

## Linux Programs

Program 1- shell script to read an input.

```
samim@9819fd877cfb527:~$ sudo nano hello.sh
samim@9819fd877cfb527:~$ sudo chmod +x hello.sh
samim@9819fd877cfb527:~$ ./hello.sh
Hello, what is your name?
Samim
Hello Samim, nice to meet you!
samim@9819fd877cfb527:~$ cat hello.sh
echo "Hello, what is your name?"
read name
echo "Hello $name, nice to meet you!"
```

Program 2- shell script to add 2 numbers.

```
samim@9819fd877cfb527:~$ sudo nano add.sh
samim@9819fd877cfb527:~$ cat add.sh
#!/usr/bin/bash
read -p "Input1 : " inp1
if [[ -z $inp1 ]]
then
    echo "Input 1 cannot be empty, please enter an integer."
    exit
fi

read -p "Input2 : " inp2
if [[ -z $inp2 ]]
then
    echo "Input 2 cannot be empty, please enter an integer."
    exit
fi

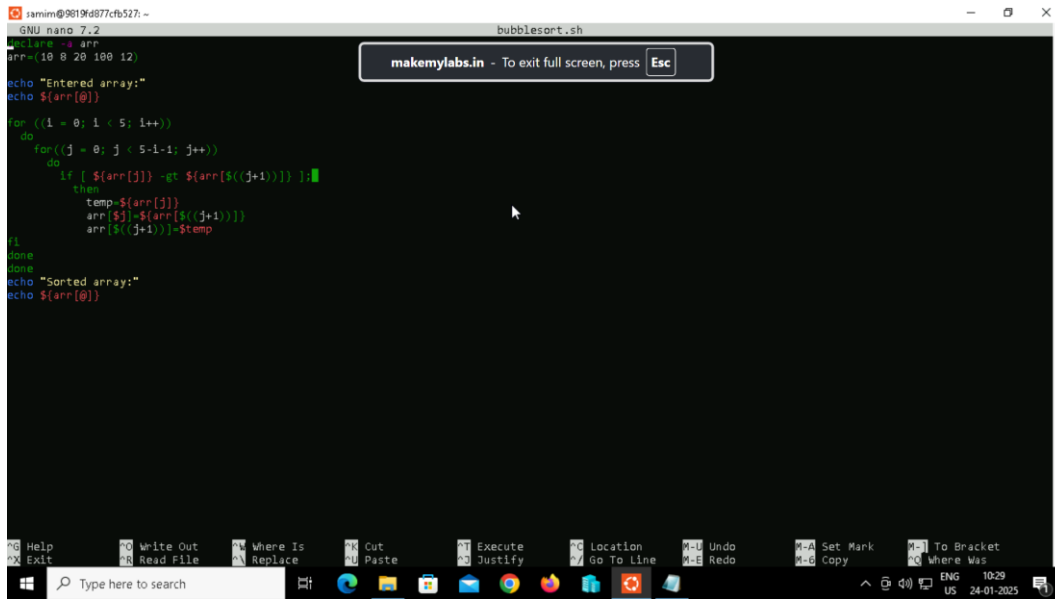
bc_val=`echo "$inp1+$inp2" | bc`
echo "BC Value : $bc_val"

expr_val=`expr $inp1 + $inp2`
echo "EXPR Value : $expr_val"
samim@9819fd877cfb527:~$ ./add.sh
Input1 : 10
Input2 : 20
BC Value : 30
EXPR Value : 30
samim@9819fd877cfb527:~$
```

Program 3- shell script for bubble sort.

```
samim@9819fd877cfb527:~$ sudo nano bubblesort.sh
samim@9819fd877cfb527:~$ chmod +x bubblesort.sh
chmod: changing permissions of 'bubblesort.sh': Operation not permitted
samim@9819fd877cfb527:~$ sudo chmod +x bubb
chmod: cannot access 'bubb': No such file or directory
samim@9819fd877cfb527:~$ sudo chmod +x bubblesort.sh
samim@9819fd877cfb527:~$ ./bubblesort.sh
Entered array:
10 8 20 100 12
Sorted array:
8 10 12 20 100
```

# Linux Programs



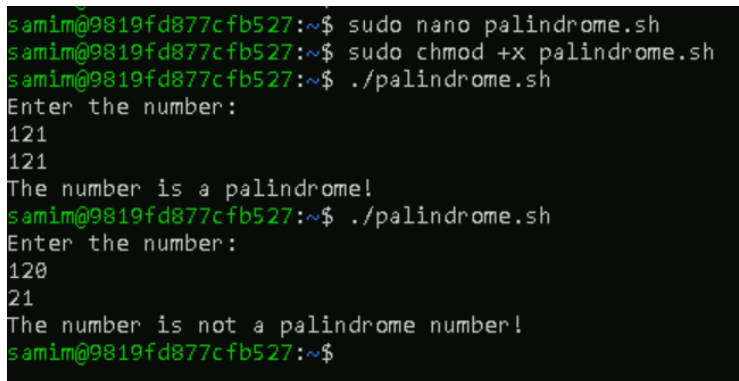
```
GNU nano 7.2      bubblesort.sh
clear -s arr
arr=(10 8 20 100 12)

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

for ((i = 0; i < 5; i++))
do
    for ((j = 0; j < 5-i-1; j++))
    do
        if [ ${arr[j]} -gt ${arr[j+1]} ]; then
            temp=${arr[j]}
            arr[j]=${arr[j+1]}
            arr[j+1]=$temp
        fi
    done
done
echo "Sorted array:"
echo ${arr[@]}
```

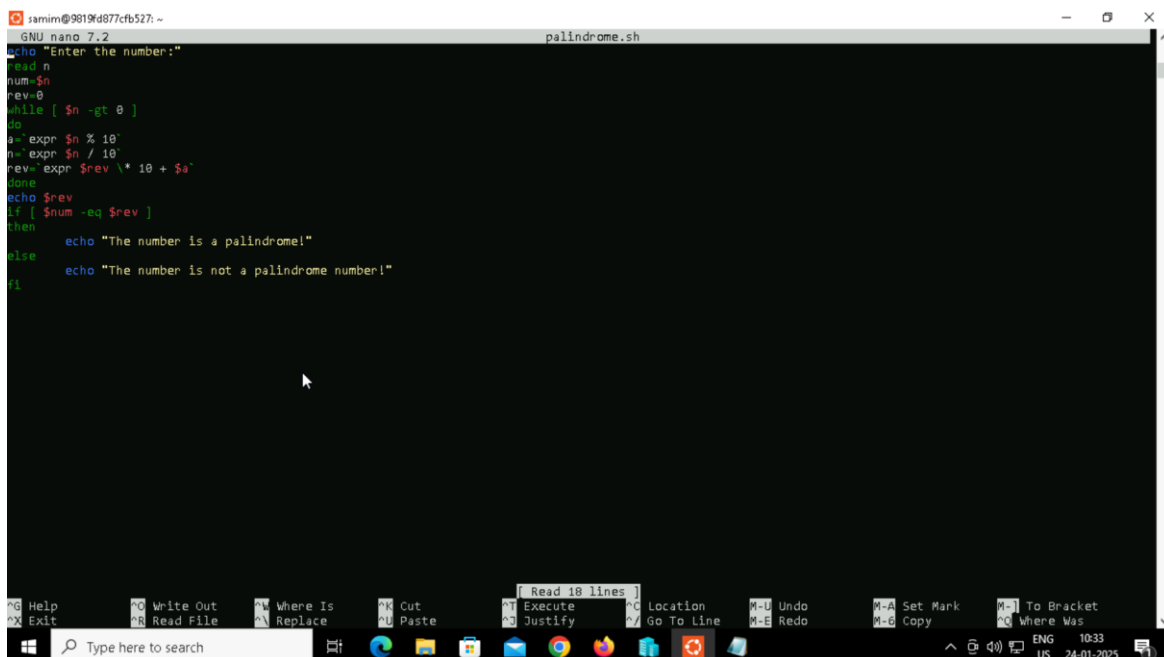
The screenshot shows a terminal window with the nano text editor open to a file named 'bubblesort.sh'. The script defines an array 'arr' with values 10, 8, 20, 100, and 12. It uses two nested loops to perform a bubble sort. The outer loop runs from i=0 to i=4, and the inner loop runs from j=0 to j=5-i-1. Inside the inner loop, it compares adjacent elements and swaps them if they are in the wrong order. After the sort, it prints the sorted array. The terminal window has a Windows taskbar at the bottom with various application icons and a search bar.

Program 4- shell script for checking if a number is palindrome or not.



```
samim@9819fd877cfb527:~$ sudo nano palindrome.sh
samim@9819fd877cfb527:~$ sudo chmod +x palindrome.sh
samim@9819fd877cfb527:~$ ./palindrome.sh
Enter the number:
121
121
The number is a palindrome!
samim@9819fd877cfb527:~$ ./palindrome.sh
Enter the number:
120
21
The number is not a palindrome number!
samim@9819fd877cfb527:~$
```

The screenshot shows a terminal window where the user has created a script named 'palindrome.sh' using 'nano', made it executable with 'chmod +x', and then run it twice. In the first run, the user enters '121' and the script outputs 'The number is a palindrome!'. In the second run, the user enters '120' and the script outputs 'The number is not a palindrome number!'. The terminal window has a Windows taskbar at the bottom.



```
GNU nano 7.2      palindrome.sh
echo "Enter the number:"
read n
num=$n
rev=0
while [ $n -gt 0 ]
do
    a=expr $n % 10
    n=expr $n / 10
    rev=expr $rev \* 10 + $a
done
echo $rev
if [ $num -eq $rev ]
then
    echo "The number is a palindrome!"
else
    echo "The number is not a palindrome number!"
fi
```

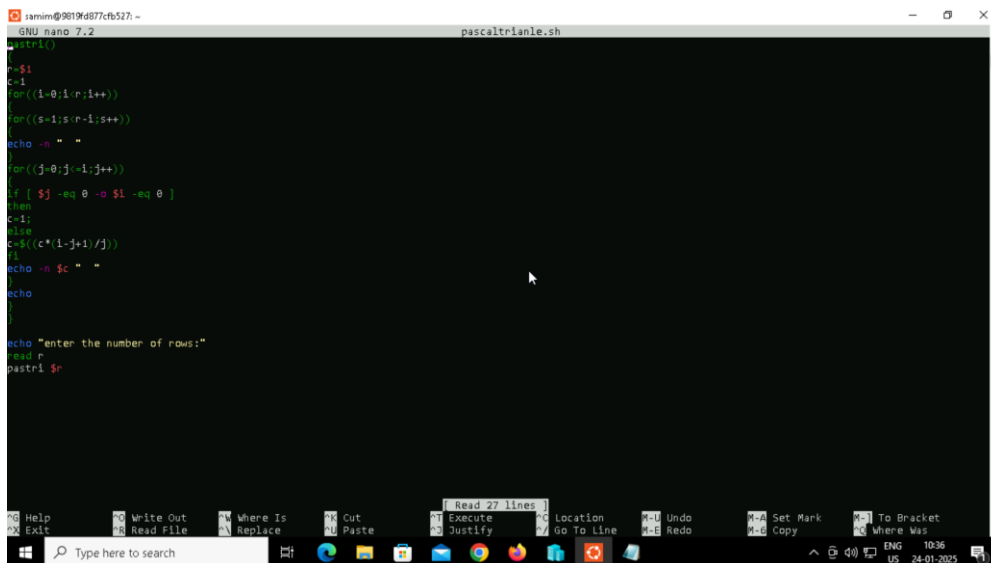
The screenshot shows a terminal window with the nano text editor open to a file named 'palindrome.sh'. The script prompts the user to enter a number, reads it into 'n', and then calculates its reverse by repeatedly extracting the last digit and building a new number 'rev'. Finally, it compares the original number 'num' with the reversed number 'rev' and prints the appropriate message. The terminal window has a Windows taskbar at the bottom.

# Linux Programs

## Program 5- Pascal Triangle

```
samim@9819fd877cfb527:~$ sudo nano palindrome.sh
sami@9819fd877cfb527:~$ sudo nano pascaltrianle.sh
sami@9819fd877cfb527:~$ sudo chmod +x pascaltrianle.sh
sami@9819fd877cfb527:~$ ./pascaltrianle.sh
enter the number of rows:
5

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

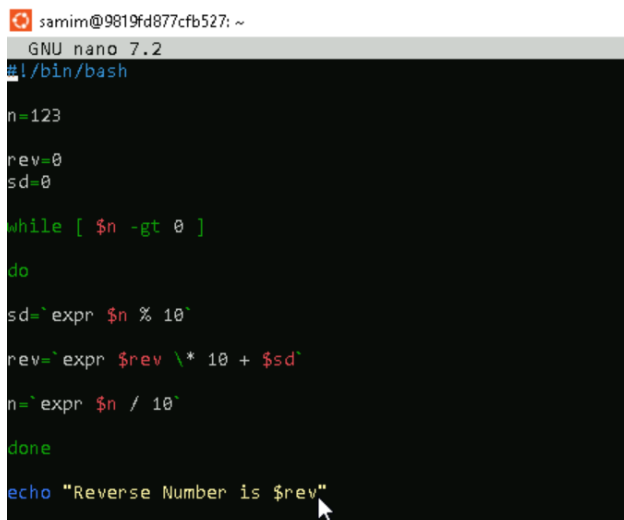


```
sami@9819fd877cfb527:~$ nano pascaltrianle.sh
GNU nano 7.2 pascaltrianle.sh
#!/bin/bash
n=$1
c=1
for ((i=0;i<n;i++))
do
    s=1
    for ((j=0;j<=i;j++))
    do
        if [ $j -eq 0 -o $j -eq $i ]; then
            c=1
        else
            c=$((c*(i-j+1)/j))
        fi
        echo -n "$c "
    done
    echo
done

echo "enter the number of rows:"
read n
pascal $n
```

## Program 6- to reverse a number

```
sami@9819fd877cfb527:~$ sudo nano revnum.sh
[sudo] password for sami:
sami@9819fd877cfb527:~$ chmod +x revnum.sh
chmod: changing permissions of 'revnum.sh': Operation not permitted
sami@9819fd877cfb527:~$ sudo chmod +x revnum.sh
sami@9819fd877cfb527:~$ ./revnum.sh
./revnum.sh: line 8: [: -gt: unary operator expected
Reverse Number is 0
sami@9819fd877cfb527:~$ sudo nano revnum.sh
sami@9819fd877cfb527:~$ ./revnum.sh
Reverse Number is 321
```



```
sami@9819fd877cfb527:~$ nano revnum.sh
GNU nano 7.2
#!/bin/bash

n=123
rev=0
sd=0

while [ $n -gt 0 ]
do
    sd=`expr $n % 10`
    rev=`expr $rev \* 10 + $sd`
    n=`expr $n / 10`
done

echo "Reverse Number is $rev"
```

## Linux Programs

### Program 7- sum of an array

```
samim@9819fd877cfb527:~$ sudo nano arr_sum.sh
samim@9819fd877cfb527:~$ chmod +x arr_sum.sh
chmod: changing permissions of 'arr_sum.sh': Operation not permitted
samim@9819fd877cfb527:~$ sudo chmod +x arr_sum.sh
samim@9819fd877cfb527:~$ ./arr_sum.sh
27
```

```
samim@9819fd877cfb527: ~
GNU nano 7.2

arr=(2 4 -5 -8 9 12)

for (( i = 0; i <= ${#arr[*]}; i++ )); do
    if (( arr[i] > 0 )); then
        sum=`expr $sum + ${arr[i]}`
    fi
done
echo "$sum"
```