

1. Arithmetic

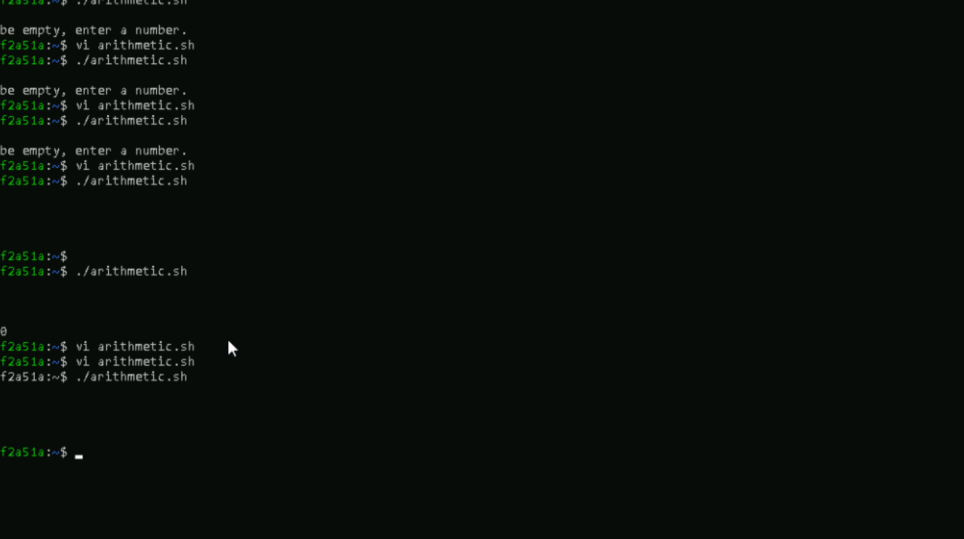
```
root1@66c459cb772a51a: ~
#1/usr/bin/bash

read -p "input1 : " inp1
if [[ -z $inp1 ]]
then
    echo "input 1 cannot be empty, enter a number."
exit
fi

read -p "input2 : " inp2
if [[ -z $inp2 ]]
then
    echo "input2 cannot be empty, enter a number."
exit
fi

bc_val=`echo "$inp1+$inp2" | bc`
echo "BC Value : $bc_val"

expr_val=`expr $inp1 + $inp2`
echo "Expr value : $expr_val"
```



```
root@1066c459cb7f2a51a: ~  
input1 : 2  
input 1 cannot be empty, enter a number.  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 89  
input 1 cannot be empty, enter a number.  
root@1066c459cb7f2a51a:~$ vi arithmetic.sh  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 5  
input 1 cannot be empty, enter a number.  
root@1066c459cb7f2a51a:~$ vi arithmetic.sh  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 4  
input 1 cannot be empty, enter a number.  
root@1066c459cb7f2a51a:~$ vi arithmetic.sh  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 4  
input2 : 4  
BC Value : 8  
Expr value : 8  
root@1066c459cb7f2a51a:~$  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 88  
input2 : 2  
BC Value : 100  
Expr value : 100  
root@1066c459cb7f2a51a:~$ vi arithmetic.sh  
root@1066c459cb7f2a51a:~$ vi arithmetic.sh  
root@1066c459cb7f2a51a:~$ ./arithmetic.sh  
input1 : 45  
input2 : 35  
BC Value : 80  
Expr value : 80  
root@1066c459cb7f2a51a:~$
```

2.PALINDROME

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2
palindrome.sh
#!/bin/bash

echo "Enter the number:"
read n

# Input validation to ensure it's a positive integer
if ! [[ "$n" =~ ^([0-9]+)$ ]]; then
    echo "Please enter a valid positive integer."
    exit 1
fi

num=$n
rev=0

# Reverse the number
while [ $n -gt 0 ]; do
    a=$((n % 10))
    n=$((n / 10))
    rev=$((rev * 10 + a))
done

echo "Reversed Number: $rev"

# Check if the number is a palindrome
if [ $num -eq $rev ]; then
    echo "The number is a palindrome!"
else
    echo "The number is not a palindrome!"
fi
```

```
root1@66c459cb7f2a51a: ~/practise
/
root1@66c459cb7f2a51a:/$ cd home
root1@66c459cb7f2a51a:/home$ ls
root1
root1@66c459cb7f2a51a:/home$ cd root1
root1@66c459cb7f2a51a:~$ ls
arithmetic.sh  harish  hi  snap
root1@66c459cb7f2a51a:~$ mkdir practise
root1@66c459cb7f2a51a:~$ cd practise
root1@66c459cb7f2a51a:~/practise$ vi palindrome.sh
root1@66c459cb7f2a51a:~/practise$ vi palindrome.sh
root1@66c459cb7f2a51a:~/practise$ vi palindrome.sh
root1@66c459cb7f2a51a:~/practise$ ls
palindrome.sh
root1@66c459cb7f2a51a:~/practise$ ll
total 8
drwxr-xr-x  2 root1 root1 4096 Jan 24 04:18 ./
drwxrwxrwx 12 root1 root1 4096 Jan 24 04:18 ../
root1@66c459cb7f2a51a:~/practise$ touch palindrome.sh
root1@66c459cb7f2a51a:~/practise$ ls
palindrome.sh
root1@66c459cb7f2a51a:~/practise$ vi palindrome.sh
root1@66c459cb7f2a51a:~/practise$ sudo nano palindrome.sh
[sudo] password for root1:
root1@66c459cb7f2a51a:~/practise$ chmod a+x palindrome.sh
root1@66c459cb7f2a51a:~/practise$ ./palindrome.sh
Enter the number:
121
Reversed Number: 121
The number is a palindrome!
root1@66c459cb7f2a51a:~/practise$ ./palindrome.sh
Enter the number:
45
Reversed Number: 54
The number is not a palindrome!
root1@66c459cb7f2a51a:~/practise$ sudo nano palindrome.sh
root1@66c459cb7f2a51a:~/practise$ ./palindrome.sh
Enter the number:
1661
Reversed Number: 1661
The number is a palindrome!
root1@66c459cb7f2a51a:~/practise$
```

3. REVERSE A NUMBER

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2 reverse_number.sh
#!/bin/bash

echo "Enter a number:"
read n # Read input from the user

rev=0

while [ $n -gt 0 ]; do
    sd=$((n % 10)) # Extract the last digit
    rev=$((rev * 10 + sd)) # Append it to the reversed number
    n=$((n / 10)) # Remove the last digit
done

echo "Reversed Number is $rev"
```

```
root1@66c459cb7f2a51a: ~/practise
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
./reverse_number.sh: line 8: [: -gt: unary operator expected
Reverse Number is 0
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh 2345
Reverse Number is 5432
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
Enter a number:
246
Reversed Number is 642
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
Enter a number:
67854
Reversed Number is 45876
root1@66c459cb7f2a51a:~/practise$
```

4. ARRAY SUM

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2 arraysom.sh
#!/bin/bash

echo "Enter the elements of the array separated by space:"
read -a arr # Read array input from the user

sum=0 # Initialize sum to 0

for (( i = 0; i < ${#arr[@]}; i++ )); do
    if (( arr[i] > 0 )); then
        sum=$((sum + arr[i])) # Use arithmetic expansion to calculate sum
    fi
done

echo "Sum of positive numbers: $sum"
```

```
root1@66c459cb7f2a51a: ~/practise
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh 2345
Reverse Number is 5432
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
Enter a number:
246
Reversed Number is 642
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
Enter a number:
67854
Reversed Number is 45876
root1@66c459cb7f2a51a:~/practise$ sudo nano arraysom.sh
root1@66c459cb7f2a51a:~/practise$ sudo chmod a+x arraysom.sh
root1@66c459cb7f2a51a:~/practise$ ./arraysom.sh
Enter the elements of the array separated by space:
2 4 6 8
Sum of positive numbers: 20
root1@66c459cb7f2a51a:~/practise$ sudo nano arraysom.sh
root1@66c459cb7f2a51a:~/practise$ ./arraysom.sh
Enter the elements of the array separated by space:
7 8 4 0 5 2
Sum of positive numbers: 26
root1@66c459cb7f2a51a:~/practise$
```

5. PASCAL TRIANGLE

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2 pascaltriangle.sh
#!/bin/bash

# Function to print Pascal's triangle
pastri() {
    r=$1 # Number of rows
    c=1
    for ((i = 0; i < r; i++)); do
        for ((s = 1; s <= r - i; s++)); do
            echo -n " " # Print spaces for alignment
        done
        for ((j = 0; j <= i; j++)); do
            if [ $j -eq 0 -o $i -eq 0 ]; then
                c=1
            else
                c=$((c * (i - j + 1) / j))
            fi
            echo -n "$c " # Print the value
        done
        echo # New line after each row
    done
}

echo "Enter the number of rows for Pascal's triangle:"
read r # Take input from the user
pastri $r
```

```
root1@66c459cb7f2a51a: ~/practise
246
Reversed Number is 642
root1@66c459cb7f2a51a:~/practise$ sudo nano reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ ./reverse_number.sh
Enter a number:
67854
Reversed Number is 45876
root1@66c459cb7f2a51a:~/practise$ sudo nano arraysum.sh
root1@66c459cb7f2a51a:~/practise$ sudo chmod a+x arraysum.sh
root1@66c459cb7f2a51a:~/practise$ ./arraysum.sh
Enter the elements of the array separated by space:
2 4 6 8
Sum of positive numbers: 20
root1@66c459cb7f2a51a:~/practise$ sudo nano arraysum.sh
root1@66c459cb7f2a51a:~/practise$ ./arraysum.sh
Enter the elements of the array separated by space:
7 8 4 0 5 2
Sum of positive numbers: 26
root1@66c459cb7f2a51a:~/practise$ sudo cp arraysum.sh pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ sudo nano pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ ./pascaltriangle.sh
Enter the number of rows for Pascal's triangle:
5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
root1@66c459cb7f2a51a:~/practise$ sudo nano pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ ./pascaltriangle.sh
Enter the number of rows for Pascal's triangle:
8
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
root1@66c459cb7f2a51a:~/practise$
```

6. BUBBLE SORT

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2
bubblesort.sh *
#!/bin/bash

echo "Enter the elements of the array separated by space:"
read -a arr # Read array input from the user

echo "Entered array:"
echo "${arr[@]}"

# Bubble sort algorithm
n=${#arr[@]} # Length of the array
for ((i = 0; i < n; i++)); do
    for ((j = 0; j < n - i - 1; j++)); do
        if [ ${arr[j]} -gt ${arr[j + 1]} ]; then
            # Swap elements
            temp=${arr[j]}
            arr[j]=${arr[j + 1]}
            arr[j + 1]=$temp
        fi
    done
done

echo "Sorted array:"
echo "${arr[@]}"
```

```
root1@66c459cb7f2a51a: ~/practise
Sum of positive numbers: 20
root1@66c459cb7f2a51a:~/practise$ sudo nano arraysum.sh
root1@66c459cb7f2a51a:~/practise$ ./arraysum.sh
Enter the elements of the array separated by space:
7 8 4 0 5 2
Sum of positive numbers: 26
root1@66c459cb7f2a51a:~/practise$ sudo cp arraysum.sh pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ sudo nano pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ ./pascaltriangle.sh
Enter the number of rows for Pascal's triangle:
5
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
root1@66c459cb7f2a51a:~/practise$ sudo nano pascaltriangle.sh
root1@66c459cb7f2a51a:~/practise$ ./pascaltriangle.sh
Enter the number of rows for Pascal's triangle:
8
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
root1@66c459cb7f2a51a:~/practise$ sudo cp arraysum.sh bubblesort.sh
root1@66c459cb7f2a51a:~/practise$ sudo nano bubblesort.sh
root1@66c459cb7f2a51a:~/practise$ ./bubblesort.sh
-bash: ./bubblesort.sh: No such file or directory
root1@66c459cb7f2a51a:~/practise$ ./bubblesort.sh
Enter the elements of the array separated by space:
8 3 6 1 6 8 3 0
Entered array:
8 3 6 1 6 8 3 0
Sorted array:
0 1 3 3 6 6 8 8
root1@66c459cb7f2a51a:~/practise$
```

7. Personalized Message

```
root1@66c459cb7f2a51a: ~/practise
GNU nano 7.2
demo.sh *
echo "Hello!, what's your name?"
read name
echo "Hello $name welcome to japan"
```

```
root1@66c459cb7f2a51a: ~/practise
root1@66c459cb7f2a51a:/$ sudo su
[sudo] password for root1:
Sorry, try again.
[sudo] password for root1:
Sorry, try again.
[sudo] password for root1:
sudo: 3 incorrect password attempts
root1@66c459cb7f2a51a:/$ sudo ls /root
[sudo] password for root1:
Sorry, try again.
[sudo] password for root1:
sudo: 1 incorrect password attempt
root1@66c459cb7f2a51a:/$ cd /home
root1@66c459cb7f2a51a:/home$ cd root1
root1@66c459cb7f2a51a:~$ ls
arithmetic.sh  harish  hi  practise  snap
root1@66c459cb7f2a51a:~$ cd practise
root1@66c459cb7f2a51a:~/practise$ ls
arraysum.sh  bubblesort.sh  palindrome.sh  pascaltriangle.sh  reverse_number.sh
root1@66c459cb7f2a51a:~/practise$ sudo nano demo.sh
[sudo] password for root1:
Sorry, try again.
[sudo] password for root1:
Sorry, try again.
[sudo] password for root1:
sudo: 2 incorrect password attempts
root1@66c459cb7f2a51a:~/practise$ nano demo.sh
root1@66c459cb7f2a51a:~/practise$ chmod a+x demo.sh
root1@66c459cb7f2a51a:~/practise$ ./demo.sh
Hello!, what's your name?
Nagi
Hello Nagi welcome to japan
root1@66c459cb7f2a51a:~/practise$
```