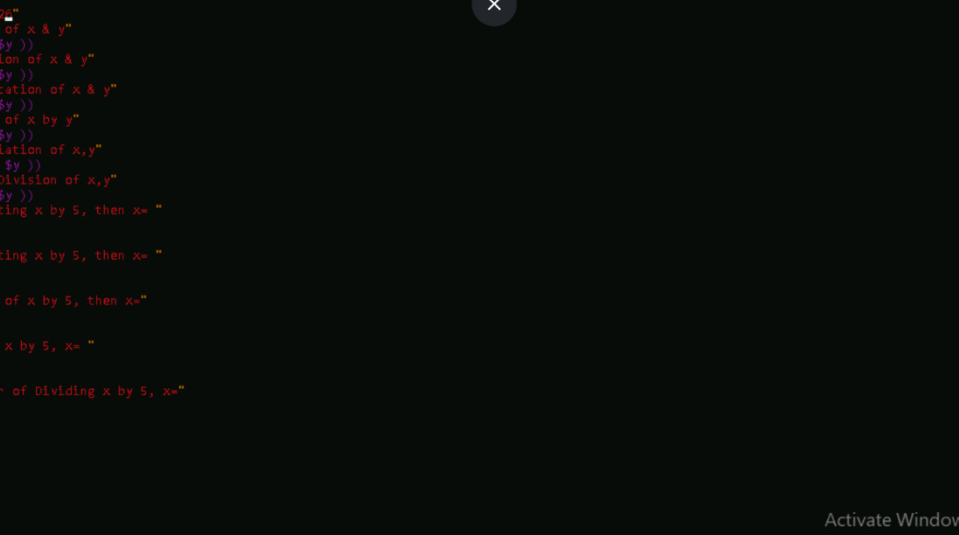


BASH ASSIGNMENT PROGRAMS

1. ARITHMETIC OPERATIONS

CODE

```
srinikar@520014aa64d7503: ~  
❯ fl /bin/bash  
x=0  
y=2  
echo "x=11, y=26"  
echo "Addition of x & y"  
echo $(( $x + $y ))  
echo "Subtraction of x & y"  
echo $(( $x - $y ))  
echo "Multiplication of x & y"  
echo $(( $x * $y ))  
echo "Division of x by y"  
echo $(( $x / $y ))  
echo "Exponentiation of x,y"  
echo $(( $x ** $y ))  
echo "Modular Division of x,y"  
echo $(( $x % $y ))  
echo "Incrementing x by 5, then x= "  
(( x += 5 ))  
echo $x  
echo "Decrementing x by 5, then x= "  
(( x -= 5 ))  
echo $x  
echo "Multiply of x by 5, then x= "  
(( x *= 5 ))  
echo $x  
echo "Dividing x by 5, x= "  
(( x /= 5 ))  
echo $x  
echo "Remainder of Dividing x by 5, x= "  
(( x %= 5 ))  
echo $x
```

Activate Windows
Go to Settings to activate Windows.

OUTPUT

```

snikar@520014aa64d7583: ~
snikar@520014aa64d7583:~$ vi arithmetic.sh
snikar@520014aa64d7583:~$ chmod +x arithmetic.sh
snikar@520014aa64d7583:~$ ./arithmetic.sh
x=11, y=26
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=
49
Dividing x by 5, x=
8
Remainder of Dividing x by 5, x=
3
snikar@520014aa64d7583:~$

```

Activate Windows
Go to Settings to activate Windows.

2. BACK TICKS

CODE

```
srikan@528014aa64d7583: ~  
$ vi /bin/backticks  
echo "a=10, b=3"  
echo "c is the value of addition c=a+b"  
a=10  
b=3  
echo "c= `expr $a + $b`"  
  
"backticks.sh" 7L, 105B
```

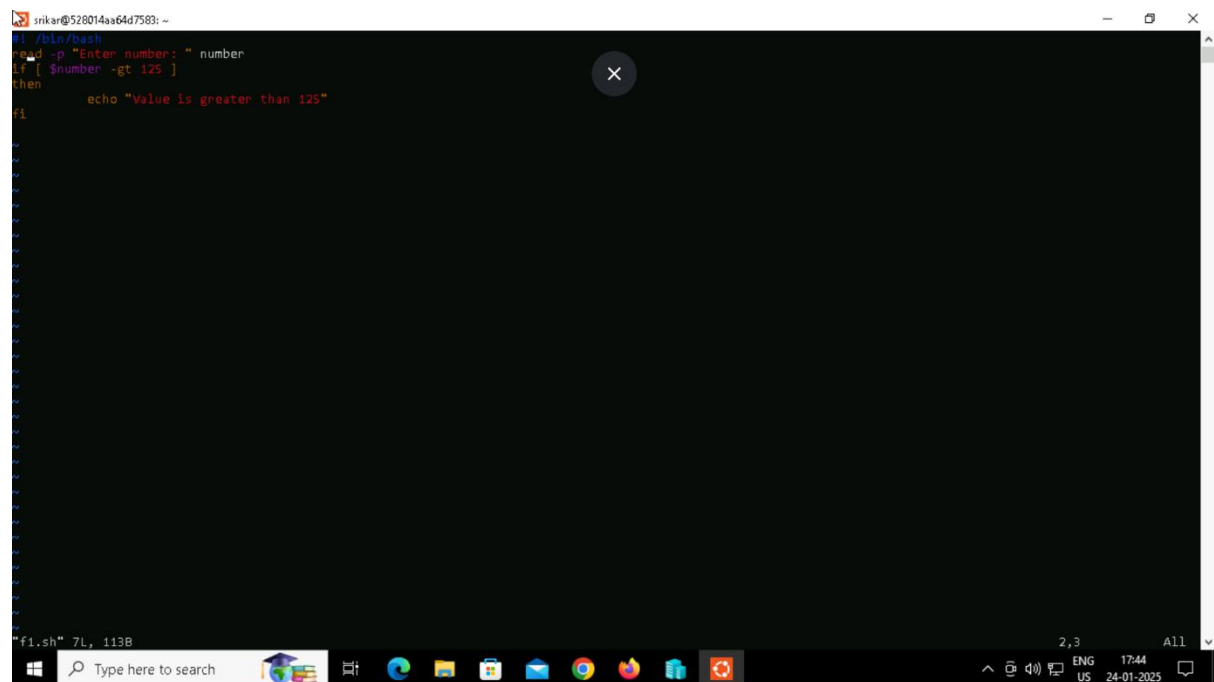
OUTPUT

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ vi backticks.sh  
srikan@528014aa64d7583:~$ chmod +x backticks.sh  
srikan@528014aa64d7583:~$ ./backticks.sh  
a=10, b=3  
c is the value of addition c=a+b  
c= 13  
srikan@528014aa64d7583:~$
```

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

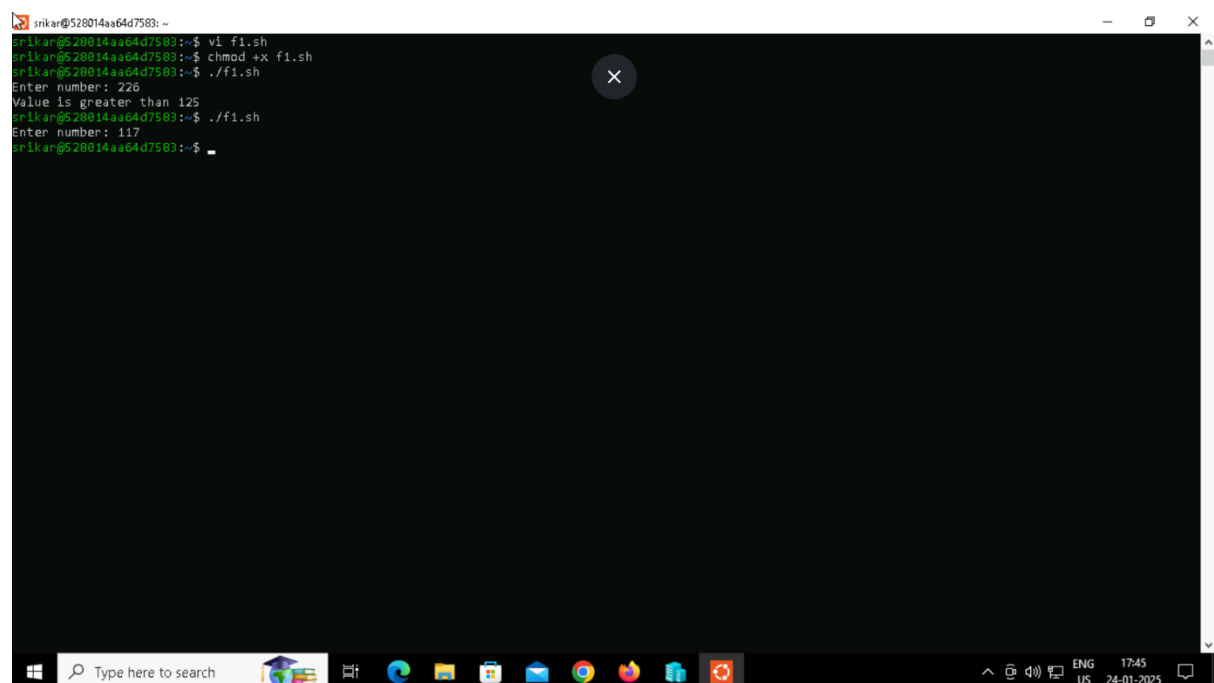
CODE

```
srikan@528014aa64d7583: ~  
$ vi f1.sh  
#!/bin/bash  
read -p "Enter number: " number  
if [ $number -gt 125 ]  
then  
    echo "Value is greater than 125"  
fi  
$
```



OUTPUT

```
srikan@528014aa64d7583: ~  
$ vi f1.sh  
srikan@528014aa64d7583:~$ v1 f1.sh  
srikan@528014aa64d7583:~$ chmod +x f1.sh  
srikan@528014aa64d7583:~$ ./f1.sh  
Enter number: 226  
Value is greater than 125  
srikan@528014aa64d7583:~$ ./f1.sh  
Enter number: 117  
srikan@528014aa64d7583:~$
```



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

CODE

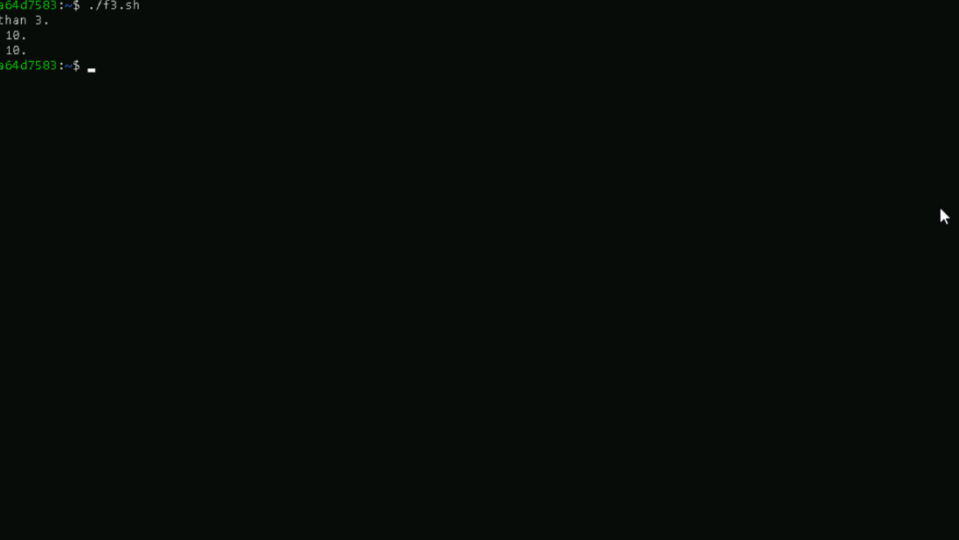
```
srikan@528014aa64d7583: ~  
$ vi f2.sh  
#!/bin/bash  
if [ "myfile" == "myfile" ];  
then  
    echo "true condition"  
fi  
  
if [ "myfile" == "yourfile" ];  
then  
    echo "false condition"  
fi
```

OUTPUT

```
srikan@528014aa64d7583: ~  
$ vi f2.sh  
srikan@528014aa64d7583:~$ v1 f2.sh  
srikan@528014aa64d7583:~$ chmod +x f2.sh  
srikan@528014aa64d7583:~$ ./f2.sh  
true condition  
srikan@528014aa64d7583:~$
```

CODE

OUTPUT



The screenshot shows a Windows terminal window with a black background and white text. The title bar at the top reads "snikar@528014aa64d7583: ~". The terminal content shows a user executing a series of commands to run a script named "f3.sh". The script contains three conditional statements that compare the value of "d" (which is 3) to the value of "i0" (which is 10). The output of the script is displayed on the lines following the execution command. The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several application icons including Edge, File Explorer, and the Microsoft Store. The system tray on the right indicates the language is ENG, the location is UK, and the time is 17:53 on 24-01-2025.

```
snikar@528014aa64d7583: ~  
snikar@528014aa64d7583:~$ vi f3.sh  
snikar@528014aa64d7583:~$ chmod +x f3.sh  
snikar@528014aa64d7583:~$ ./f3.sh  
10 is greater than 3.  
3 is less than 10.  
10 is equal to 10.  
snikar@528014aa64d7583:~$
```

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

CODE

```
srikan@528014aa64d7583: ~  
$ vi f4.sh  
if [ 8 -gt 0 ] && [ 10 -eq 10 ];  
then  
    echo "Conditions are true"  
fi  
  
TRUE && FALSE  
if [ "mylife" == "mylife" ] && [ 3 -gt 10 ];  
then  
    echo "Conditions are false"  
fi
```

OUTPUT

```
srikan@528014aa64d7583: ~  
$ vi f4.sh  
srikan@528014aa64d7583:~$ v1 f4.sh  
srikan@528014aa64d7583:~$ chmod +x f4.sh  
srikan@528014aa64d7583:~$ ./f4.sh  
Conditions are true  
srikan@528014aa64d7583:~$
```

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

CODE

```
srikan@528014aa64d7583: ~  
# /bin/bash  
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
```

OUTPUT

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ v1 f5.sh  
srikan@528014aa64d7583:~$ chmod +x f5.sh  
srikan@528014aa64d7583:~$ ./f5.sh  
Condition is true.  
srikan@528014aa64d7583:~$
```

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

CODE

```
srikan@528014aa64d7583: ~  
# /bin/bash  
if [[ 18 -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
```

OUTPUT

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ v1 f6.sh  
srikan@528014aa64d7583:~$ chmod +x f6.sh  
srikan@528014aa64d7583:~$ ./f6.sh  
Condition is true.  
srikan@528014aa64d7583:~$
```


9. In this example, we explained how to use multiple conditions with the if-else statement in Bash.

CODE

```
snikar@528014aa64d7583: ~  
#/bin/bash  
# 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  
"f7.sh" 16L, 401B
```

OUTPUT

```
snikar@528014aa64d7583: ~  
snikar@528014aa64d7583:~$ v1 f7.sh  
snikar@528014aa64d7583:~$ chmod +x f7.sh  
snikar@528014aa64d7583:~$ ./f7.sh  
Given condition is not true.  
snikar@528014aa64d7583:~$
```

10. If-else statement in single line

CODE

```
snikan@528014aa64d7583: ~  
#!/bin/bash  
-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
```

OUTPUT

```
snikan@528014aa64d7583: ~  
snikan@528014aa64d7583:~$ vi f8.sh  
snikan@528014aa64d7583:~$ chmod +x f8.sh  
snikan@528014aa64d7583:~$ ./f8.sh  
Enter a value:10  
The value you typed is greater than 9.  
snikan@528014aa64d7583:~$ ./f8.sh  
Enter a value:6  
The value you typed is not greater than 9.  
snikan@528014aa64d7583:~$
```

11. Nested If-else

CODE

```
srikan@528014aa64d7583: ~  
#!/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
```

OUTPUT

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ vi f10.sh  
srikan@528014aa64d7583:~$ chmod +x f10.sh  
srikan@528014aa64d7583:~$ ./f10.sh  
Enter a value:20  
The value you typed is greater than 9.  
srikan@528014aa64d7583:~$ ./f10.sh  
Enter a value:6  
The value you typed is not greater than 9.  
srikan@528014aa64d7583:~$
```

BASH ELSE IF (ELIF STATEMENTS)

Example 1

CODE

```
srikan@528014aa64d7583: ~  
#!/bin/bash  
  
read -p "Enter a number of quantity:" num  
  
if [ $num -gt 100 ];  
then  
echo "Eligible for 18% 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
```

OUPUT

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ vi f11.sh  
srikan@528014aa64d7583:~$ chmod +x f11.sh  
srikan@528014aa64d7583:~$ ./f11.sh  
Enter a number of quantity:150  
Eligible for 18% discount  
srikan@528014aa64d7583:~$ ./f11.sh  
Enter a number of quantity:50  
Eligible for 5% discount  
srikan@528014aa64d7583:~$ ./f11.sh  
Enter a number of quantity:100  
Lucky Draw Winner  
Eligible to get the item for free  
srikan@528014aa64d7583:~$
```

Example 2

CODE

```
srikan@528014aa64d7583: ~  
#!/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

```
srikan@528014aa64d7583: ~  
srikan@528014aa64d7583:~$ vi f12.sh  
srikan@528014aa64d7583:~$ chmod +x f12.sh  
srikan@528014aa64d7583:~$ ./f12.sh  
Enter a number of quantity:250  
Eligible for 20% discount  
srikan@528014aa64d7583:~$ ./f12.sh  
Enter a number of quantity:200  
Lucky Draw Winner  
Eligible to get the item for free  
srikan@528014aa64d7583:~$ ./f12.sh  
Enter a number of quantity:150  
Eligible for 10% discount  
srikan@528014aa64d7583:~$ ./f12.sh  
Enter a number of quantity:100  
Lucky Draw Winner  
Eligible to get the item for free  
srikan@528014aa64d7583:~$ ./f12.sh  
Enter a number of quantity:50  
No discount  
srikan@528014aa64d7583:~$
```