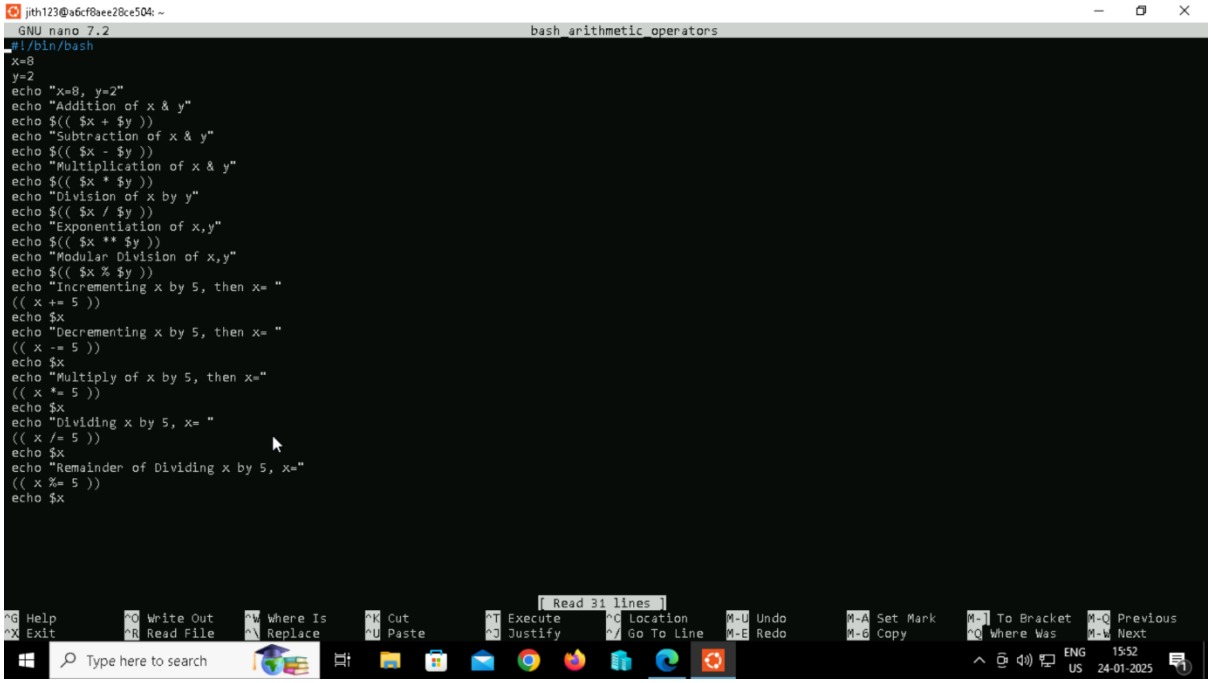
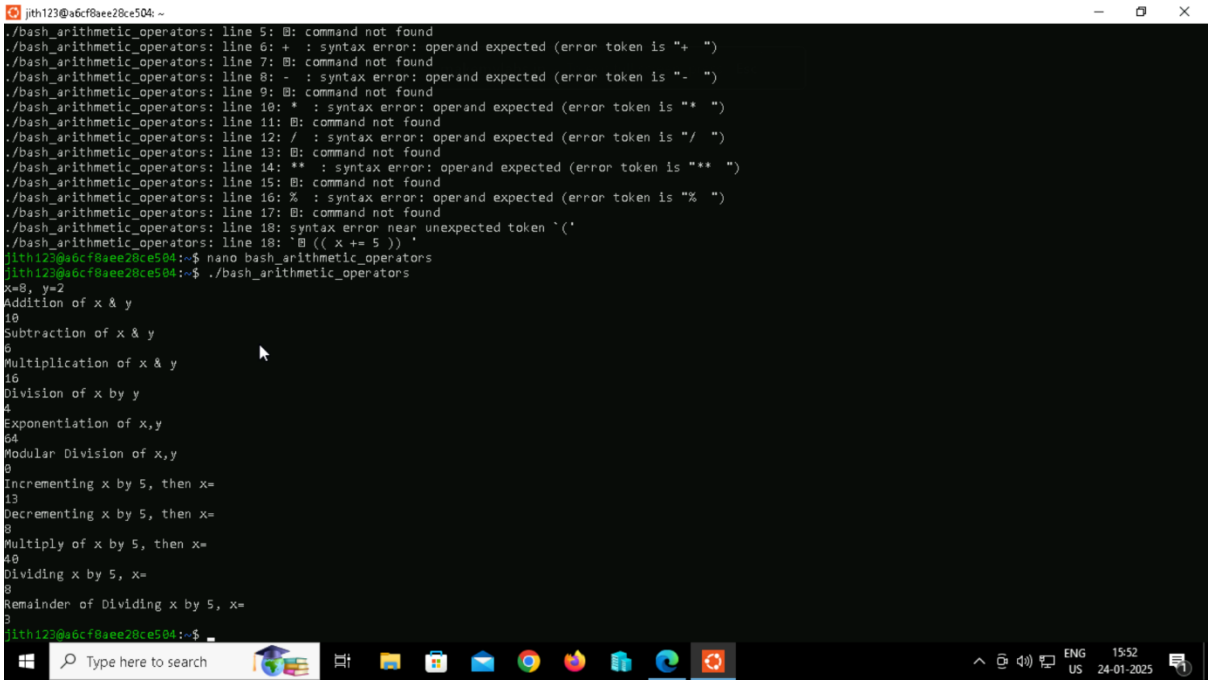
Bash Arithmetic Operators Code:

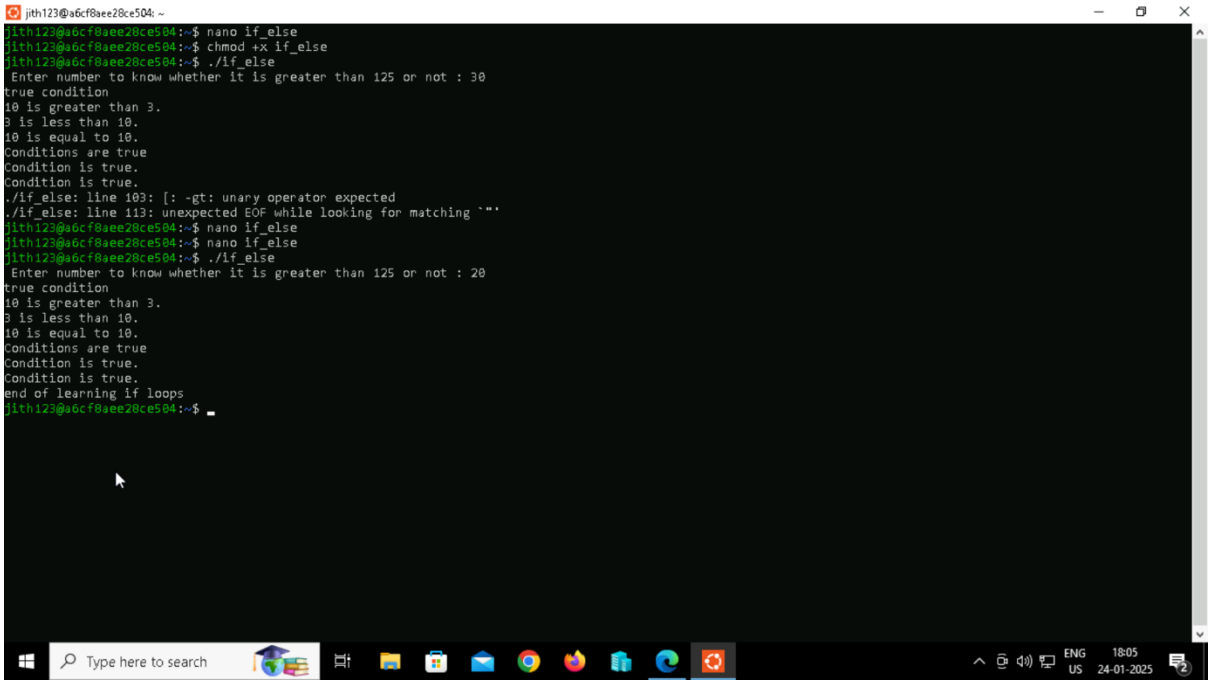


Output:



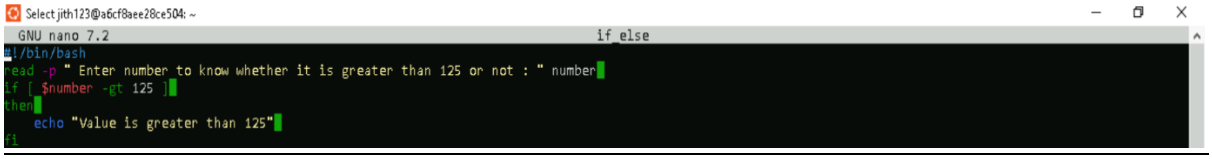
If-else statements with eq ,gt,lt,&&,||, operators :

Output of if-else programs in same file:

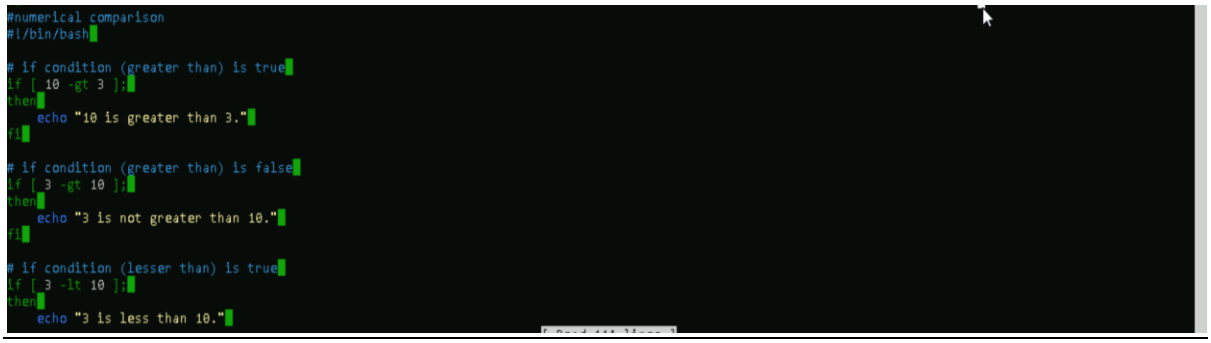


Actual code:

Value greater than 125:



Numerical comparison:



```
#numerical comparison
#!/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."
```

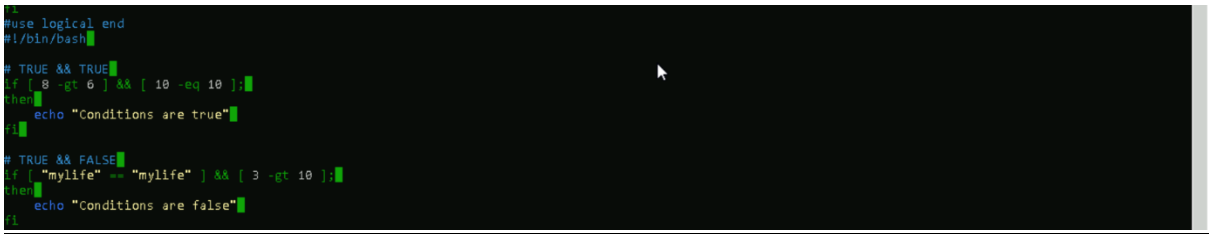
String comparison:

```
#string comparison
#!/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
```

Logical OR:

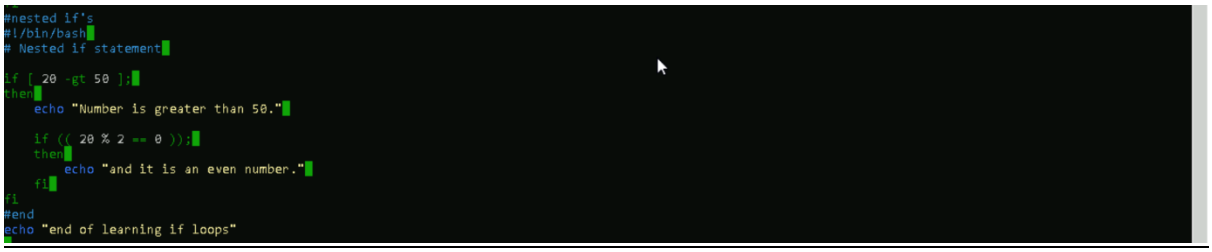
```
#logical or
#!/bin/bash

# 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
```

Logical And



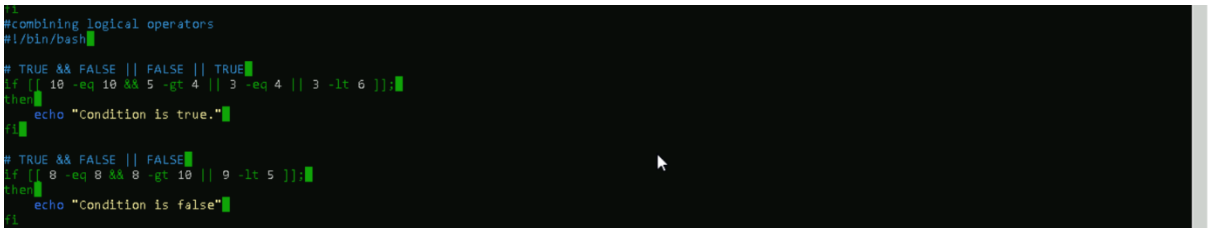
```
#use logical end
#!/bin/bash

# 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
```

Nested if:



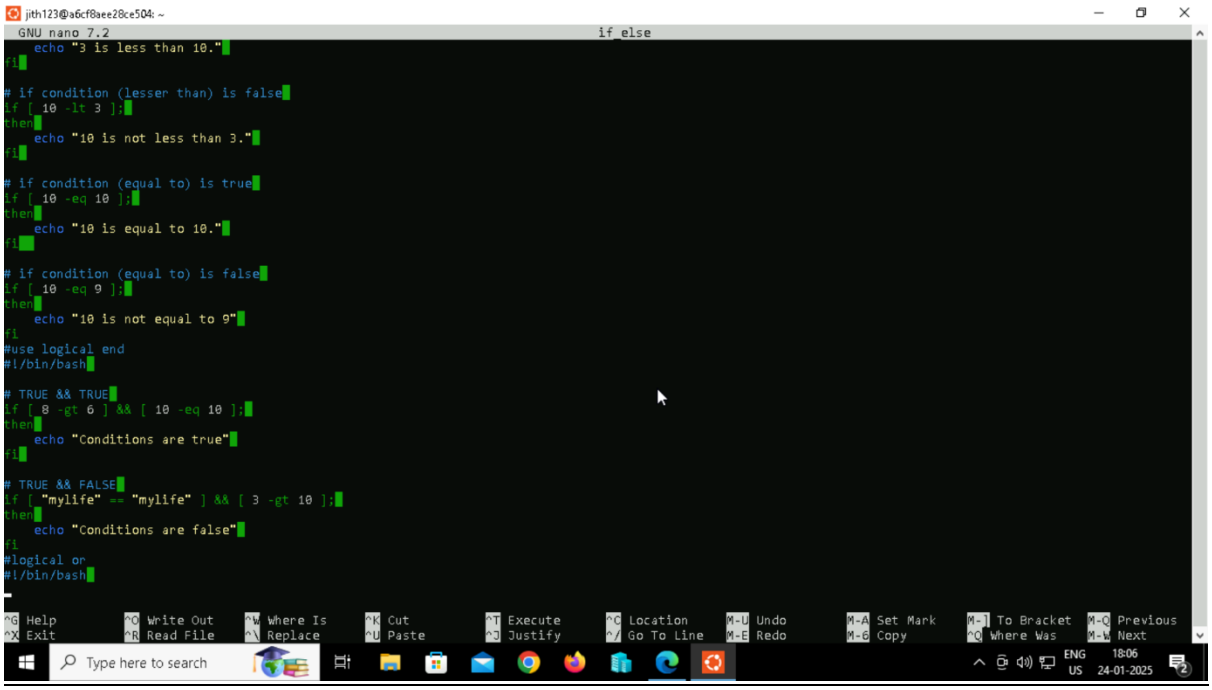
```
#nested if's
#!/bin/bash
# Nested if statement

if [ 20 -gt 50 ];
then
    echo "Number is greater than 50."

    if (( 20 % 2 == 0 ));
    then
        echo "and it is an even number."
    fi
fi
#end
echo "end of learning if loops"
```

Combining Logical operators:

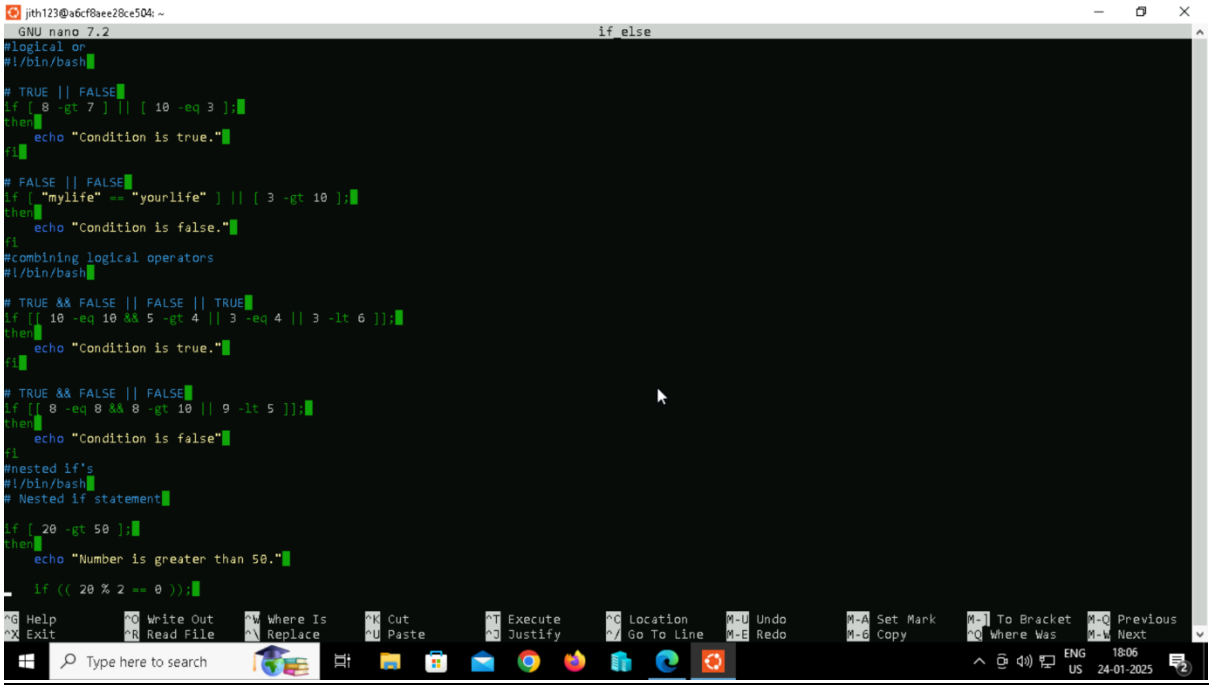


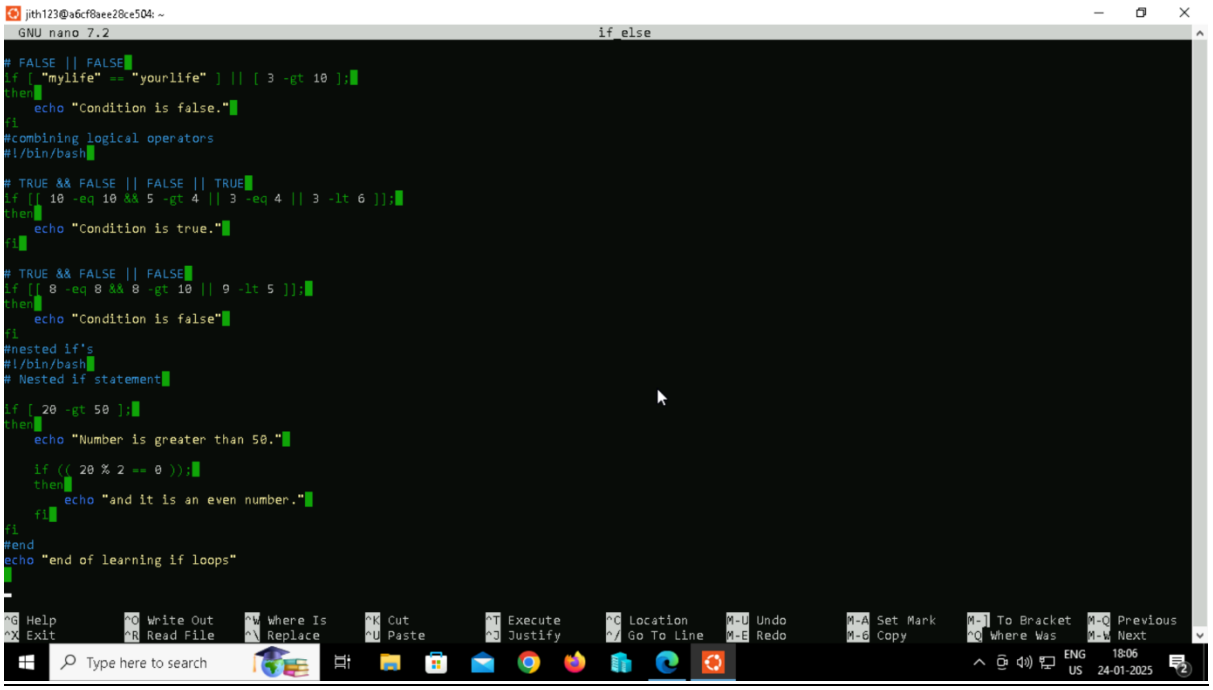
```
#combining logical operators
#!/bin/bash

# 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 8 && 8 -gt 10 || 9 -lt 5 ]];
then
    echo "Condition is false"
fi
```

All codes in one file:



```
jith123@a6cf8aee28ce504: ~
  GNU nano 7.2                                  if_else
    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
#use logical end
#!/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
#logical or
#!/bin/bash

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


jith123@a6cf8aee28ce504: ~
GNU nano 7.2
if_else
#logical or
#!/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
#combining logical operators
#!/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 8 && 8 -gt 10 || 9 -lt 5]];
then
echo "Condition is false"
fi
#nested if's
#!/bin/bash
Nested if statement
if [20 -gt 50];
then
echo "Number is greater than 50."
if ((20 % 2 == 0));
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next
Type here to search
ENG US
18:06
24-01-2025


jith123@a6cf8aee28ce504: ~
GNU nano 7.2
if_else
# FALSE || FALSE
if ["mylife" == "yourlife"] || [3 -gt 10];
then
echo "Condition is false."
fi
#combining logical operators
#!/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 8 && 8 -gt 10 || 9 -lt 5]];
then
echo "Condition is false"
fi
#nested if's
#!/bin/bash
Nested if statement
if [20 -gt 50];
then
echo "Number is greater than 50."
if ((20 % 2 == 0));
then
echo "and it is an even number."
fi
fi
#end
echo "end of learning if loops"
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next
Type here to search
ENG US
18:06
24-01-2025

Bash if -else Example-1:

```
 GNU nano 7.2                                                bash_if-else_ex-1
#!/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
```

Bash if -else Example-2:

```
 GNU nano 7.2                                                bash_if-else_ex-2
#!/bin/bash
# When condition is true
# TRUE && FALSE || FALSE || 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
#TRUE && FALSE || FALSE || TRUE
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
```

Bash if -else statement in single line E-3:

```
jith123@a6cf8aee28ce504: ~
 GNU nano 7.2                                                bash_if-else_ex-3
#!/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
```

Bash Nested if -else Example-4:

```
 GNU nano 7.2                                                bash_if-else_ex-4
#!/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
```

Bash else-if Example-5:

```
 GNU nano 7.2                                          bash_if-else_ex-5
#!/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
```

Bash else-if Example-6:

```
 GNU nano 7.2                                          bash_if-else_ex-6
#!/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
```

Output of all Bash if -else programs from Ex-1 to 6:

```
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-1
10 is greater than 3.
3 is not greater than 10.
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-2
Given condition is true.
Given condition is not true.
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-3
Enter a value: 47
The value you typed is greater than 9.
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-4
Enter a value:30
The value you typed is greater than 9.
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-5
Enter a number of quantity:10
Eligible for 5% discount
jith123@a6cf8aee28ce504:~$ ./bash_if-else_ex-6
Enter a number of quantity:25
No discount
```