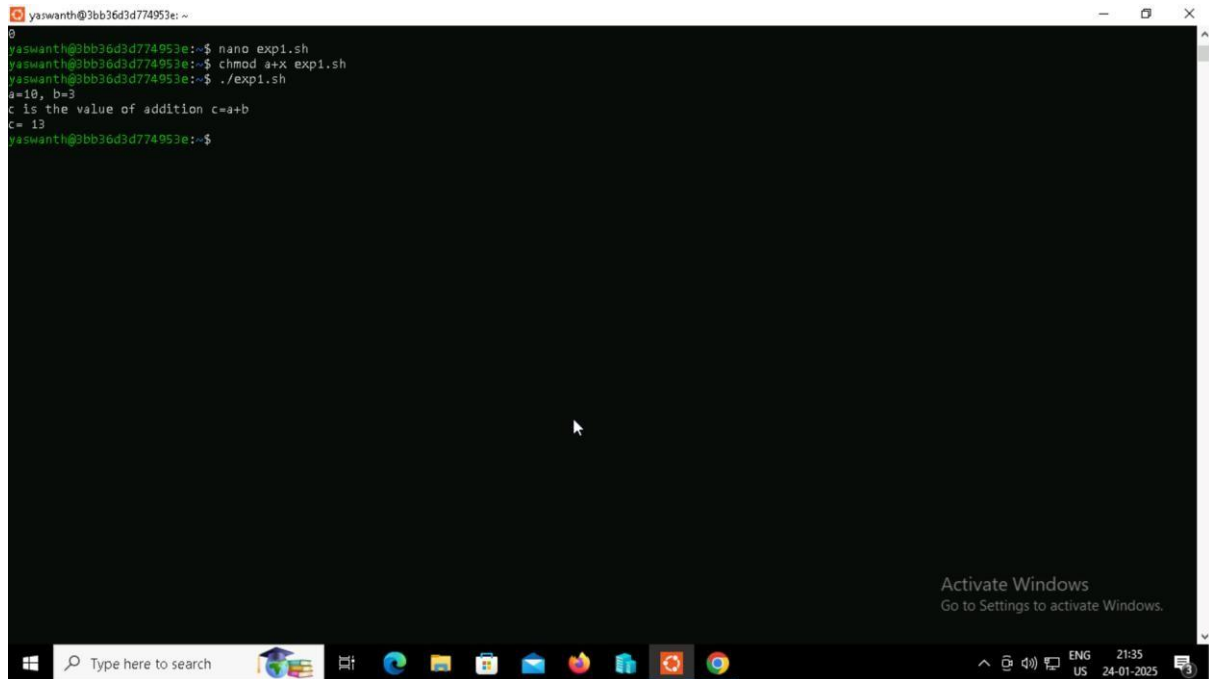


# BASH SCRIPTING PROGRAMS

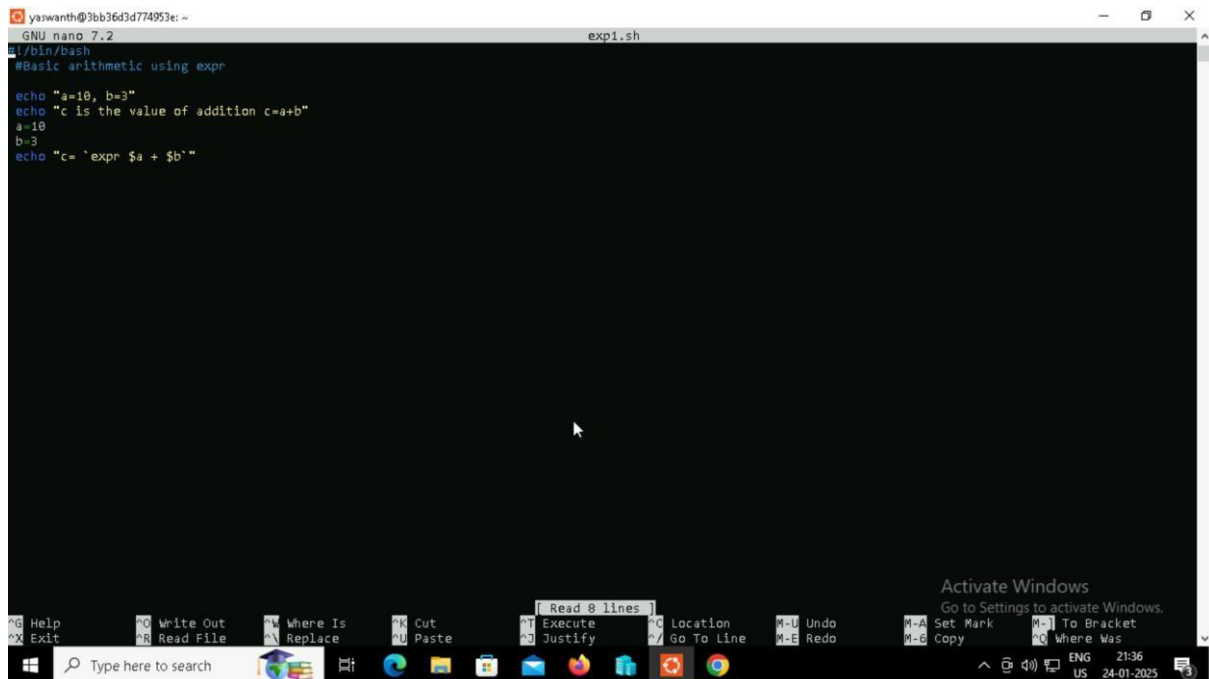
## 1.BASH SCRIPT PROGRAM FOR ADDITION OF TWO NUMBERS USING 'EXPR':

```
yaswanth@3bb36d3d774953e: ~  
$ nano exp1.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x exp1.sh  
yaswanth@3bb36d3d774953e:~$ ./exp1.sh  
a=10, b=3  
c is the value of addition c=a+b  
c= 13  
yaswanth@3bb36d3d774953e:~$
```



## NANO FOR EDIT AND MANAGE CODE :

```
GNU nano 7.2 exp1.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`"
```



## 2.ARITHMETIC OPERATIONS USING TWO NUMBERS:

```
yaswanth@3bb36d3d774953e:~$ nano arithmetic.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x arithmetic.sh
yaswanth@3bb36d3d774953e:~$ ./arithmetic.sh
a=4 , b=5
Addition of a and b :
9
Subtraction of a and b :
-1
Multiplication of a and b :
20
Division a and b:
0
Modulus a and b:
4
Incrementing a :
5
decremneting b :
4
expo a and b :
1024
dividing a by 4
0
yaswanth@3bb36d3d774953e:~$
```

Activate Windows  
Go to Settings to activate Windows.

```
GNU nano 7.2 arithmetic.sh
#!/bin/bash
a=4
b=5
echo "a=4 , b=5"
echo "Addition of a and b :"
echo " $((a + b))"
echo "Subtraction of a and b :"
echo " $((a - b))"
echo "Multiplication of a and b :"
echo " $((a * b))"
echo "Division a and b:"
echo " $((a / b))"
echo "Modulus a and b:"
echo " $((a % b))"
echo "Incrementing a :"
echo " $((a+1))"
echo "decremneting b :"
echo " $((b-1))"
echo "expo a and b :"
echo " $((a ** b))"
echo "dividing a by 4"
echo " $((a/5))"
```

Read 24 lines

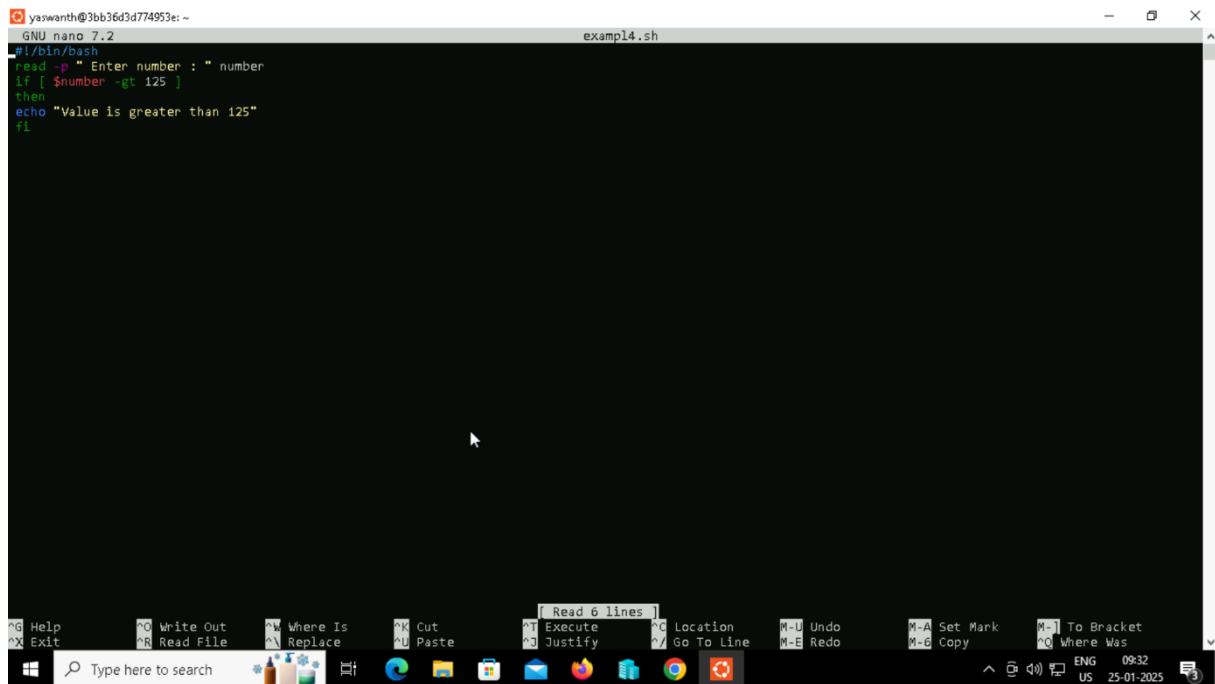
Activate Windows  
Go to Settings to activate Windows.

### 3.ARITHMETIC OPERATIONS USING 3 NUMBERS :

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano arthi2.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x arthi2.sh  
yaswanth@3bb36d3d774953e:~$ ./arthi2.sh  
Addition  
z = 16  
Substraction  
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  
yaswanth@3bb36d3d774953e:~$
```

```
GNU nano 7.2 arthi2.sh  
#!/bin/bash  
  
x=10  
y=6  
z=0  
  
echo "Addition"  
let "z = $(x + y)"  
echo "z= $z"  
  
echo "Substraction"  
let "z = $(x - y)"  
echo "z= $z"  
  
echo "Multiplication"  
let "z = $(x * y)"  
echo "z = $z"  
  
echo "Division"  
let "z = $(x / y)"  
echo "z = $z"  
  
echo "Exponentiation"  
let "z = $(x ** y)"  
echo "z = $z"  
  
echo "Modular Division"  
let "z = $(x % y)"  
echo "z = $z"  
  
let "x += 5"  
echo "Incrementing x by 5, then x= "  
echo $x  
  
let "x -= 5"  
echo "Decrementing x by 5, then x= "  
echo $x
```

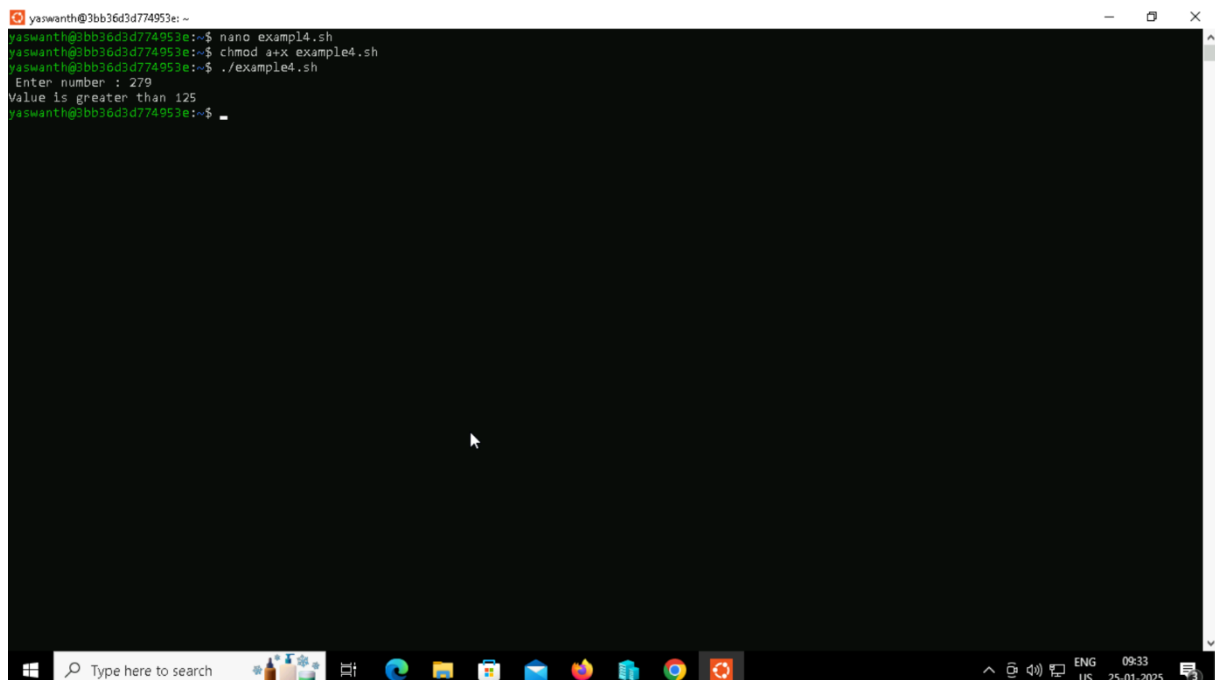
4 .In this example, take a user-input of any number and check if the value is greater than 125.



The screenshot shows a terminal window with the nano 7.2 editor open. The file being edited is 'examp14.sh'. The script content is as follows:

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

The terminal window has a standard Linux desktop environment with a taskbar at the bottom showing various application icons and system status information (ENG US, 09:32, 25-01-2025).

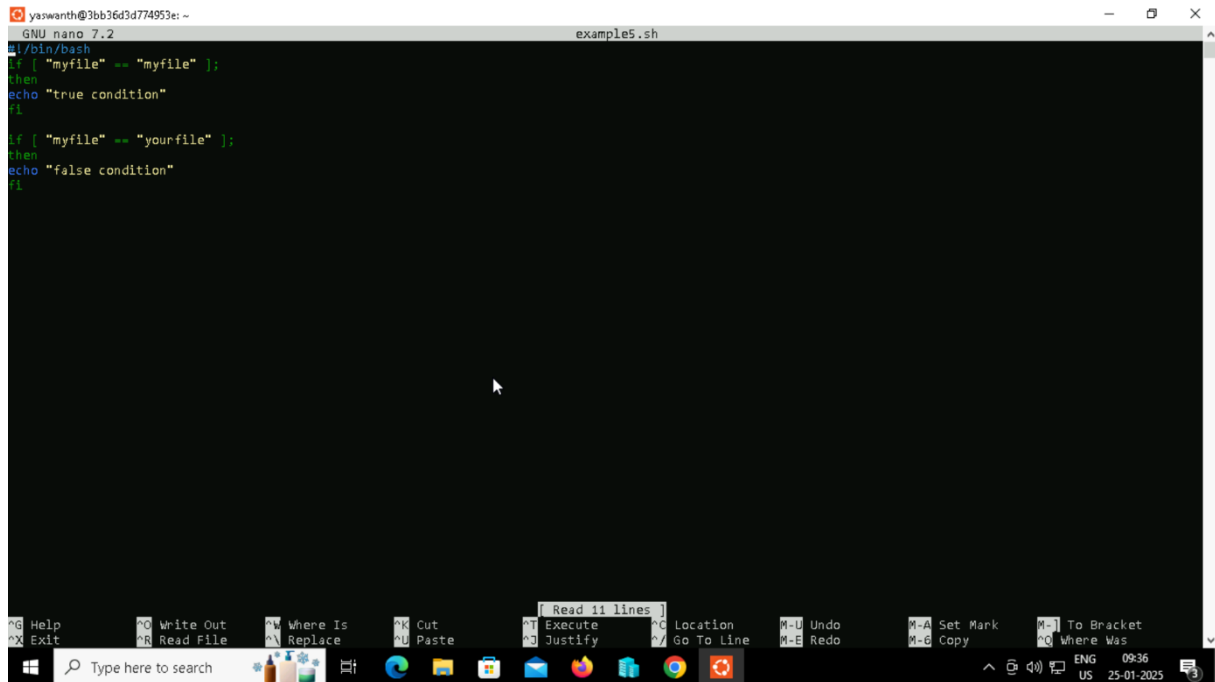


The screenshot shows the same terminal window after the script has been executed. The user has entered '279' in response to the prompt 'Enter number :'. The output of the script is 'Value is greater than 125'.

```
yeswanth@3bb36d3d774953e:~$ nano examp14.sh
yeswanth@3bb36d3d774953e:~$ chmod a+x example4.sh
yeswanth@3bb36d3d774953e:~$ ./example4.sh
Enter number : 279
Value is greater than 125
yeswanth@3bb36d3d774953e:~$
```

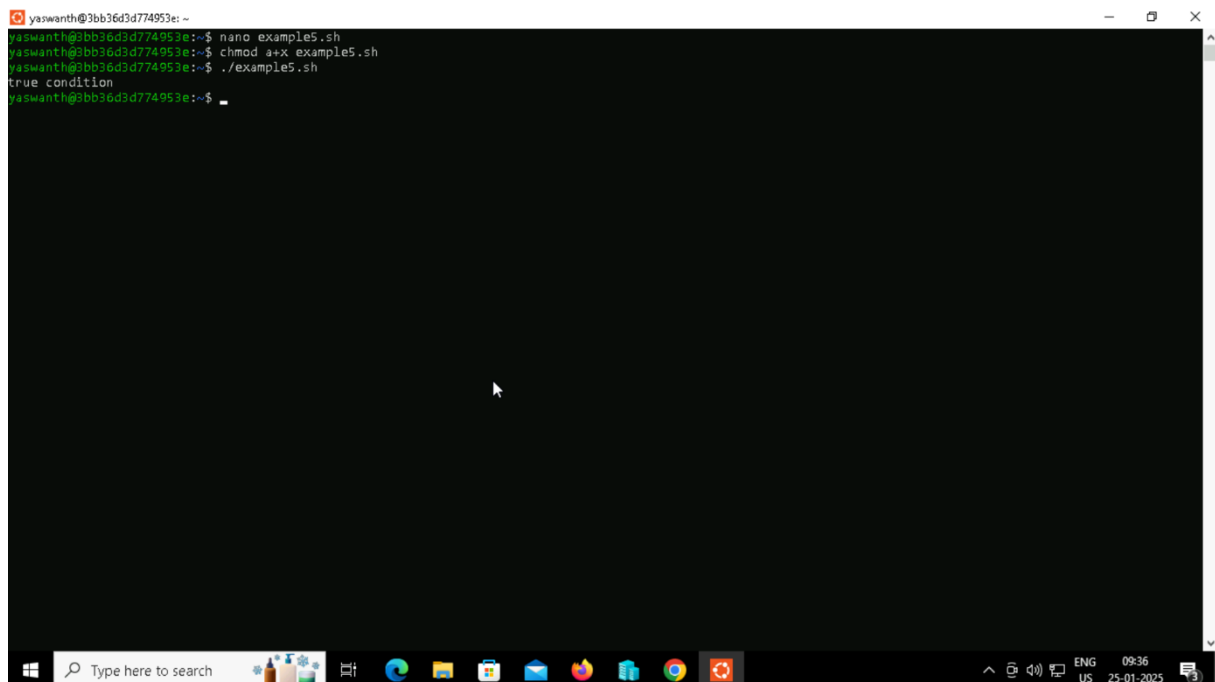
The terminal window shows the same desktop environment as the previous screenshot.

5. In this example, we demonstrate the usage of **if statement** with a simple scenario of comparing two strings:



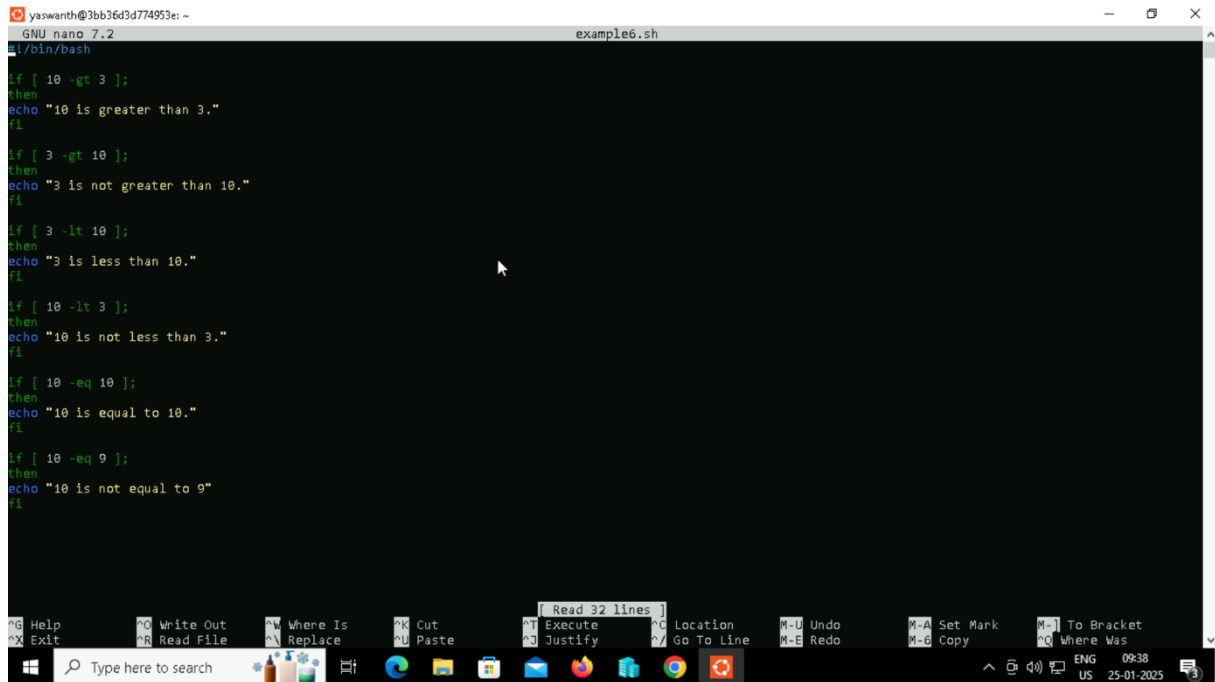
```
GNU nano 7.2 example5.sh
#!/bin/bash
if [ "myfile" == "myfile" ];
then
echo "true condition"
fi

if [ "myfile" == "yourfile" ];
then
echo "false condition"
fi
```



```
yaswanth@3bb36d3d774953e: ~
yaswanth@3bb36d3d774953e:~$ nano example5.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x example5.sh
yaswanth@3bb36d3d774953e:~$ ./example5.sh
true condition
yaswanth@3bb36d3d774953e:~$
```

6. In this example, we demonstrate how to compare numbers by using the if statement



```
GNU nano 7.2 example6.sh
#!/bin/bash

if [ 10 -gt 3 ];
then
echo "10 is greater than 3."
fi

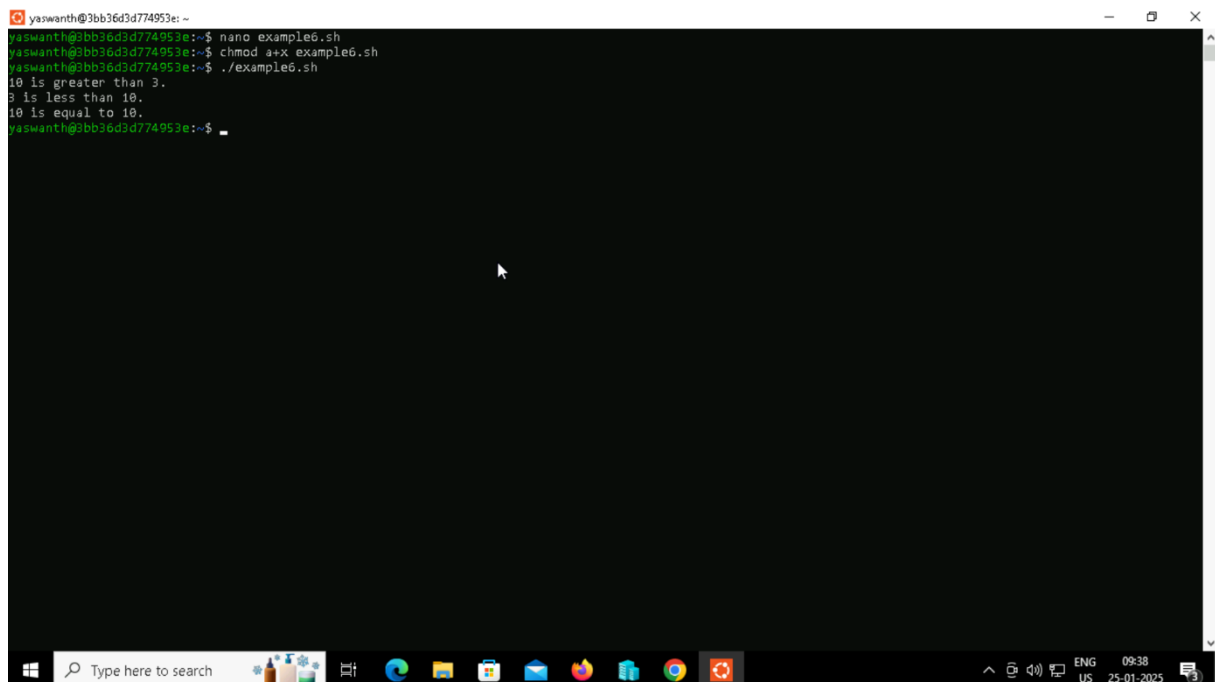
if [ 3 -gt 10 ];
then
echo "3 is not greater than 10."
fi

if [ 3 -lt 10 ];
then
echo "3 is less than 10."
fi

if [ 10 -lt 3 ];
then
echo "10 is not less than 3."
fi

if [ 10 -eq 10 ];
then
echo "10 is equal to 10."
fi

if [ 10 -eq 9 ];
then
echo "10 is not equal to 9"
fi
```



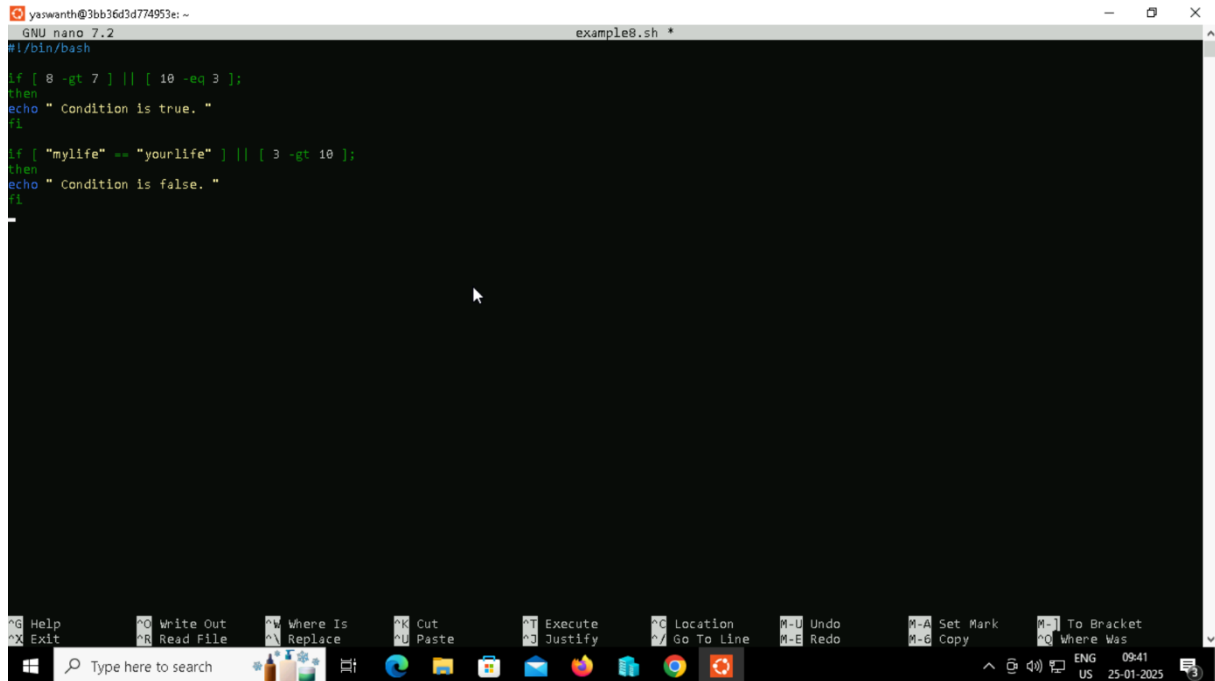
```
yaswanth@3bb36d3d774953e: ~
yaswanth@3bb36d3d774953e:~$ nano example6.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x example6.sh
yaswanth@3bb36d3d774953e:~$ ./example6.sh
10 is greater than 3.
3 is less than 10.
10 is equal to 10.
yaswanth@3bb36d3d774953e:~$
```

7. In this example, we will define how to use AND operator to include multiple conditions in the if expression

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano example7.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x example7.sh  
yaswanth@3bb36d3d774953e:~$ ./example7.sh  
Conditions are true  
yaswanth@3bb36d3d774953e:~$
```

```
yaswanth@3bb36d3d774953e: ~  
GNU nano 7.2 example7.sh  
#!/bin/bash  
  
if [ 8 -gt 6 ] && [ 10 -eq 10 ];  
then  
echo "Conditions are true"  
fi  
  
if [ "mylife" == "mylife" ] && [ 3 -gt 10 ];  
then  
echo "Conditions are false"  
fi
```

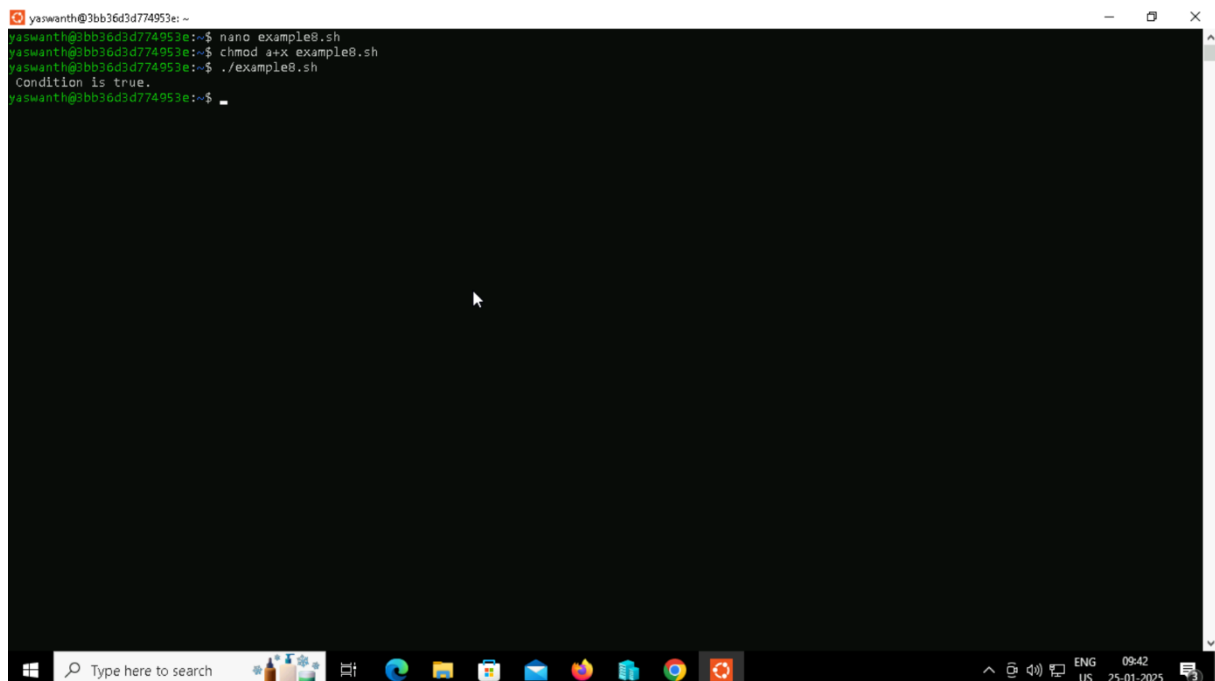
8. In this example, we will define how to use OR operator to include multiple conditions in the if expression



```
GNU nano 7.2 example8.sh
#!/bin/bash

if [ 8 -gt 7 ] || [ 10 -eq 3 ];
then
echo " Condition is true. "
fi

if [ "mylife" == "yourlife" ] || [ 3 -gt 10 ];
then
echo " Condition is false. "
fi
```



```
yaswanth@3bb36d3d774953e: ~
yaswanth@3bb36d3d774953e:~$ nano example8.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x example8.sh
yaswanth@3bb36d3d774953e:~$ ./example8.sh
Condition is true.
yaswanth@3bb36d3d774953e:~$
```



9. In this example, we will define how to use AND and OR to include multiple conditions in the if expression:

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ chmod a+x example9.sh  
chmod: cannot access 'example9.sh': No such file or directory  
yaswanth@3bb36d3d774953e:~$ nano example9.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x example9.sh  
yaswanth@3bb36d3d774953e:~$ ./example9.sh  
Condition is true.  
yaswanth@3bb36d3d774953e:~$
```

```
GNU nano 7.2 example9.sh  
#!/bin/bash  
  
if [[ 10 -eq 10 && 5 -gt 4 || 3 -eq 4 || 3 -lt 6 ]];  
then  
echo "Condition is true."  
fi  
  
if [[ 8 -eq 8 && 8 -gt 10 || 9 -lt 5 ]];  
then  
echo "Condition is false"  
fi
```

10. In this example, we will find "if a given number is greater than 50 and if it is an even number" by using nested if expression

```
yaswanth@3bb36d3d774953e: ~$ nano example11.sh
yaswanth@3bb36d3d774953e:~$ chmod a+x example11.sh
yaswanth@3bb36d3d774953e:~$ ./example11.sh
./example11.sh: line 3: [: -gt: unary operator expected
yaswanth@3bb36d3d774953e:~$ nano example11.sh
yaswanth@3bb36d3d774953e:~$ ./example11.sh
./example11.sh: line 3: [: -gt: unary operator expected
yaswanth@3bb36d3d774953e:~$ ./example11.sh 100
./example11.sh: line 3: [100: command not found
yaswanth@3bb36d3d774953e:~$ nano example11.sh
yaswanth@3bb36d3d774953e:~$ ./example11.sh 100
Number is greater than 50.
and it is an even number.
yaswanth@3bb36d3d774953e:~$
```

```
yaswanth@3bb36d3d774953e: ~$ nano example11.sh
GNU nano 7.2 example11.sh
#!/bin/bash

if [ -z "$1" ]
then
    echo "Please provide a number as an argument."
    exit 1
fi

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
```

11. Following example consists of two different scenarios where in the first if-else statement, the condition is true, and in the second if-else statement, the condition is false

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano ifelse1.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x ifelse1.sh  
yaswanth@3bb36d3d774953e:~$ ./ifelse1.sh  
-bash: ./ifelse1.sh: No such file or directory  
yaswanth@3bb36d3d774953e:~$ ./ifelse1.sh  
Given condition is true.  
Given condition is not true.  
yaswanth@3bb36d3d774953e:~$
```

```
GNU nano 7.2 ifelse1.sh  
#!/bin/bash  
  
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  
  
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
```

12. In this example, how to use multiple conditions with the if-else statement in Bash. I used bash logical operators to join multiple conditions

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano ifelse2.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x ifelse2.sh  
yaswanth@3bb36d3d774953e:~$ ./ifelse2.sh  
Enter a value:89  
The value you typed is greater than 9.  
yaswanth@3bb36d3d774953e:~$ ./ifelse2.sh  
Enter a value:5  
The value you typed is not greater than 9.  
yaswanth@3bb36d3d774953e:~$
```

```
GNU nano 7.2 ifelse2.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
```

13. Following is an example explaining how to make use of the nested ifelse statement in Bash:

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano ifelse3.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x ifelse3.sh  
yaswanth@3bb36d3d774953e:~$ ./ifelse3.sh  
Enter a number of quantity:100  
Lucky Draw Winner  
Eligible to get the item for free  
yaswanth@3bb36d3d774953e:~$ ./ifelse3.sh  
Enter a number of quantity:80  
Eligible for 5% discount  
yaswanth@3bb36d3d774953e:~$ ./ifelse3.sh  
Enter a number of quantity:10  
Eligible for 5% discount  
yaswanth@3bb36d3d774953e:~$ ./ifelse3.sh  
Enter a number of quantity:100  
Eligible for 10% discount  
yaswanth@3bb36d3d774953e:~$
```

```
GNU nano 7.2 ifelse3.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
```

14. Following example consists of two different scenarios wherein the first else-if statement, the condition is true, and in the second else-if statement, the condition is false

```
yaswanth@3bb36d3d774953e: ~  
yaswanth@3bb36d3d774953e:~$ nano ifelse4.sh  
yaswanth@3bb36d3d774953e:~$ chmod a+x ifelse4.sh  
yaswanth@3bb36d3d774953e:~$ ./ifelse4.sh  
Enter a number of quantity:180  
Eligible for 10% discount  
yaswanth@3bb36d3d774953e:~$ ./ifelse4.sh  
Enter a number of quantity:100  
Lucky Draw Winner  
Eligible to get the item for free  
yaswanth@3bb36d3d774953e:~$ ./ifelse4.sh  
Enter a number of quantity:90  
No discount  
yaswanth@3bb36d3d774953e:~$
```

```
yaswanth@3bb36d3d774953e: ~  
GNU nano 7.2 ifelse4.sh  
#!/bin/bash  
  
read -p "Enter a number of quantity:" num  
  
if [ $num -gt 200 ];  
then  
    echo "Eligible for 20% discount"  
elif [[ $num == 200 || $num == 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
```