

Arithmetic operations using Bash:

```
ssethumadv@019c021bbee4513: ~
GNU nano 7.2 arithmetic.sh *
# Division
echo "Division of x by y"
echo $(( $x / $y ))

# Exponentiation
echo "Exponentiation of x,y"
echo $(( $x ** $y ))

# Module
echo "Modular Division of x,y"
echo $(( $x % $y ))

# Incrementing x by 5
echo "Incrementing x by 5, then x= "
(( x += 5 ))
echo $x

# Decrementing x by 5
echo "Decrementing x by 5, then x= "
(( x -= 5 ))
echo $x

# Multiply x by 5
echo "Multiply of x by 5, then x="
(( x *= 5 ))
echo $x

# Dividing x by 5
echo "Dividing x by 5, x= "
(( x /= 5 ))
echo $x

# Remainder of Dividing x by 5
echo "Remainder of Dividing x by 5, x="
(( x %= 5 ))
echo $x

File Name to Write: arithmetic.sh
NG Help
NG Cancel
M-D DOS Format
M-M Mac Format
M-A Append
M-B Prepend
M-B Backup File
Browse
Type here to search
ENG US 14:58 24-01-2025
```

:

```
ssethumadv@019c021bbee4513: ~
ssethumadv@019c021bbee4513:~$ nano arithmetic.sh
ssethumadv@019c021bbee4513:~$ chmod a+x ar
arithmetic.sh arraysun.sh
ssethumadv@019c021bbee4513:~$ chmod a+x arithmetic.sh
ssethumadv@019c021bbee4513:~$ ./arithmetic.sh
x=8, y=2
Addition of x & y
10
Subtraction of x & y
6
Multiplication of x & y
16
Division of x by y
4
Exponentiation of x,y
64
Modular Division of x,y
0
Incrementing x by 5, then x=
13
Decrementing x by 5, then x=
8
Multiply of x by 5, then x=
40
Dividing x by 5, x=
0
Remainder of Dividing x by 5, x=
3
ssethumadv@019c021bbee4513:~$
```

Let command:

```
ssethumadv@019c021bbee4513: ~
GNU nano 7.2 letcommand.sh
#!/bin/bash

x=10
y=6
z=0

# Addition
echo "Addition"
let "z = $(( x + y ))"
echo "z = $z"

# Subtraction
echo "Subtraction"
let "z = $(( x - y ))"
echo "z = $z"

# Multiplication
echo "Multiplication"
let "z = $(( x * y ))"
echo "z = $z"

# Division
echo "Division"
let "z = $(( x / y ))"
echo "z = $z"

# Exponentiation
echo "Exponentiation"
let "z = $(( x ** y ))"
echo "z = $z"

# Modular Division
echo "Modular Division"
let "z = $(( x % y ))"
echo "z = $z"

# Incrementing x by 5

NG Help      NO Write Out  NV Where Is  NU Cut       [ Read 61 lines ]  V Location  W-U Undo     W-A Set Mark  W-] To Bracket
OX Exit      NR Read File  NR Replace  NU Paste     Execute        VV Go To Line  W-E Redo     W-G Copy     W-; Where Was
```

```
ssethumadv@019c021bbee4513: ~
8
Multiply of x by 5, then x=
10
Dividing x by 5, x=
8
Remainder of Dividing x by 5, x=
3
ssethumadv@019c021bbee4513:~$ nano arithmetic.sh
ssethumadv@019c021bbee4513:~$ nano letcommand.sh
ssethumadv@019c021bbee4513:~$ chmod a+x letcommand.sh
ssethumadv@019c021bbee4513:~$ ./letcommand.sh
Addition
z = 16
Subtraction
z = 4
Multiplication
z = 60
Division
z = 1
Exponentiation
z = 1000000
Modular Division
z = 4
Incrementing x by 5, then x=
15
Decrementing x by 5, then x=
10
Multiply of x by 5, then x=
50
Dividing x by 5, x=
10
Remainder of Dividing x by 5, x=
0
ssethumadv@019c021bbee4513:~$
```

Expr command:

```
ssethumadav@019c021bbee4513: ~
GNU nano 7.2                                expr.sh *
#!/bin/bash

# Basic arithmetic using expr
echo "a=10, b=3"
echo "c is the value of addition c=a+b"
a=10
b=3
echo "c= $(expr $a + $b)"

echo "c is the value of subtraction c=a-b"
echo "c= $(expr $a - $b)"

echo "c is the value of multiplication c=a*b"
echo "c= $(expr $a \* $b)"

echo "c is the value of division c=a/b"
echo "c= $(expr $a / $b)"

echo "c is the value of modulo c=a%b"
echo "c= $(expr $a % $b)"

Multiplication
z = 60
Division
z = 1
Exponentiation
z = 1000000
Modular Division
z = 4
Incrementing x by 5, then x=
15
Decrementing x by 5, then x=
10
Multiply of x by 5, then x=
50
Dividing x by 5, x=
10
Remainder of Dividing x by 5, x=
0
ssethumadav@019c021bbee4513:~$ nano letcommand.sh
ssethumadav@019c021bbee4513:~$ nano expr.sh
ssethumadav@019c021bbee4513:~$ chmod a+x expr.sh
ssethumadav@019c021bbee4513:~$ ./expr.sh
a=10, b=3
c is the value of addition c=a+b
c= 13
c is the value of subtraction c=a-b
c= 7
c is the value of multiplication c=a*b
c= 30
c is the value of division c=a/b
c= 3
c is the value of modulo c=a%b
c= 1
ssethumadav@019c021bbee4513:~$
```

1. Check nano: (to check a number greater than 125)



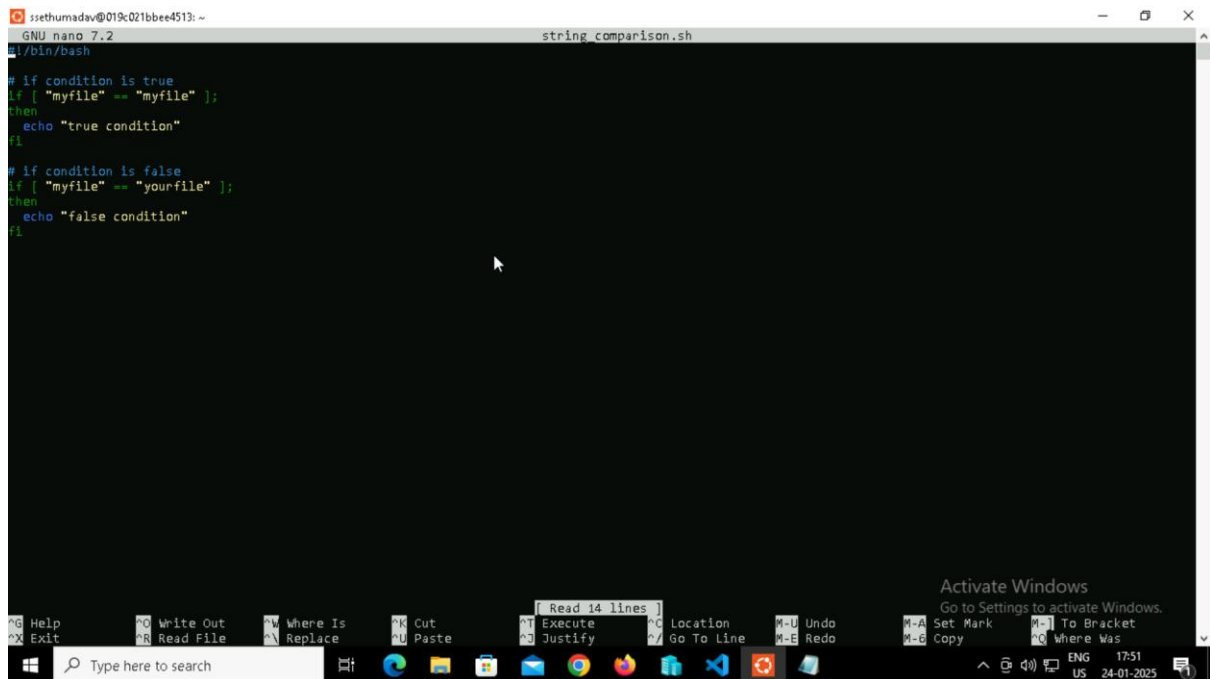
```
GNU nano 7.2 check_num.sh
#!/bin/bash
read -p "Enter number: " number
if [ $number -gt 125 ]
then
echo "Value is greater than 125"
fi
```

Activate Windows
Go to Settings to activate Windows.

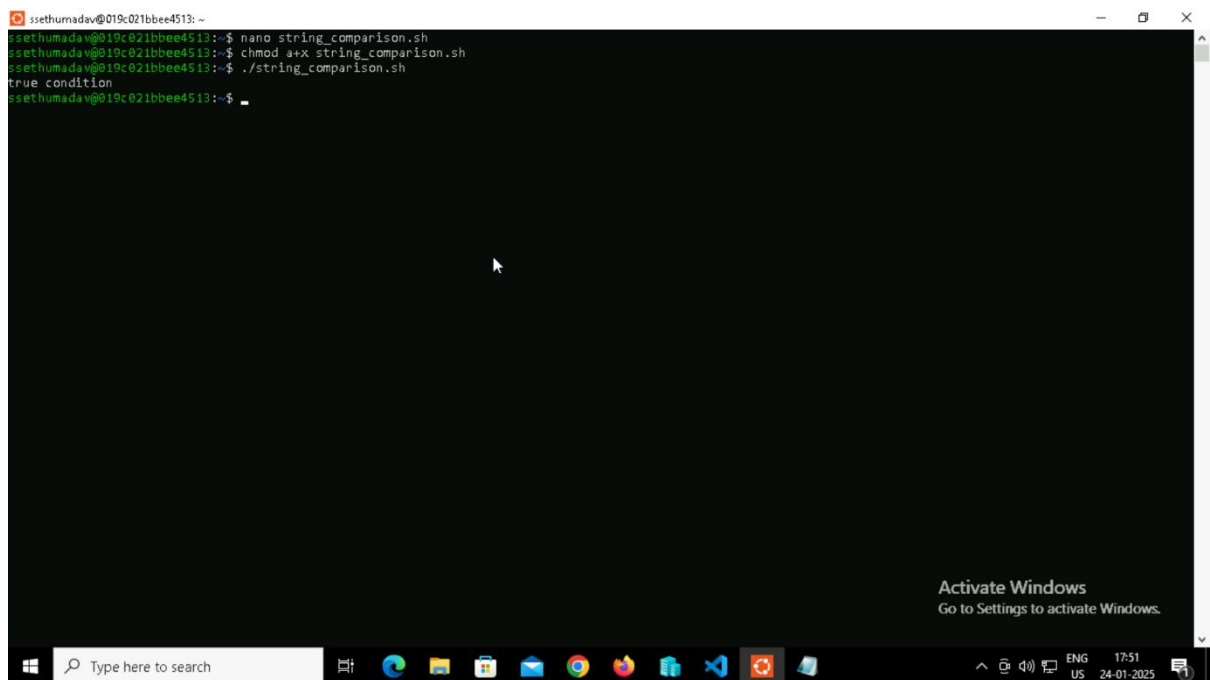
```
ssethumadav@019c021bbee4513: ~
ssethumadav@019c021bbee4513:~$ nano check_num.sh
ssethumadav@019c021bbee4513:~$ chmod a+x check_num.sh
ssethumadav@019c021bbee4513:~$ ./check_num.sh
Enter number: 211
Value is greater than 125
ssethumadav@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.

2.String comparison:



```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2 string_comparison.sh  
#!/bin/bash  
  
# if condition is true  
if [ "myfile" == "myfile" ];  
then  
    echo "true condition"  
fi  
  
# if condition is false  
if [ "myfile" == "yourfile" ];  
then  
    echo "false condition"  
fi
```



```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano string_comparison.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x string_comparison.sh  
ssethumadv@019c021bbee4513:~$ ./string_comparison.sh  
true condition  
ssethumadv@019c021bbee4513:~$ _
```

3.Number Comparison:

```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2 number_comparison.sh  
#!/bin/bash  
  
#if condition (greater than) is true  
if [ 10 -gt 3 ];  
then  
    echo "10 is greater than 3."  
fi  
  
#if condition (greater than) is false  
if [ 3 -gt 10 ];  
then  
    echo "3 is not greater than 10."  
fi  
  
#if condition (lesser than) is true  
if [ 3 -lt 10 ];  
then  
    echo "3 is less than 10."  
fi  
  
#if condition (lesser than) is false  
if [ 10 -lt 3 ];  
then  
    echo "10 is not less than 3."  
fi  
  
#if condition (equal to) is true  
if [ 10 -eq 10 ];  
then  
    echo "10 is equal to 10."  
fi  
  
#if condition (equal to) is false  
if [ 10 -eq 9 ];  
then  
    echo "10 is not equal to 9"  
fi  
  
Read 38 lines  
Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket  
Exit Read File Replace Paste Justify Go To Line Redo Copy Copy Where Was
```

```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano number_comparison.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x number_comparison.sh  
ssethumadv@019c021bbee4513:~$ ./number_comparison.sh  
10 is greater than 3.  
3 is less than 10.  
10 is equal to 10.  
ssethumadv@019c021bbee4513:~$
```

4.Using and operator:

```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2                                and_operator.sh  
#!/bin/bash  
  
# TRUE && TRUE  
if [ 8 -gt 6 ] && [ 10 -eq 10 ];  
then  
    echo "Conditions are true"  
fi  
  
# TRUE && FALSE  
if [ "mylife" == "mylife" ] && [ 3 -gt 10 ];  
then  
    echo "Conditions are false"  
fi
```

Activate Windows
Go to Settings to activate Windows.

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark Copy To Bracket Where Was

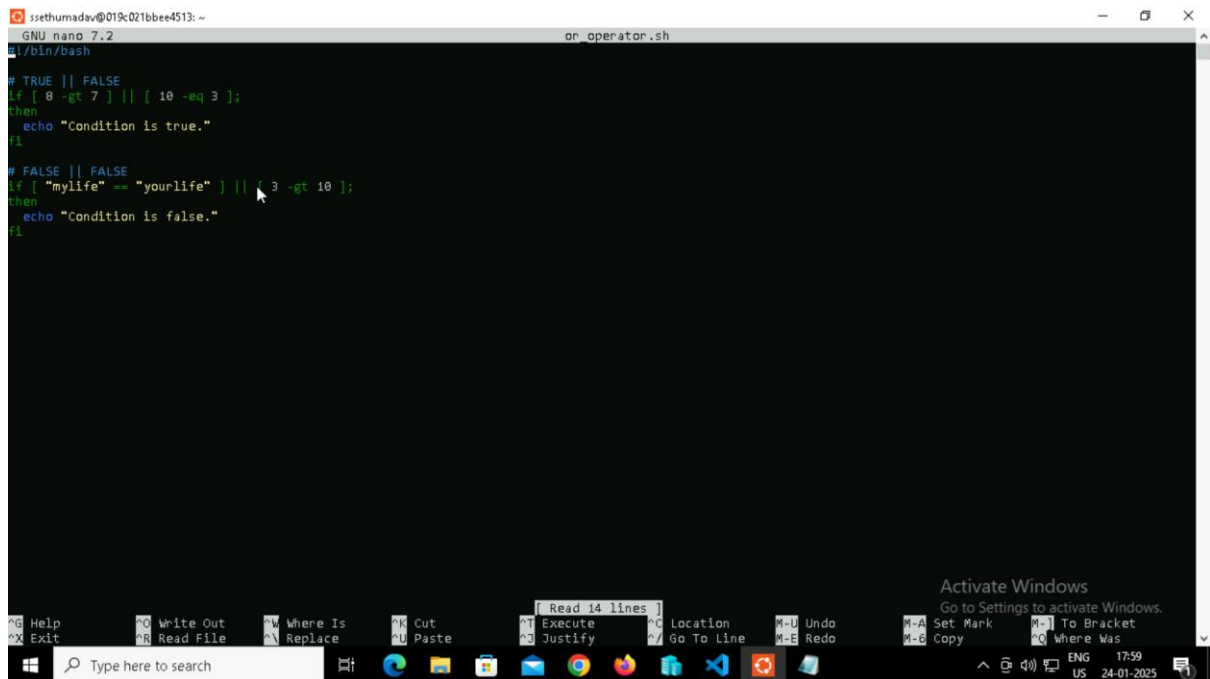
Type here to search

```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano and_operator.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x a  
and_operator.sh arithmetic.sh arraysum.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x and_operator.sh  
ssethumadv@019c021bbee4513:~$ ./and_operator.sh  
Conditions are true  
ssethumadv@019c021bbee4513:~$
```

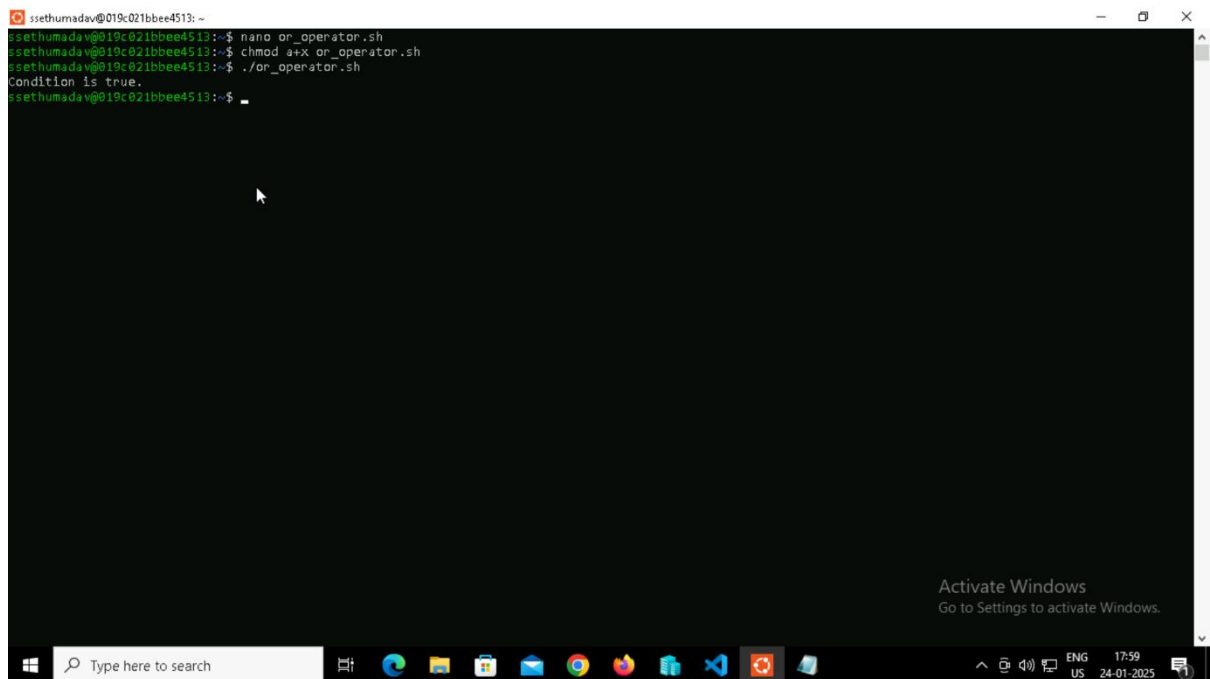
Activate Windows
Go to Settings to activate Windows.

Type here to search

5.Using Or operator:



```
ssethumadav@019c021bbee4513: ~  
GNU nano 7.2 or_operator.sh  
#!/bin/bash  
  
# TRUE || FALSE  
if [ 8 -gt 7 ] || [ 10 -eq 3 ];  
then  
    echo "Condition is true."  
fi  
  
# FALSE || FALSE  
if [ "mylife" == "yourlife" ] || [ 3 -gt 10 ];  
then  
    echo "Condition is false."  
fi
```



```
ssethumadav@019c021bbee4513: ~  
ssethumadav@019c021bbee4513:~$ nano or_operator.sh  
ssethumadav@019c021bbee4513:~$ chmod a+x or_operator.sh  
ssethumadav@019c021bbee4513:~$ ./or_operator.sh  
Condition is true.  
ssethumadav@019c021bbee4513:~$
```


6.and_or_operator:

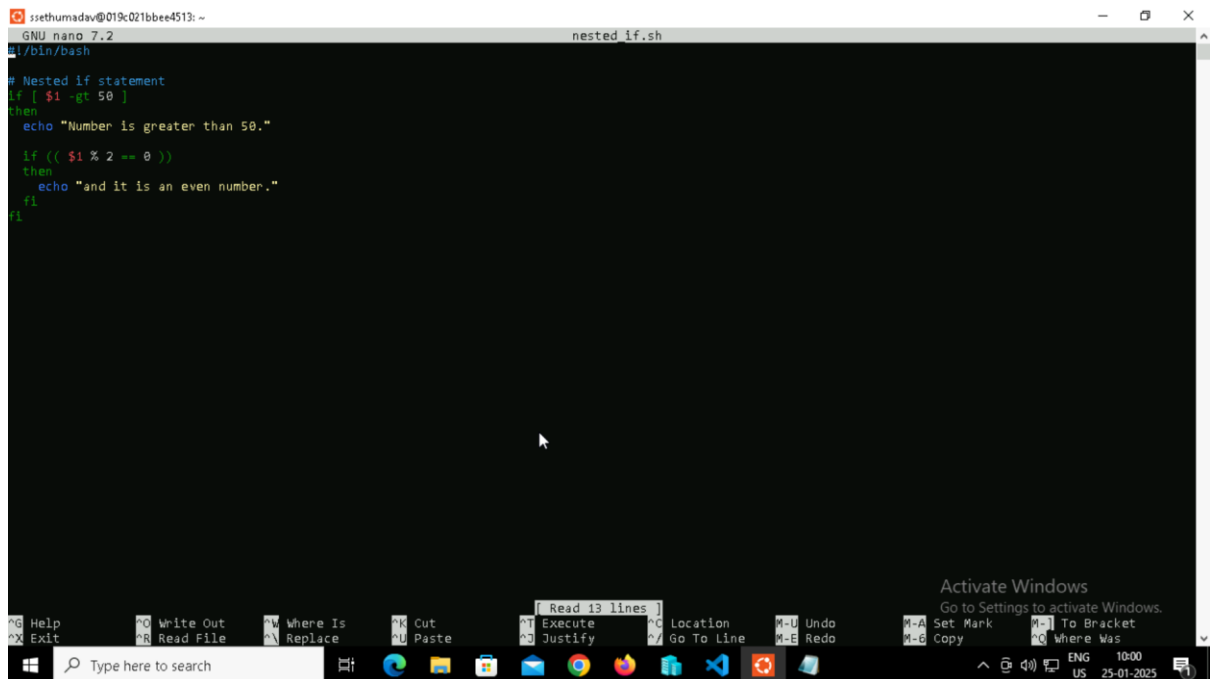
```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2                                and_or_operator.sh  
#!/bin/bash  
  
# TRUE && FALSE || FALSE || TRUE  
if [[ 10 -eq 10 && 5 -gt 4 || 3 -eq 4 || 3 -lt 6 ]];  
then  
    echo "Condition is true."  
fi  
  
# TRUE && FALSE || FALSE  
if [[ 8 -eq 0 && 8 -gt 10 || 9 -lt 5 ]];  
then  
    echo "Condition is false"  
fi
```

Activate Windows
Go to Settings to activate Windows.

```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano or_operator.sh  
ssethumadv@019c021bbee4513:~$ chmod +x or_operator.sh  
ssethumadv@019c021bbee4513:~$ ./or_operator.sh  
Condition is true.  
ssethumadv@019c021bbee4513:~$ nano or_operator.sh  
ssethumadv@019c021bbee4513:~$ nano and_or_operator.sh  
ssethumadv@019c021bbee4513:~$ chmod +x and_or_operator.sh  
ssethumadv@019c021bbee4513:~$ ./and_or_operator.sh  
Condition is true.  
ssethumadv@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.

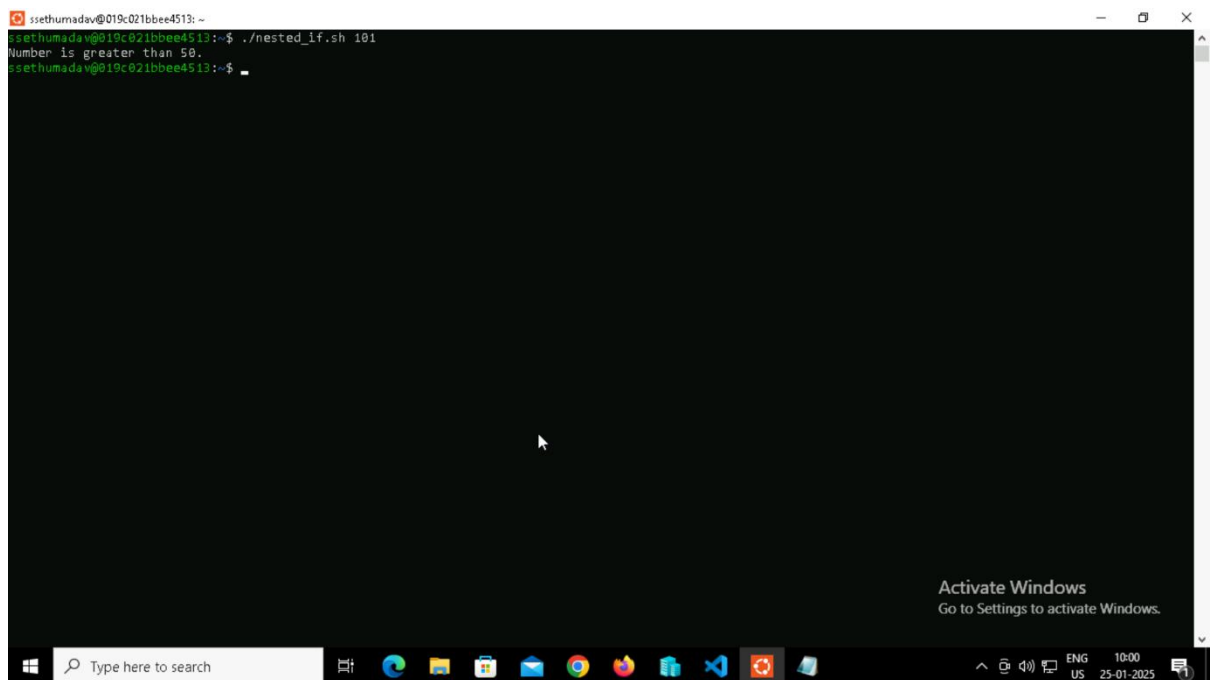
7.Nested_If:



The screenshot shows a Windows terminal window with the nano editor open. The editor is editing a file named 'nested_if.sh'. The script content is as follows:

```
#!/bin/bash
# Nested if statement
if [ $1 -gt 50 ]
then
    echo "Number is greater than 50."
    if (( $1 % 2 == 0 ))
    then
        echo "and it is an even number."
    fi
fi
```

The terminal window title is 'nested_if.sh'. The nano editor's status bar at the bottom shows 'GNU nano 7.2' and 'nested_if.sh'. The Windows taskbar is visible at the bottom, showing the search bar and various application icons. An 'Activate Windows' watermark is present in the bottom right corner of the terminal window.



The screenshot shows the same Windows terminal window after the script has been executed. The prompt is now 'ssethumada@019c021bbee4513:~\$'. The command entered was './nested_if.sh 101', and the output displayed is 'Number is greater than 50.' followed by a new prompt 'ssethumada@019c021bbee4513:~\$'.

Example 1: (simple if)

```
ssethumadav@019c021bbee4513: ~  
GNU nano 7.2                                one.sh  
#!/bin/bash  
# when the condition is true  
if [ 10 -gt 3 ]; then  
    echo "10 is greater than 3."  
else  
    echo "10 is not greater than 3."  
fi  
  
# when the condition is false  
if [ 3 -gt 10 ]; then  
    echo "3 is greater than 10."  
else  
    echo "3 is not greater than 10."  
fi
```

Activate Windows
Go to Settings to activate Windows.

```
ssethumadav@019c021bbee4513: ~  
ssethumadav@019c021bbee4513:~$ nano one.sh  
ssethumadav@019c021bbee4513:~$ chmod a+x o  
one.sh  
ssethumadav@019c021bbee4513:~$ chmod a+x one.sh  
ssethumadav@019c021bbee4513:~$ ./one.sh  
10 is greater than 3.  
3 is not greater than 10.  
ssethumadav@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.

Example 2: multiple conditions with ifelse:

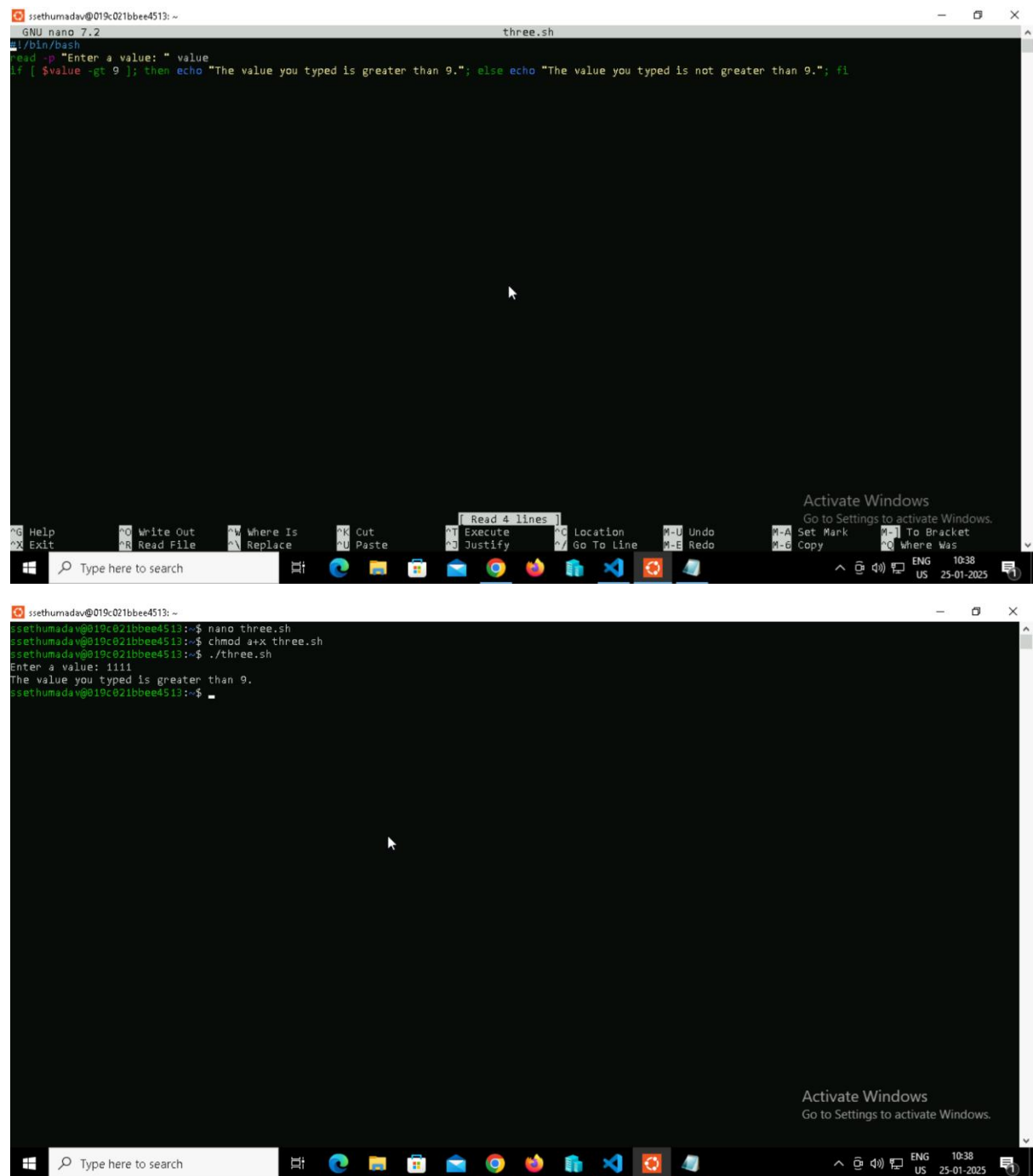
```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2 two.sh  
#!/bin/bash  
# When condition is true  
if [[ 10 -gt 9 && 10 == 9 || 2 -lt 1 || 25 -gt 20 ]]; then  
    echo "Given condition is true."  
else  
    echo "Given condition is false."  
fi  
  
# When condition is false  
if [[ 10 -gt 9 && 10 == 8 || 3 -gt 4 || 8 -gt 8 ]]; then  
    echo "Given condition is true."  
else  
    echo "Given condition is not true."  
fi
```

Activate Windows
Go to Settings to activate Windows.

```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano two.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x two.sh  
ssethumadv@019c021bbee4513:~$ ./two.sh  
Given condition is true.  
Given condition is not true.  
ssethumadv@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.

Example 3: ifelse in single line:



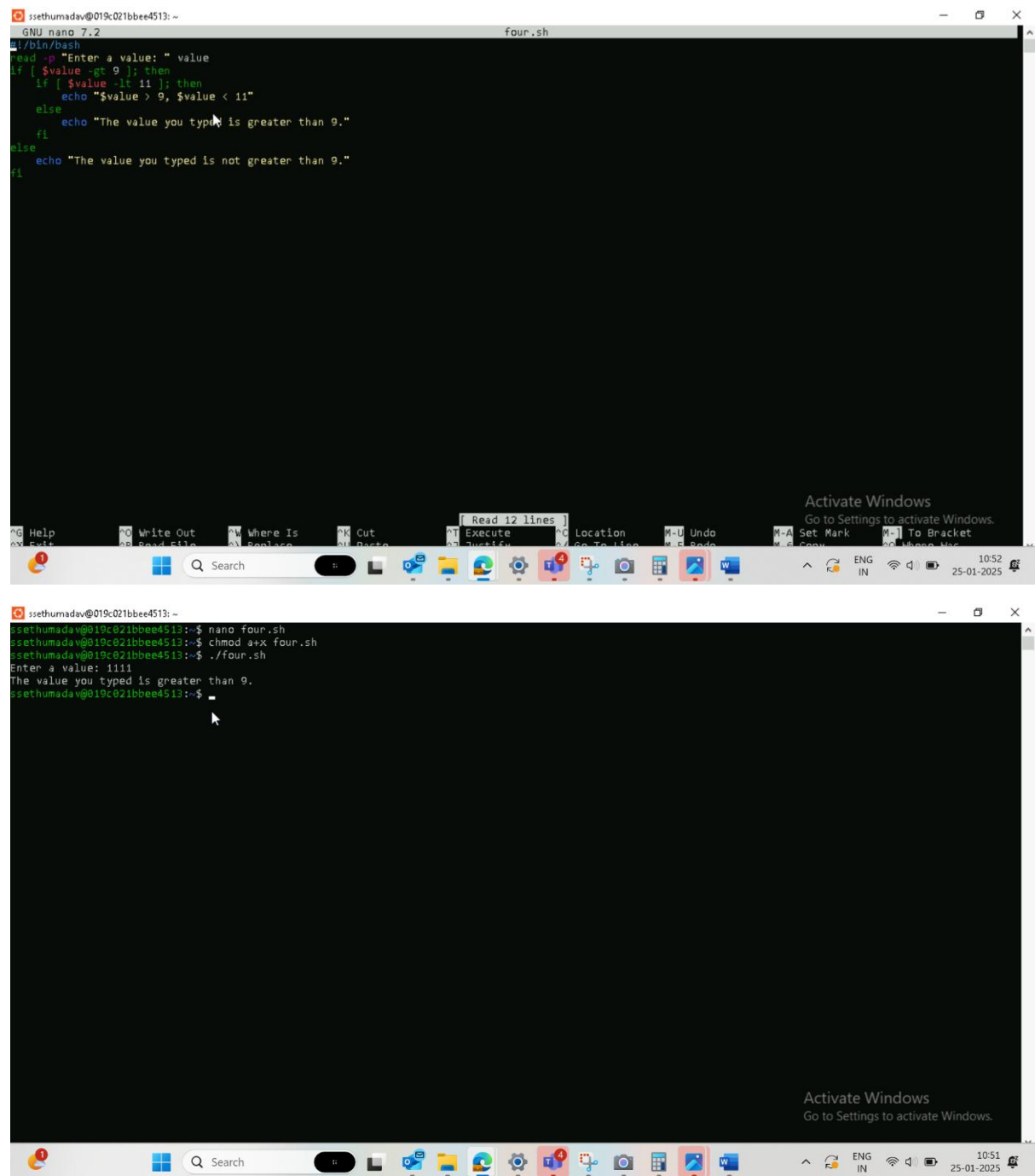
The image consists of two screenshots of a Windows terminal window. The top screenshot shows the nano text editor editing a file named 'three.sh'. The script content is as follows:

```
GNU nano 7.2 three.sh
#!/bin/bash
read -p "Enter a value: " value
if [ $value -gt 9 ]; then echo "The value you typed is greater than 9."; else echo "The value you typed is not greater than 9."; fi
```

The bottom screenshot shows the terminal after the script has been executed. The user has run 'nano three.sh', 'chmod +x three.sh', and './three.sh'. The prompt 'Enter a value:' is followed by the input '1111', and the output is 'The value you typed is greater than 9.'.

```
ssethumadv@019c021bbee4513: ~
ssethumadv@019c021bbee4513:~$ nano three.sh
ssethumadv@019c021bbee4513:~$ chmod +x three.sh
ssethumadv@019c021bbee4513:~$ ./three.sh
Enter a value: 1111
The value you typed is greater than 9.
ssethumadv@019c021bbee4513:~$
```

Example 4:nested ifelse:



The image consists of two screenshots of a Windows terminal window. The top screenshot shows the creation of a script file named `four.sh` using the `nano` text editor. The script contains a nested if-else structure that prompts the user for a value and checks if it is greater than 9. The bottom screenshot shows the execution of the script, where the user enters the value `1111`, and the script outputs `The value you typed is greater than 9.`

```
ssethumadv@019c021bbee4513: ~  
GNU nano 7.2 four.sh  
#!/bin/bash  
read -p "Enter a value: " value  
if [ $value -gt 9 ]; then  
    if [ $value -lt 11 ]; then  
        echo "$value > 9, $value < 11"  
    else  
        echo "The value you typed is greater than 9."  
    fi  
else  
    echo "The value you typed is not greater than 9."  
fi
```

```
ssethumadv@019c021bbee4513: ~  
ssethumadv@019c021bbee4513:~$ nano four.sh  
ssethumadv@019c021bbee4513:~$ chmod a+x four.sh  
ssethumadv@019c021bbee4513:~$ ./four.sh  
Enter a value: 1111  
The value you typed is greater than 9.  
ssethumadv@019c021bbee4513:~$
```

Example 5:Elif:

```
GNU nano 7.2 five.sh
#!/bin/bash
read -p "Enter a number of quantity: " num

if [ $num -gt 100 ]; then
    echo "Eligible for 10% discount"
elif [ $num -lt 100 ]; then
    echo "Eligible for 5% discount"
else
    echo "Lucky Draw Winner"
    echo "Eligible to get the item for free"
fi
```

Activate Windows
Go to Settings to activate Windows.

```
ssethumadv@019c021bbee4513: ~
ssethumadv@019c021bbee4513:~$ nano five.sh
ssethumadv@019c021bbee4513:~$ chmod a+x five.sh
ssethumadv@019c021bbee4513:~$ ./five.sh
Enter a number of quantity: 10
Eligible for 5% discount
ssethumadv@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.

Example 6: Elif with multiple conditions

```
ssethumadav@019c021bbee4513: ~  
GNU nano 7.2 six.sh  
#!/bin/bash  
read -p "Enter a number of quantity: " num  
  
if [ $num -gt 200 ]; then  
    echo "Eligible for 20% discount"  
elif [[ $num -eq 200 || $num -eq 100 ]]; then  
    echo "Lucky Draw Winner"  
    echo "Eligible to get the item for free"  
elif [ $num -gt 100 ] && [ $num -lt 200 ]; then  
    echo "Eligible for 10% discount"  
elif [ $num -lt 100 ]; then  
    echo "No discount"  
fi
```

Activate Windows
Go to Settings to activate Windows.

```
ssethumadav@019c021bbee4513: ~  
ssethumadav@019c021bbee4513:~$ nano six.sh  
ssethumadav@019c021bbee4513:~$ chmod a+x six.sh  
ssethumadav@019c021bbee4513:~$ ./six.sh  
Enter a number of quantity: 1111  
Eligible for 20% discount  
ssethumadav@019c021bbee4513:~$
```

Activate Windows
Go to Settings to activate Windows.