OS LAB

```
#!/bin/bash
echo Opi
read x
read y
#Arithmetic Operators
echo Addition: `expr $x + $y`
echo Subtraction: `expr $x - $y`
echo Multiplication: `expr $x \* $y`
echo Division: `expr $x / $y`
echo Modulus: `expr $x % $y`
#Relational Operators
#equal -eq/==
if [ $x -eq $y ]; then
    echo Values are equal
else
    echo Values are not equal
fi
#not equal -ne/!=
if [ $x -ne $y ]; then
    echo Values are not equal
else
    echo Values are equal
fi
#greater -gt/>
if [ $x -gt $y ]; then
    echo $x is greater
```

```
else
    echo $y is greater
fi

#less -lt/<
if [ $x -lt $y ]; then
    echo $x is not greater than $y
else
    echo $x is greater than $y
fi
# -ge
if [ $x -ge $y ]; then
    echo $x is greater or equal to $y
else
    echo $x is not greater or equal to $y
fi
# -le
if [ $x -le $y ]; then
    echo $x is less or equal to $y
else
    echo $x is not less or equal to $y
fi
#String Operators
read str1
if [ -z "str1" ]; then
    echo Empty
else
```

```
        echo Not empty
fi
#Floating point
num1=20.6
num2=10
echo "$num1+$num2" | bc
#Even-Odd(1 number)
read -p "Enter a number: " n
check=`expr $n % 2`
if [ $check -eq 0 ]; then
        echo $n is even
else
        echo $n is odd
fi
#even odd =2 numbers
read -p "Enter 1st num: " n1
read -p "Enter 2nd num: " n2
check1=`expr $n1 % 2`
check2=`expr $n2 % 2`
if [ $check1 == 0 ] && [ $check2 == 0 ]; then
        echo Both even
elif [ $check1 == 0 ] && [ $check2 != 0 ]; then
        echo $n1 even and $n2 odd
elif [ $check1 != 0 ] && [ $check2 == 0 ]; then
        echo $n1 odd and $n2 even
else
        echo Both odd      fi
```

#switch-case

#Print days

```
read -p "Enter a number between 1 to 7: " day
case $day in
    1) echo "Saturday" ;;
    2) echo "Sunday" ;;
    3) echo "Monday" ;;
    4) echo "Tuesday" ;;
    5) echo "Wednesday" ;;
    6) echo "Thursday" ;;
    7) echo "Friday" ;;
    *) echo "Invalid" ;;
esac
```

#calculator

```
read -p "Enter 1st num: " cal1
read -p "Enter 2nd num: " cal2
read -p "Enter a operator to calculate( + or - or m or / or %): " op
case $op in
    +) echo Addition: `expr $cal1 + $cal2` ;;
    -) echo Subtraction: `expr $cal1 - $cal2` ;;
    m) echo Multiplication: `expr $cal1 \* $cal2` ;;
    /) echo Division: `expr $cal1 / $cal2` ;;
    %) echo Modulus: `expr $cal1 % $cal2` ;;
    *) echo Invalid ;;
Esac
```

```
#for loop
#print numbers 1 to 5
for number in 1 2 3 4 5; do
    echo $number
done
#print 1-10
for number in {1..10}; do
    echo $number
done
#sum of 1-10
sum=0
for((i=1; i<=10; i++)); do
    sum=$((sum+i))
done
echo $sum
#while loop
#print 1-10
n=1
while [ $n -le 10 ]; do
    echo $n
    n=$((n+1))
done
#sum of 1-10
sum=0
n=1
while [ $n -le 10 ]; do
    sum=`expr $sum + $n`
```

```
        n=`expr $n + 1`
done
echo Sum: $sum
```

#Case With Loop

```
select name in alal dulal rahim
do
        case $name in
                alal) echo ALAL IS SELECTED ;;
                dulal) echo DULAL IS SELECTED ;;
                rahim) echo RAHIM IS SELECTED ;;
                *) echo Default ;;
        esac
done
```

#until loop

```
a=0
until [ $a -lt 10 ]; do
        echo $a
        a=`expr $a + 1`
done
```

#Break

```
for (( i=1; i<10; i++ )); do
 if [ $i –gt 5 ]; then
 break
 fi
 echo “$i”
 done
```

```
#Continue
 for (( i=1; i<10; i++ )); do
 if [ $i –eq 5 ]; then
 continue
 fi
 echo “$i”
 done
#array
os=('ubuntu' 'windows' 'linux')
echo "${os[0]}"
echo "${os[1]}"
echo "${os[2]}"
#legnth of array
echo "${#os[@]}"
#Removing value from array
 unset os[2]
 # Printing array after removal
echo "After removal: ${os[@]}"
echo "Length after removal: ${#os[@]}"
```

#create and put value and print value

```
echo "Enter elements separated by spaces:"
read -a myArray  # Taking input into the array

# Accessing individual elements
echo "First element: ${myArray[0]}"
echo "Second element: ${myArray[1]}"
echo "Third element: ${myArray[2]}"

# Printing the entire array
echo "All elements: ${myArray[@]}"

# Printing the length of the array
echo "Number of elements: ${#myArray[@]}"
```

#arrayinput

```
echo "Enter 2 Integer Seperated by Sapace: "
read -a numbers
sum=0
for i in "${numbers[@]}"
do
	sum=$((sum + i))
done
echo "The Sum of The Array Elements is : $sum"
```

output:

```
Enter 2 Integer Seperated by Sapace:
2 3
The Sum of The Array Elements is : 5
```

#file parsing

```
echo Name Name-ID > data.txt
echo MD. SHOHANUR RAHMAN SHOHAN-46013 >> data.txt
echo Rafiul Hassan -47048 >> data.txt
echo Abu Towsif-47019 >> data.txt
echo Farjana Opi-47018 >> data.txt
echo
file=data.txt


while IFS='-' read -r name id
do
        echo "Name: $name, ID: $id"
done < "$file"
```

output:

```
Name: Name Name, ID: ID
Name: MD. SHOHANUR RAHMAN SHOHAN, ID: 46013
Name: Rafiul Hassan , ID: 47048
Name: Abu Towsif, ID: 47019
Name: Farjana Opi, ID: 47018
```

#function

```
myFunction()
{
        echo "Oh! Actually, it Works"
}
myFunction
```

output:

```
Oh! Actually, it Works
```

```
#checking file exsisting
filename="info.txt"  # File name to check

if [ -f "$filename" ]; then
   # Perform your operations here if the file exists
   echo "File exists. Performing operation..."
   # Example operation: Displaying the content of the file
   cat "$filename"
else
   echo "File does not exist."
fi
```