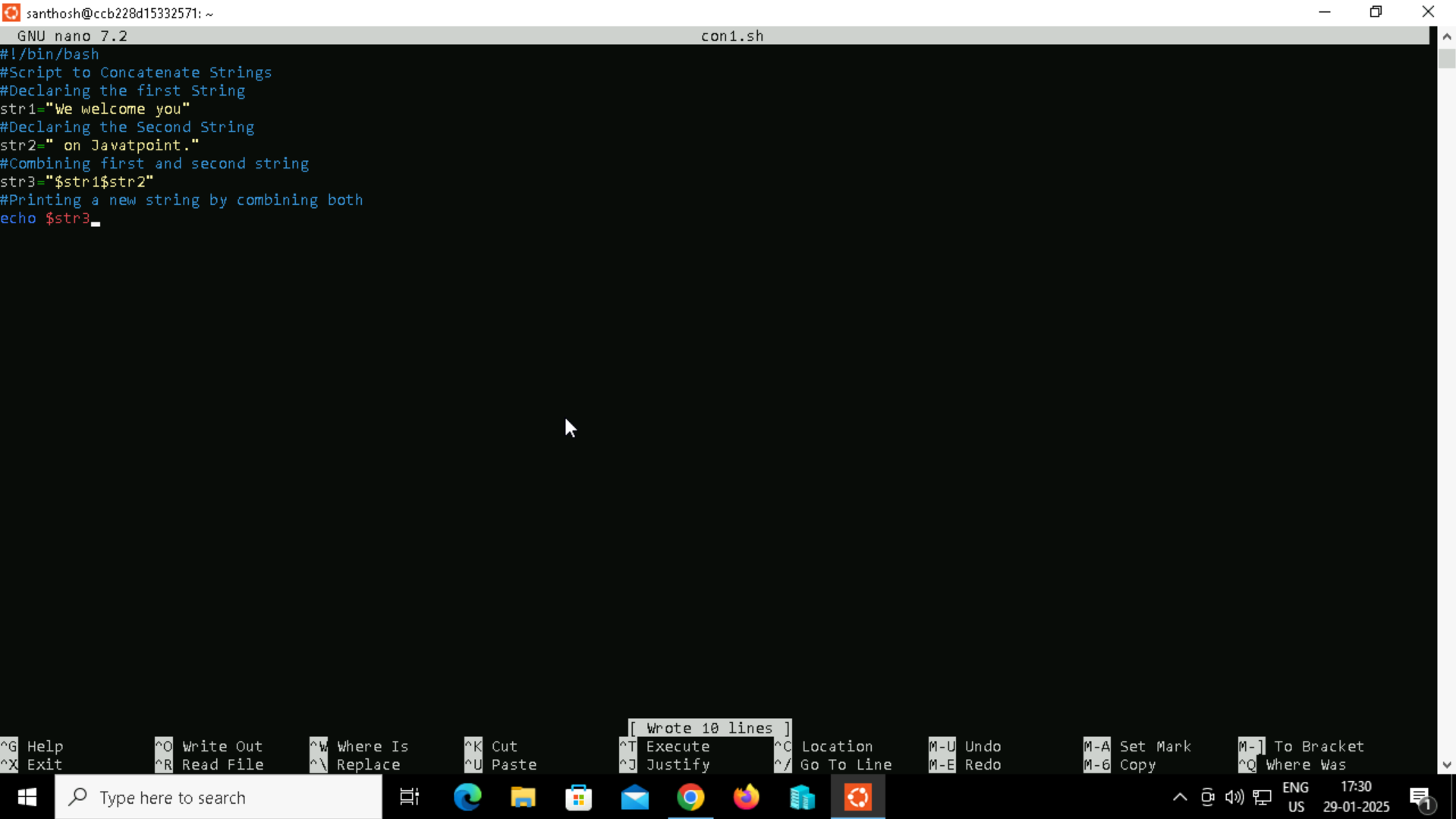
In bash scripting, we can add or join two or more strings together, which is known as string concatenation. It is one of the common requirement for any programming language. A special character or built-in function is applied to perform string concatenation. However, Bash does not contain any built-in function to combine string data or variables. The easiest method to perform string concatenation in bash is to write variables side by side.

**Example 1: Write Variables Side by Side**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

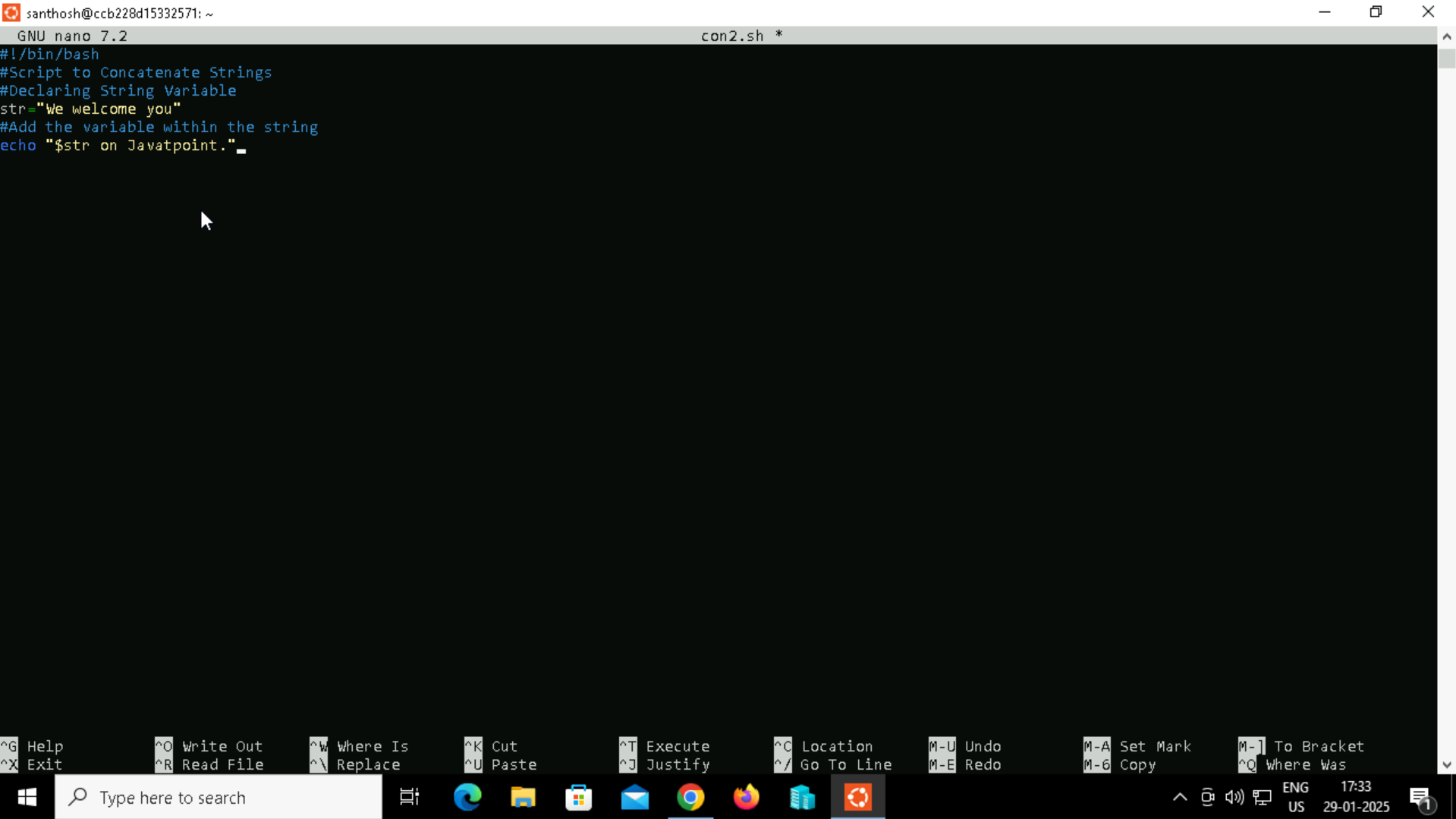
**Example 2: Using Double Quotes**

Another easy method is to use variables inside the string, which is defined with double-quotes. The string variable can be applied in any position of the string data**.**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

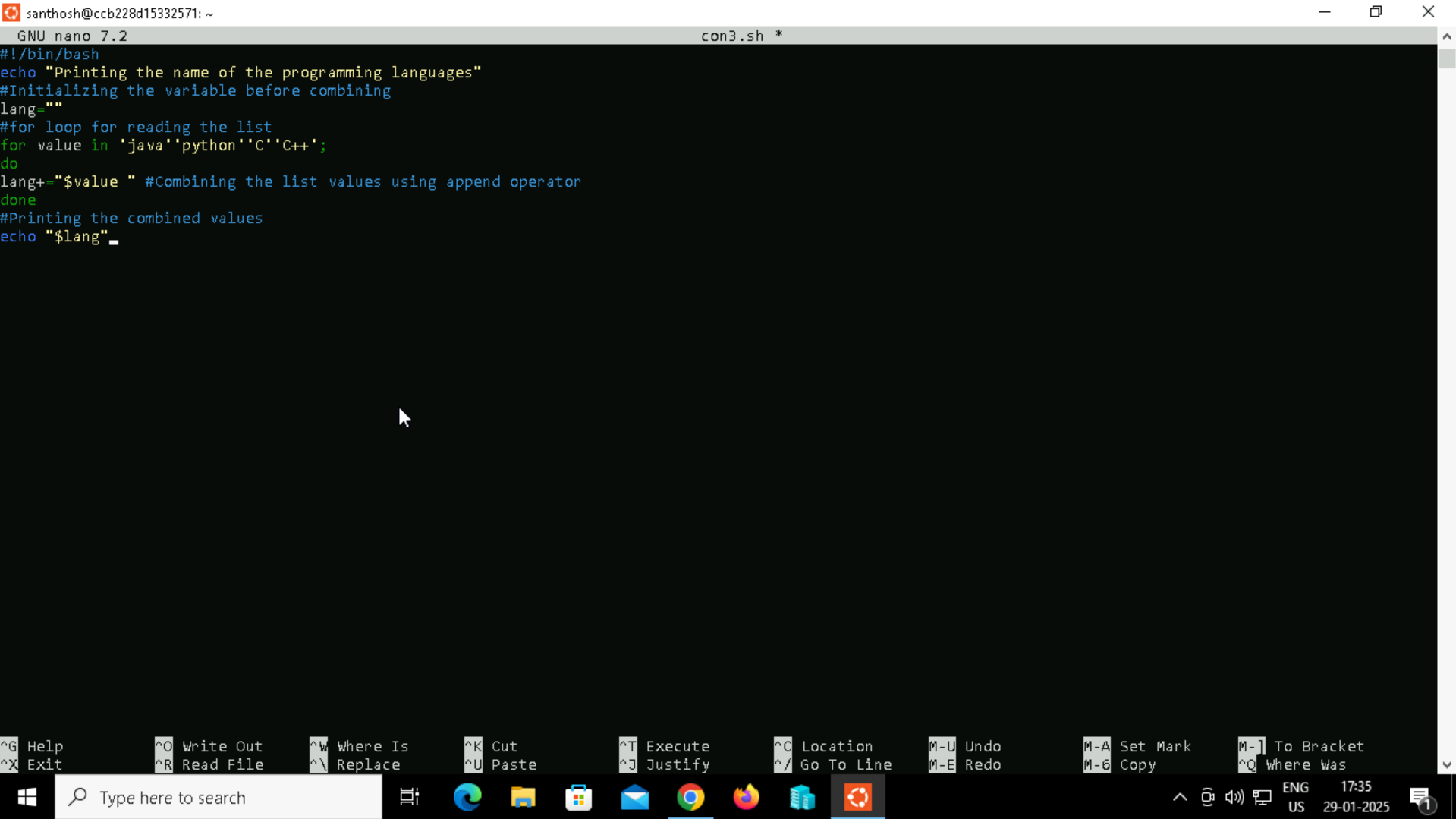
**Example 3: Using Append Operator with Loop**

Most of the popular programming languages provide support for append operator (+=) which is the combination of the plus and equal sign. It will add new strings to the end of the string variable.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



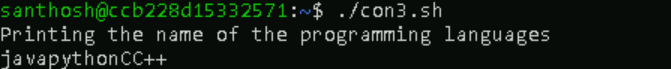
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

**Example 4: Using the Printf Function**

In bash, printf is a function which is used to print and concatenate the strings.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

**Example 5: Using Literal Strings**

String concatenation can also be performed with a literal string by using curly braces{}. They should be used in such a way that the variable does not mix up with the literal string.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

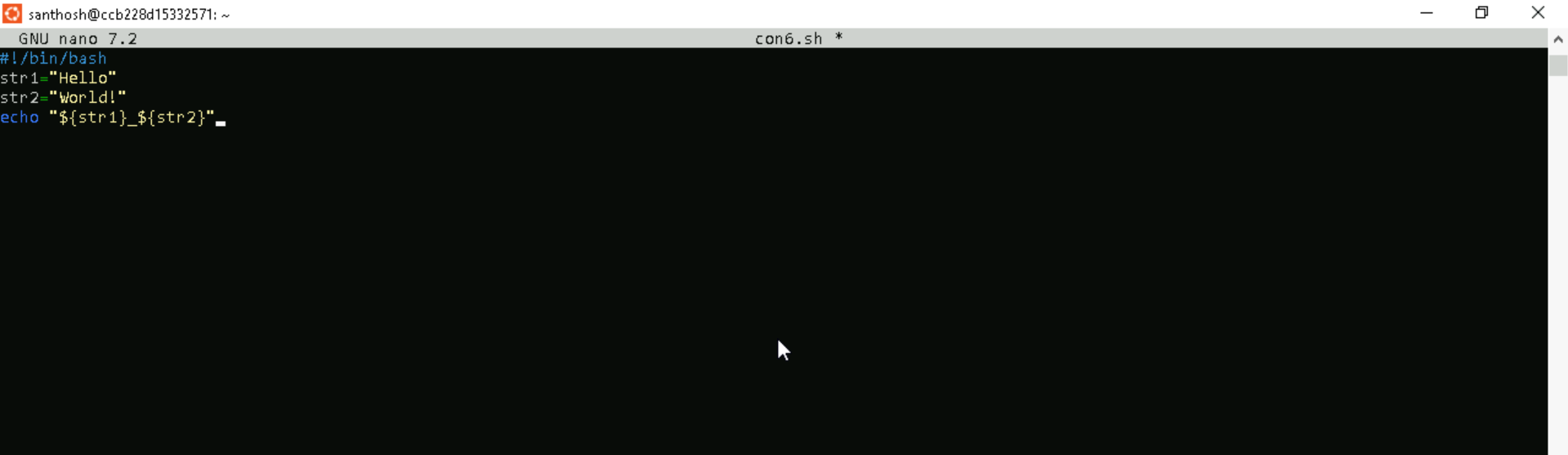
**Example 6: Using Underscore Using underscore**

For concatenating the string in bash shell is one of the common tasks. It is mostly used for assigning a name to the files.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

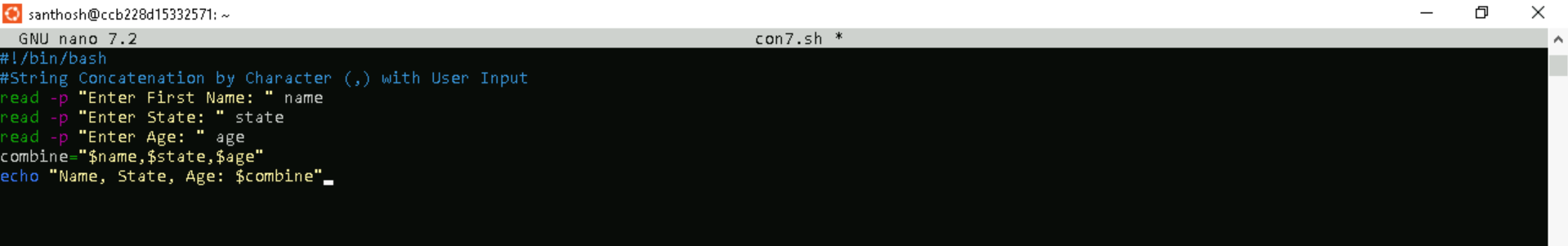
****

**Example 7: Using any Character**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



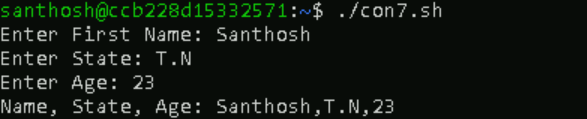
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output**.**

****

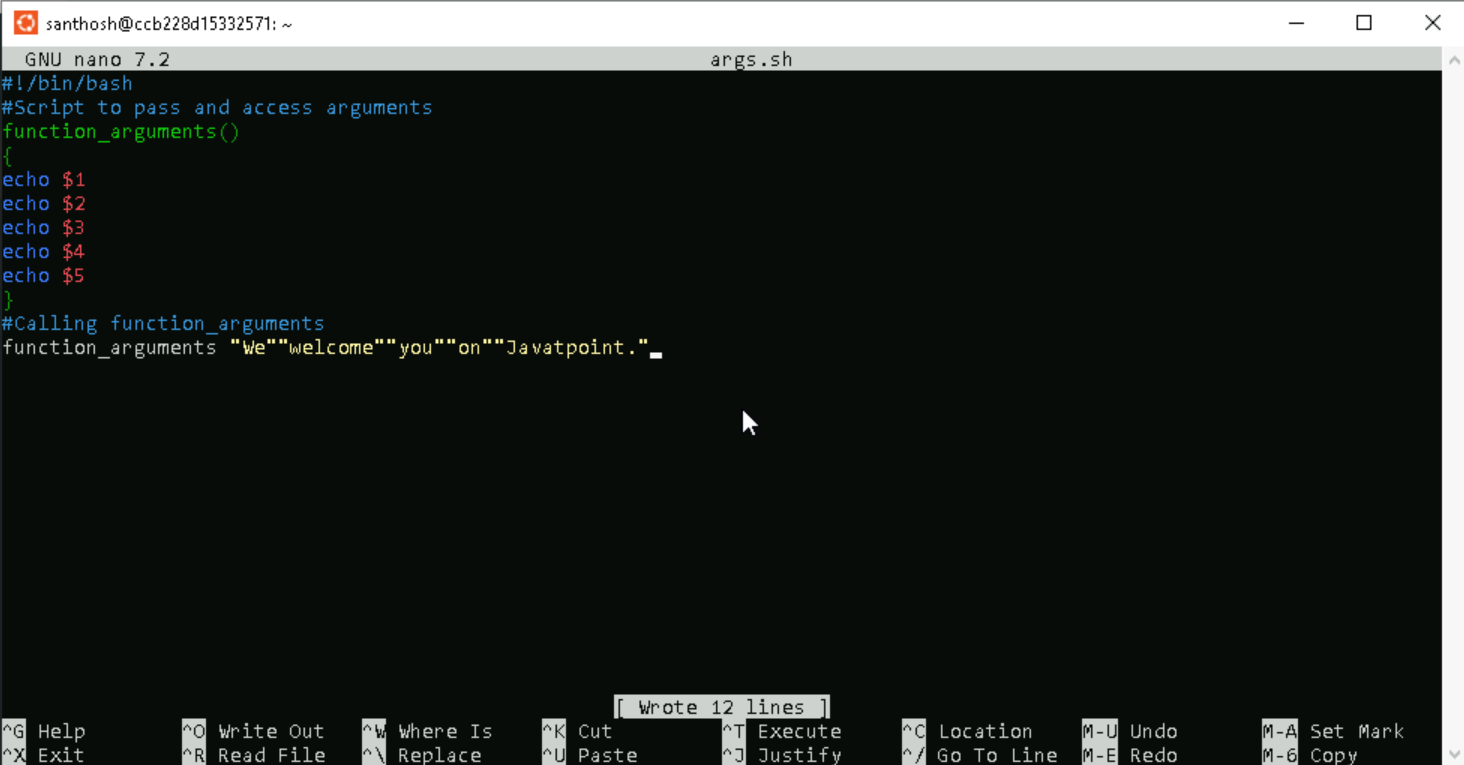
**Bash Functions**

Functions in bash scripting are a great option to reuse code. A Bash function can be defined as a set of commands which can be called several times within bash script. The purpose of function in bash is to help you make your scripts more readable and avoid writing the same code again and again.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



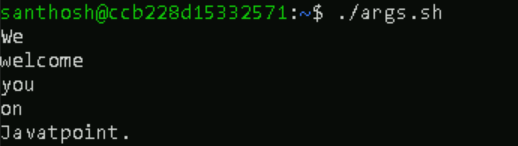
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.



**Variable Scope**

Global variables are defined as the variables which can be accessed anywhere within the script regardless of the scope. By default, all the variables are defined as global variables, even if they are declared inside the function. We can also create variables as a local variable. Local variables can be declared within the function body with the ?local? keyword when they are assigned for first time. They are only accessible inside that function.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



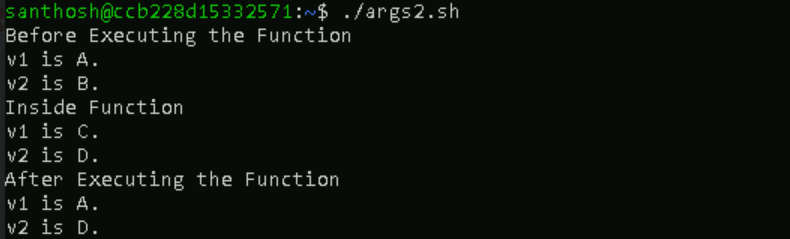
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

****

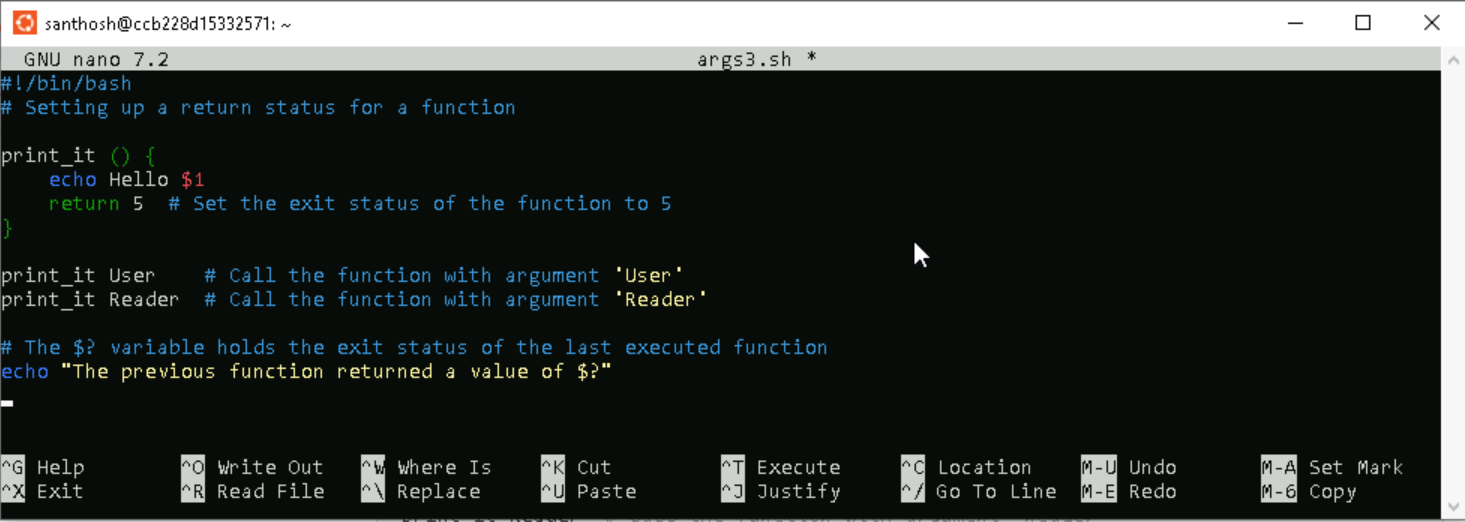
**Return Values**

**The return status can be indicated by using the 'return' keyword, and it is assigned to the variable $?. The return statement terminates the function and works as the function's exit status.**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

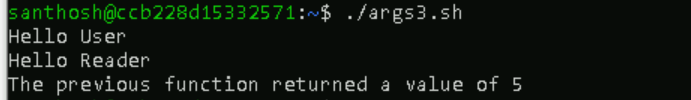
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

****

**Another better option to return a value from a function is to send the value to stdout using echo or printf commands, as shown below:**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

****

**Overriding Commands**

**In this example, we have overridden the 'echo' command and added the time stamp in the form of the argument to the 'echo' command.**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Bash Array**

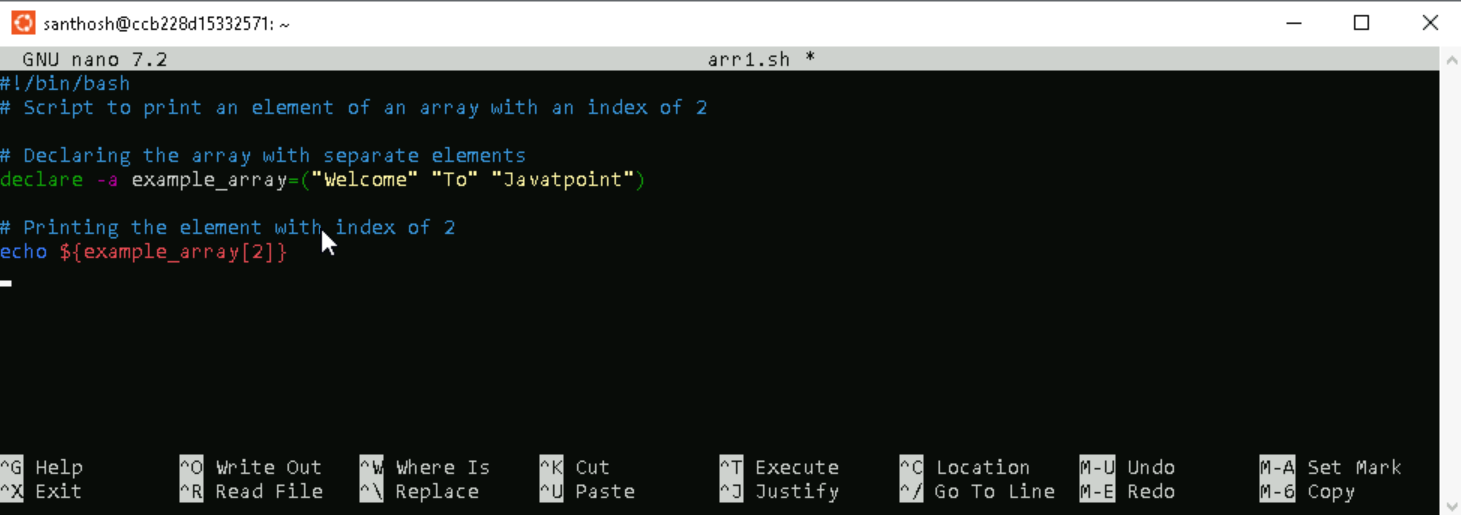
An array can be defined as a collection of similar type of elements. Unlike most of the programming languages, arrays in bash scripting need not be the collection of similar elements. Since Bash does not discriminate the string from a number, an array may contain both strings and numbers.

Example 1 : let's print an element of an array with an index of 2:

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

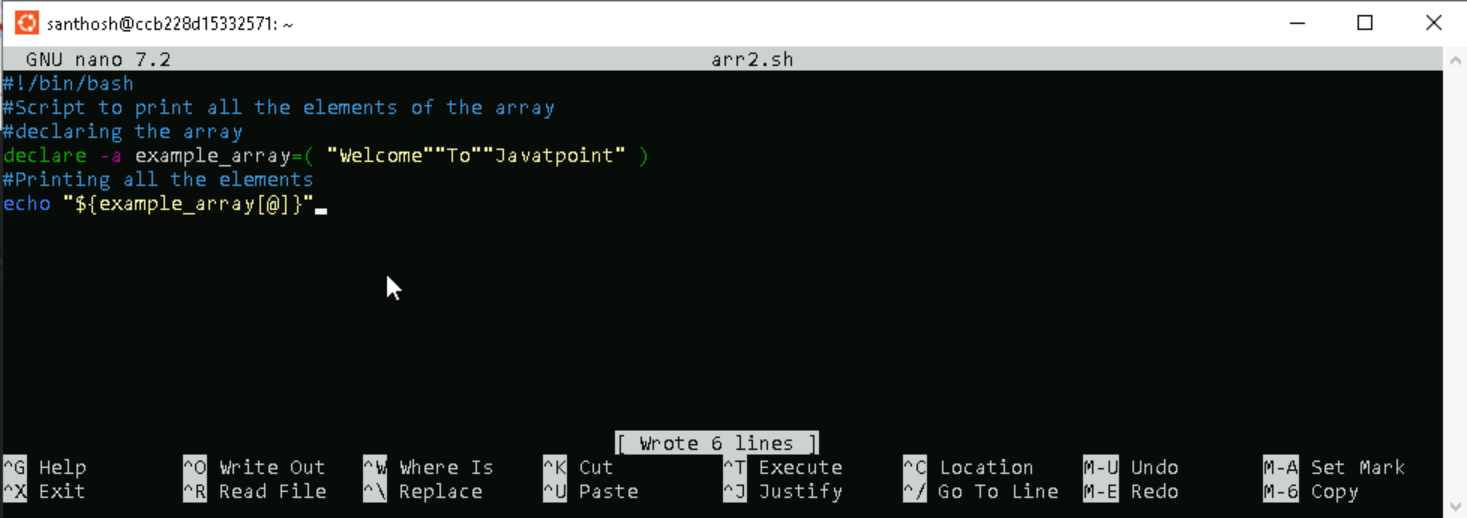


If we use @ or \* in the place of a specified index, it will expand to all members of the array. To print all the elements, we can use the following form:

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.



**Printing the Keys of an Array**

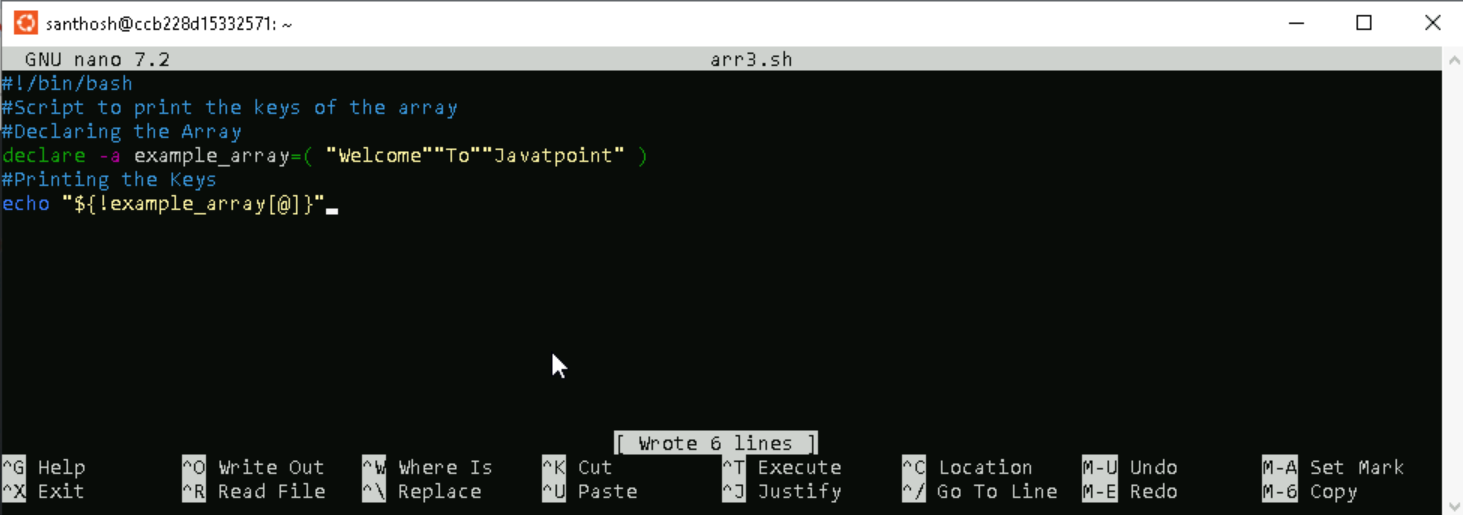
We can also retrieve and print the keys used in indexed or associative arrays, instead of their respective values. It can be performed by adding the ! operator before the array name as below:

1. ${!ARRAY\_NAME[index]}

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

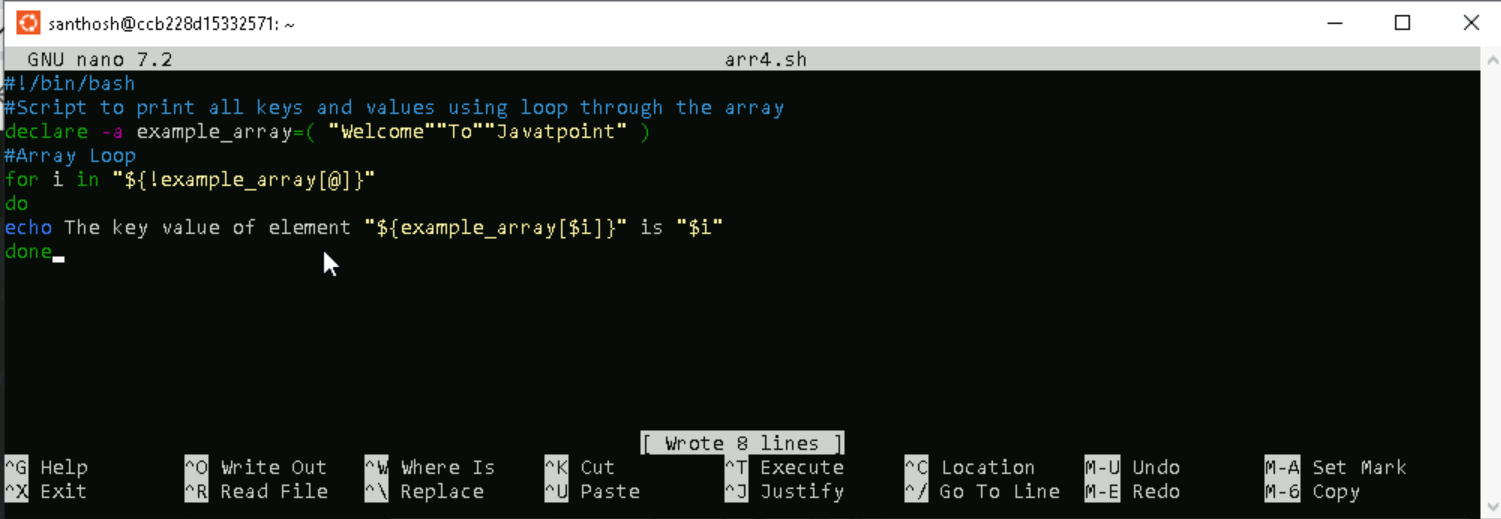
****

**Loop through the Array**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

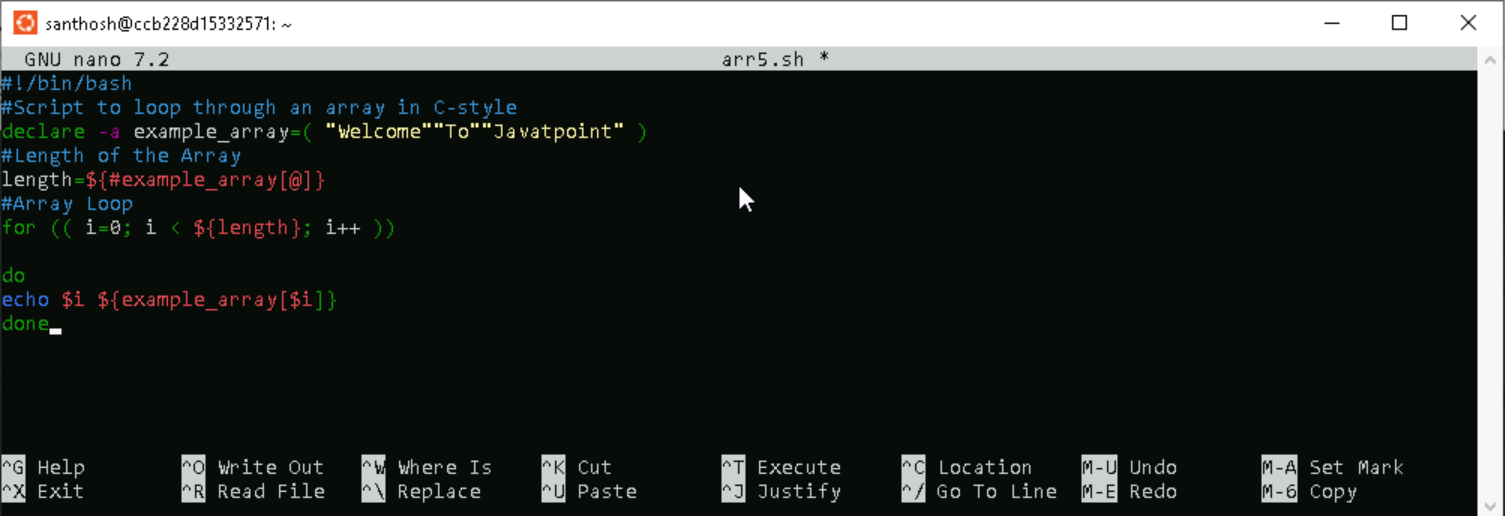


**Another common method to loop through an array is to retrieve the length of the array and use the C-style loop:**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



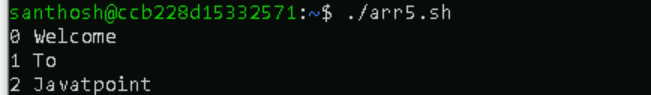
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



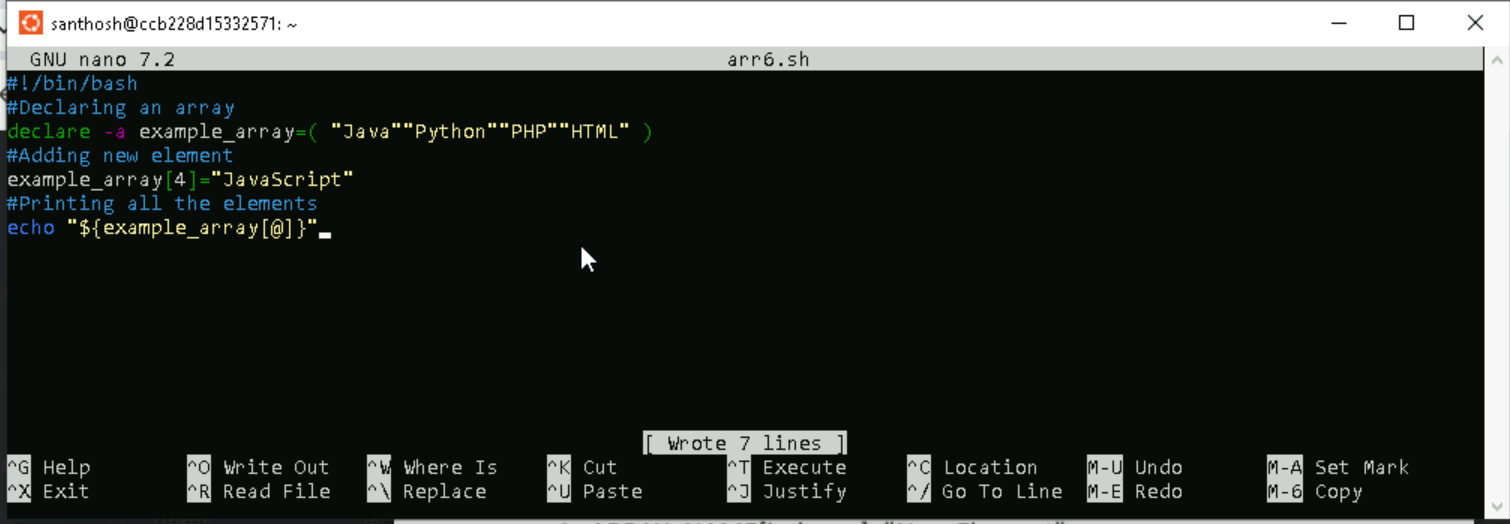
**Step 4:** Executing the output.

****

**Adding Elements to an Array**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output**.**

****

**Updating Array Element**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.



**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

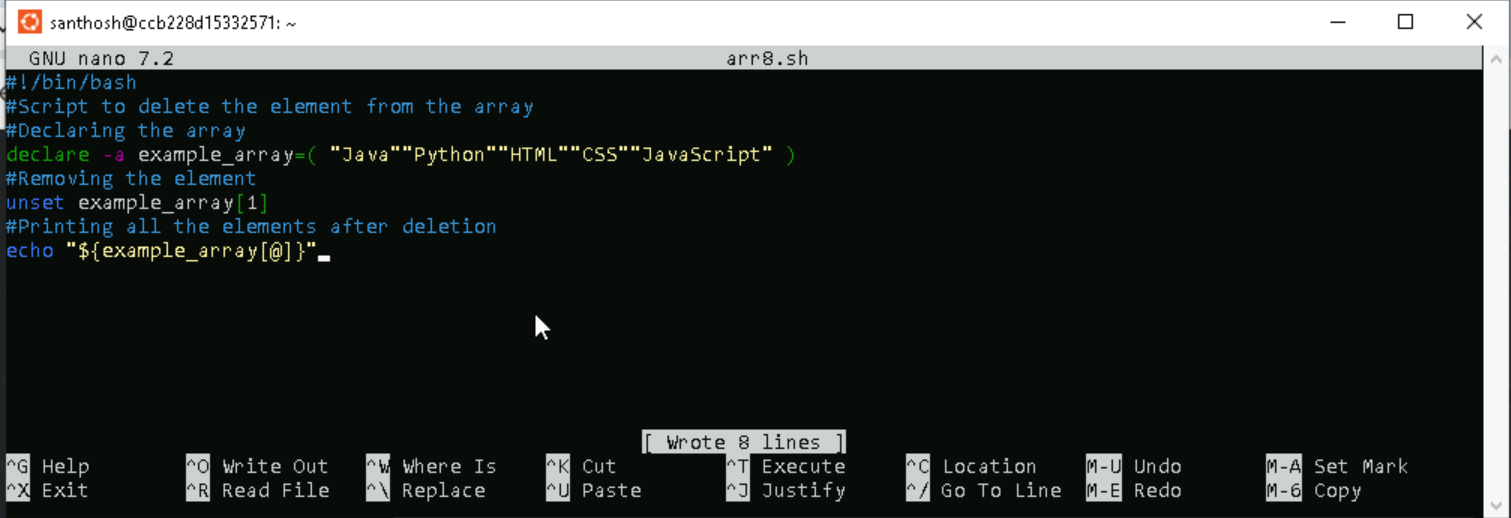


**Deleting an Element from an Array**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.



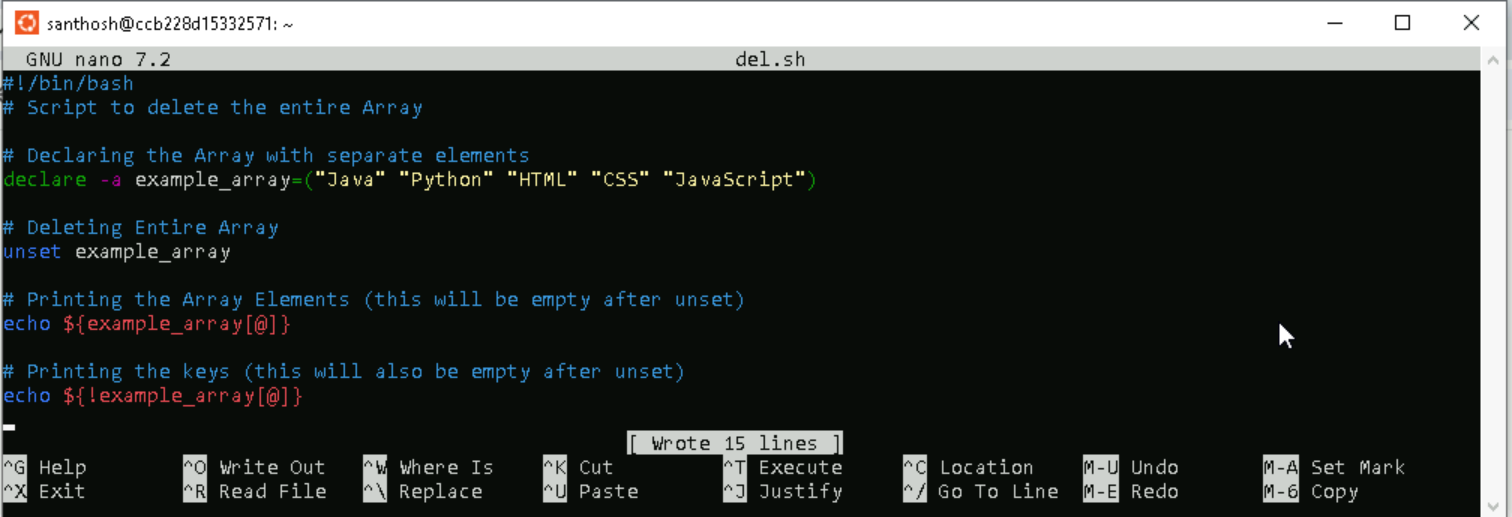
**Deleting the Entire Array**

Deleting an entire array is a very simple task. It can be performed by passing the array name as an argument to the 'unset' command without specifying the index or key.

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.

****

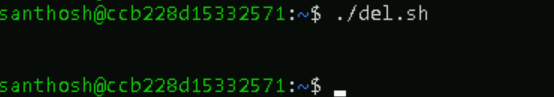
**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.

****

**Step 4:** Executing the output.

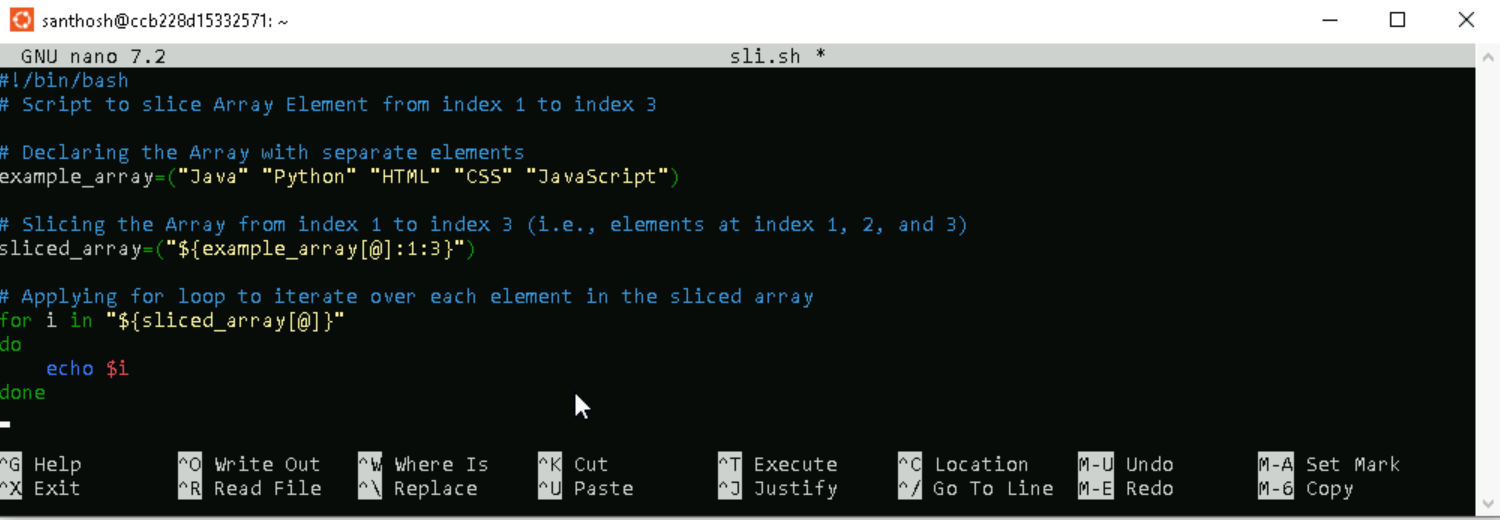
****

**Slice Array Elements**

**Step 1:** Creating a bash script using touch command and adding the script by editing the file using nano command.



**Step 2:** Creating the script for a simple scenario to demonstrate the use of the case statement.

****

**Step 3:** Providing the necessary permissions for the ex.sh script.



**Step 4:** Executing the output.

