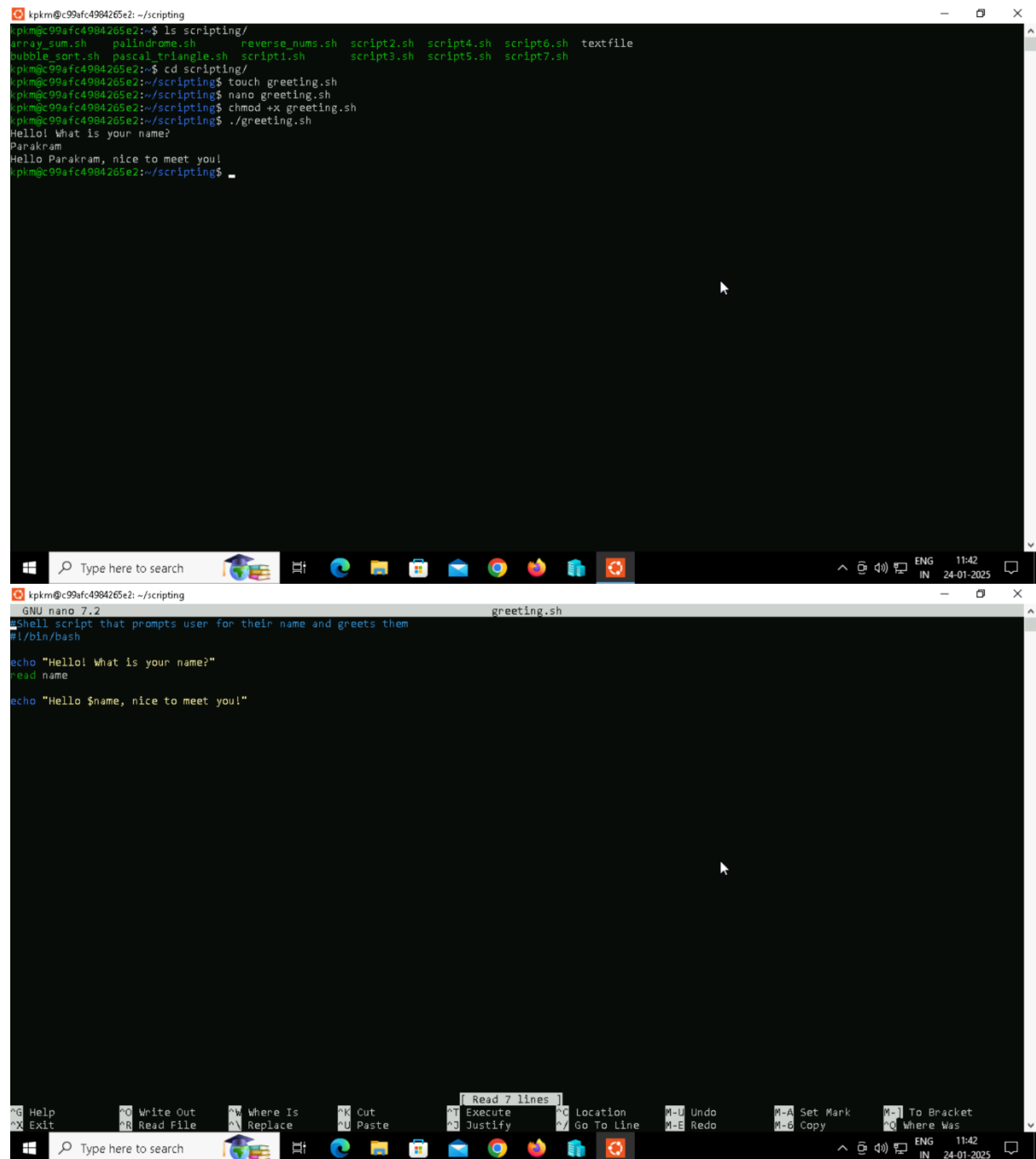


Q. Customized greeting for the user



The image shows a Windows desktop with two terminal windows. The top window is a standard command prompt showing the user navigating to a directory named 'scripting' and creating a new file 'greeting.sh'. The user then runs the script, which prompts for a name and displays a personalized greeting. The bottom window is a nano text editor editing the 'greeting.sh' file. The code in the editor uses 'echo' to display messages and 'read' to capture user input.

```
kpkmc99afc4984265e2: ~/scripting
kpkmc99afc4984265e2:~$ ls scripting/
array_sum.sh  palindrome.sh  reverse_nums.sh  script2.sh  script4.sh  script6.sh  textfile
bubble_sort.sh  pascal_triangle.sh  script1.sh  script3.sh  script5.sh  script7.sh
kpkmc99afc4984265e2:~$ cd scripting/
kpkmc99afc4984265e2:~/scripting$ touch greeting.sh
kpkmc99afc4984265e2:~/scripting$ nano greeting.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x greeting.sh
kpkmc99afc4984265e2:~/scripting$ ./greeting.sh
Hello! What is your name?
Parakram
Hello Parakram, nice to meet you!
kpkmc99afc4984265e2:~/scripting$
```

```
GNU nano 7.2 greeting.sh
#Shell script that prompts user for their name and greets them
#!/bin/bash

echo "Hello! What is your name?"
read name

echo "Hello $name, nice to meet you!"
```

Q. Arithmetic operations

```
kpkmc99afc4984265e2: ~/scripting
kpkmc99afc4984265e2:~$ ls
backup.sh  scripting  snap
kpkmc99afc4984265e2:~$ cd scripting/
kpkmc99afc4984265e2:~/scripting$ ls
script1.sh script2.sh script3.sh script4.sh script5.sh script6.sh textfile
kpkmc99afc4984265e2:~/scripting$ touch script7.sh
kpkmc99afc4984265e2:~/scripting$ nano script7.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x script7.sh
kpkmc99afc4984265e2:~/scripting$ ./script7.sh
Input1 : 32
Input2 : 34
BC Value : 66
EXPR Value : 66
kpkmc99afc4984265e2:~/scripting$ ./script7.sh
Input1 : 23
Input2 : 47
BC Value : 70
EXPR Value : 70
kpkmc99afc4984265e2:~/scripting$ ./script7.sh
Input1 : 23
Input2 : 46
BC Value : 69
EXPR Value : 69
kpkmc99afc4984265e2:~/scripting$
```

array_sum.sh

```
GNU nano 7.2
# Shell script to find the sum of array elements
#!/bin/bash

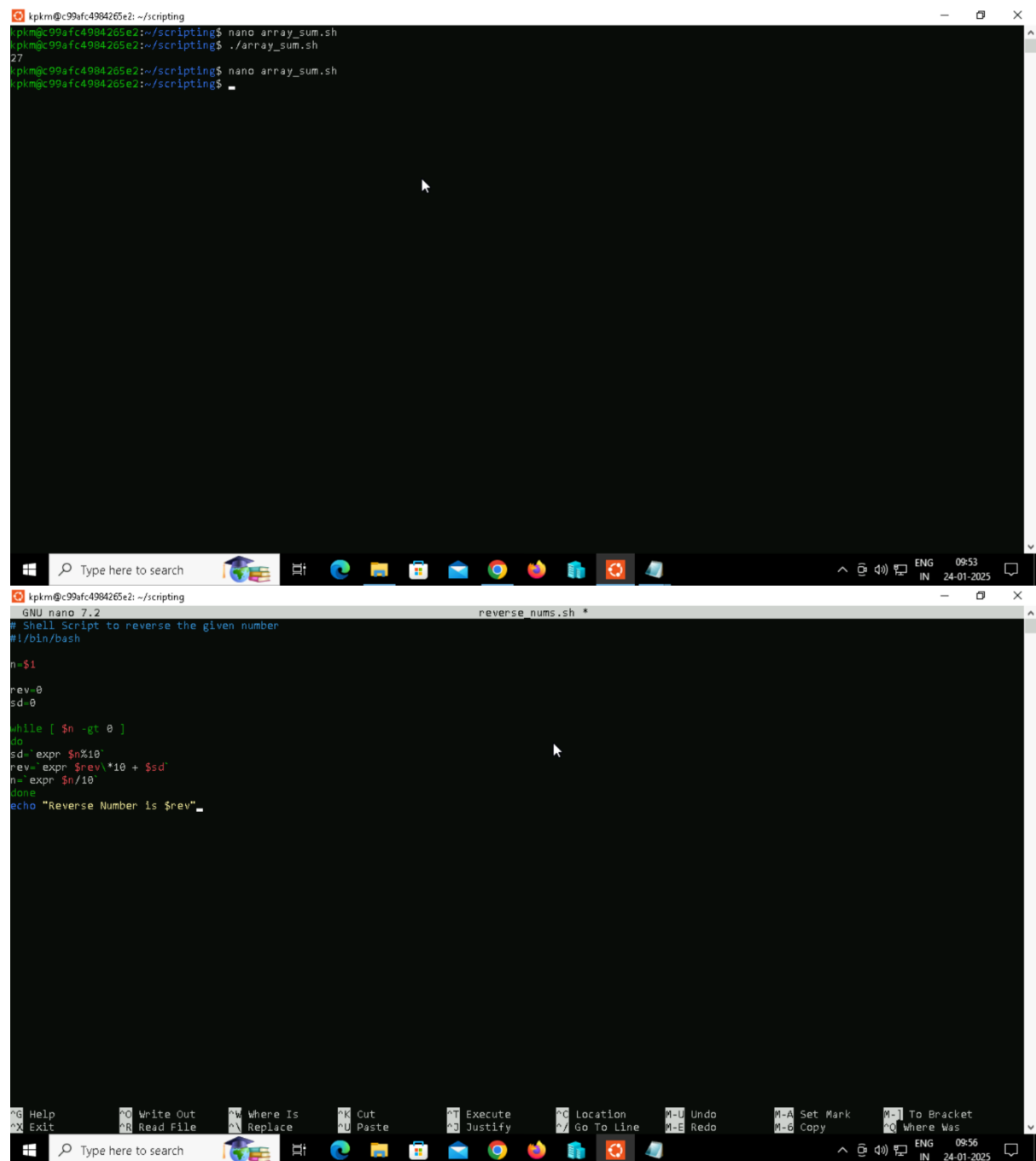
arr=(2 4 -5 -8 9 12)
sum=0

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

Wrote 12 lines

Help Exit Write Out Read File Where Is Replace Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark Copy To Bracket Where Was

Q. Sum of array



```
kpkm@c99afc4984265e2: ~/scripting
kpkm@c99afc4984265e2:~/scripting$ nano array_sum.sh
kpkm@c99afc4984265e2:~/scripting$ ./array_sum.sh
27
kpkm@c99afc4984265e2:~/scripting$ nano array_sum.sh
kpkm@c99afc4984265e2:~/scripting$
```

```
GNU nano 7.2 reverse_nums.sh *
# Shell Script to reverse the given number
#!/bin/bash

n=$1
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"
```

Q. Reverse a number

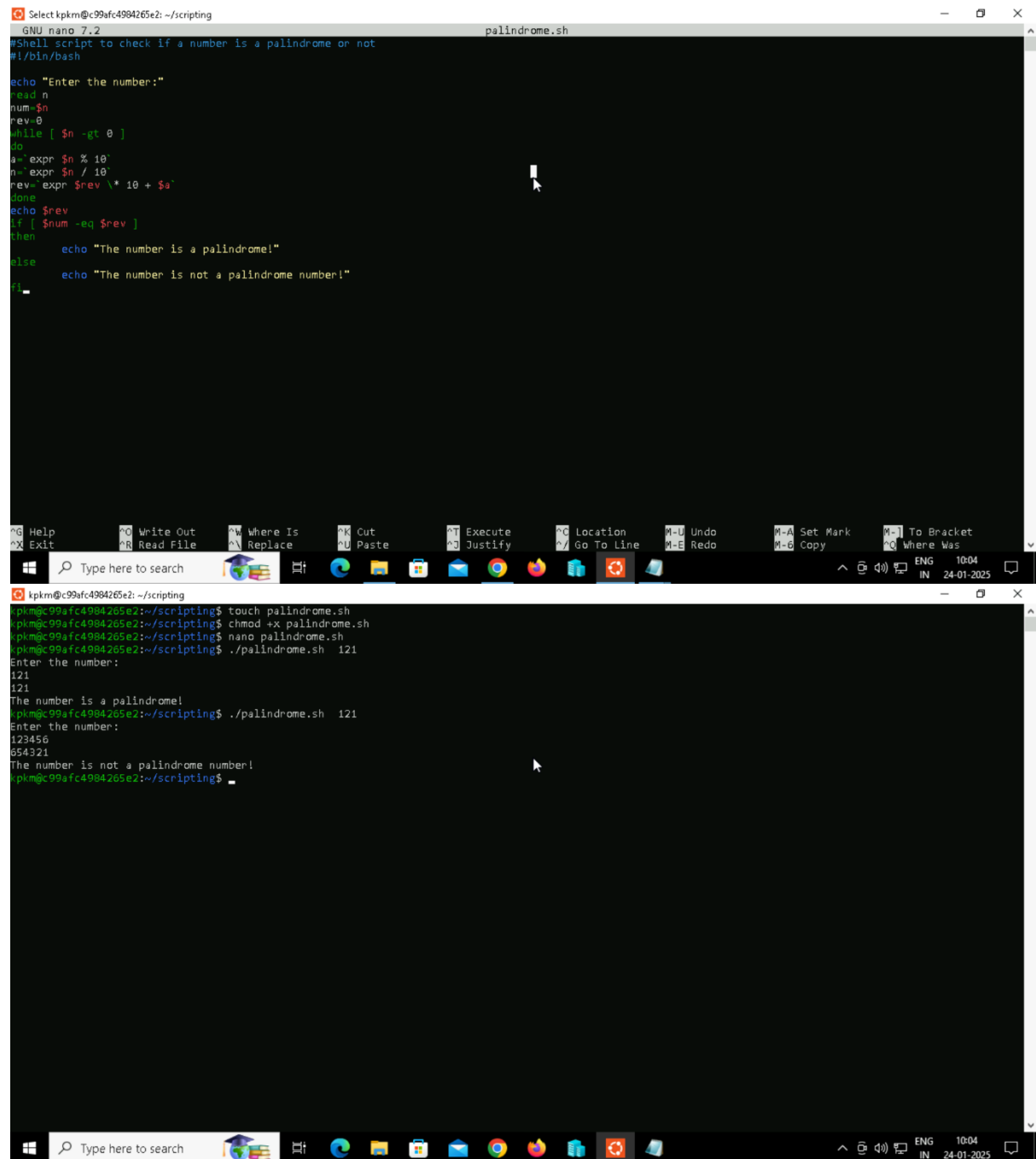
```
kpkmc99afc4984265e2: ~/scripting
kpkmc99afc4984265e2:~/scripting$ nano array_sum.sh
kpkmc99afc4984265e2:~/scripting$ ./array_sum.sh
27
kpkmc99afc4984265e2:~/scripting$ nano array_sum.sh
kpkmc99afc4984265e2:~/scripting$ nano array_sum.sh
kpkmc99afc4984265e2:~/scripting$ ./array_sum.sh
14
kpkmc99afc4984265e2:~/scripting$ nano reverse_nums.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x reverse_nums.sh
kpkmc99afc4984265e2:~/scripting$ ./reverse_nums.sh
./reverse_nums.sh: line 9: [: -gt: unary operator expected
Reverse Number is 0
kpkmc99afc4984265e2:~/scripting$ nano reverse_nums.sh
kpkmc99afc4984265e2:~/scripting$ ./reverse_nums.sh 346
expr: non-integer argument
./reverse_nums.sh: line 9: [: 346/10: integer expression expected
Reverse Number is
kpkmc99afc4984265e2:~/scripting$ nano reverse_nums.sh
kpkmc99afc4984265e2:~/scripting$ ./reverse_nums.sh 346
./reverse_nums.sh: line 1: #: command not found
Reverse Number is 643
kpkmc99afc4984265e2:~/scripting$ nano reverse_nums.sh
kpkmc99afc4984265e2:~/scripting$
```

```
GNU nano 7.2 reverse_nums.sh *
# Shell Script to reverse the given number
#!/bin/bash

n=$1
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"
```

Q. Palindrome check



The image consists of two screenshots of a terminal window. The top screenshot shows the creation of a shell script named `palindrome.sh` using `GNU nano 7.2`. The script prompts the user to enter a number and checks if it is a palindrome by reversing the digits and comparing them. The bottom screenshot shows the execution of the script with test cases: `121` (which is a palindrome) and `123456` (which is not a palindrome).

```
GNU nano 7.2
palindrome.sh
#Shell script to check if a number is a palindrome or not
#!/bin/bash

echo "Enter the number:"
read n
num=$n
rev=0
while [ $n -gt 0 ]
do
a=$((n % 10))
n=$((n / 10))
rev=$((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
_
```

```
kpkmc99afc4984265e2: ~/scripting
kpkmc99afc4984265e2:~/scripting$ touch palindrome.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x palindrome.sh
kpkmc99afc4984265e2:~/scripting$ nano palindrome.sh
kpkmc99afc4984265e2:~/scripting$ ./palindrome.sh 121
Enter the number:
121
121
The number is a palindrome!
kpkmc99afc4984265e2:~/scripting$ ./palindrome.sh 121
Enter the number:
123456
654321
The number is not a palindrome number!
kpkmc99afc4984265e2:~/scripting$
```

Q. Bubble sort of static array

```
kpkmc@c99afc4984265e2: ~/scripting
GNU nano 7.2 bubble_sort.sh
#Bubble sort in shell with static array
#!/bin/bash

declare -a arr
arr=(10 20 40 -21 -8 16)

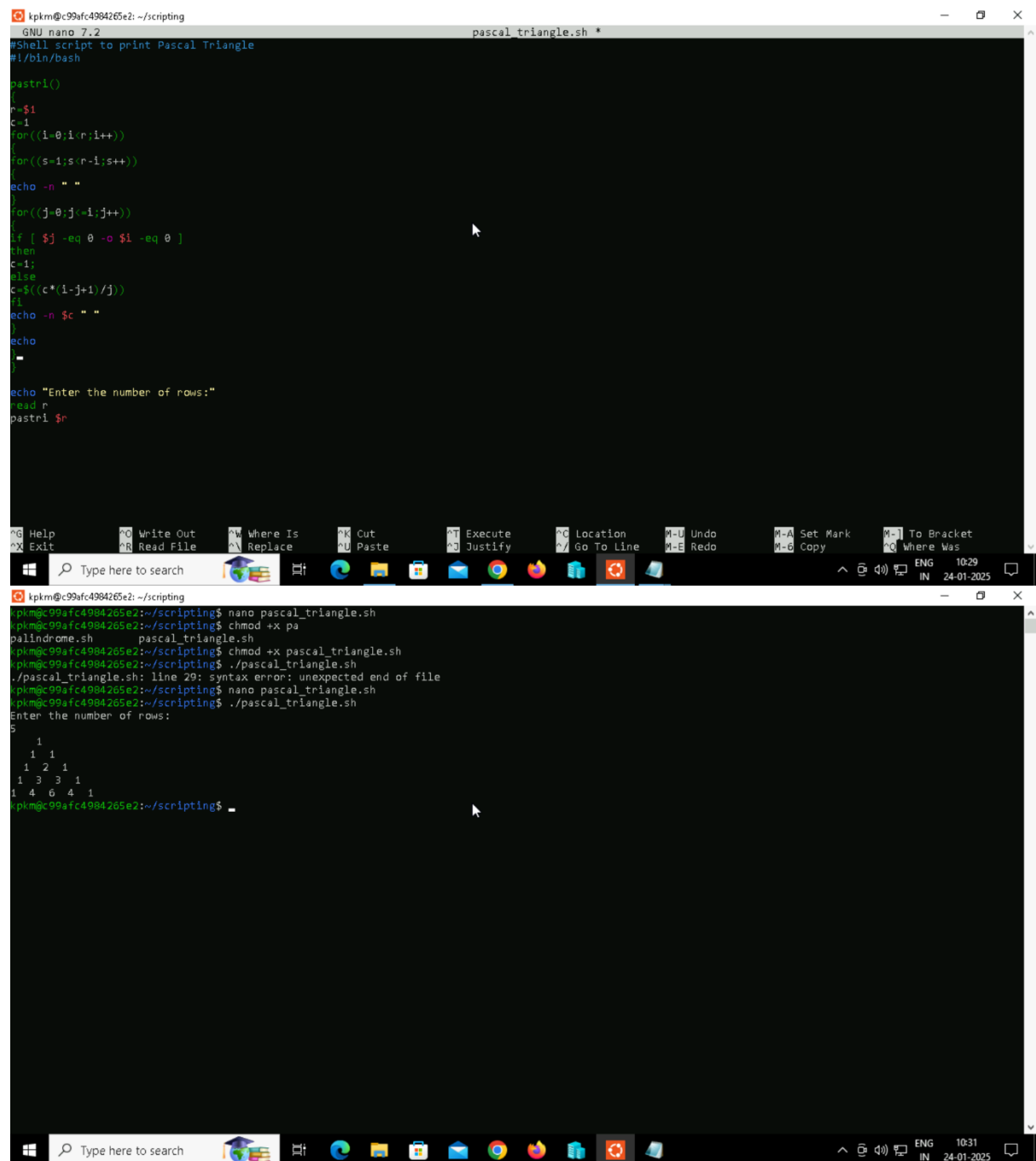
echo "Entered array:"
echo ${arr[@]}

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[@]}

Wrote 23 lines
Help Write Out Where Is Cut Execute Location Undo Set Mark To Bracket
Exit Read File Replace Paste Justify Go To Line Redo Copy Where Was

kpkmc@c99afc4984265e2: ~/scripting
kpkmc@c99afc4984265e2:~/scripting$ nano bubble_sort.sh
kpkmc@c99afc4984265e2:~/scripting$ chmod +x bubble_sort.sh
kpkmc@c99afc4984265e2:~/scripting$ ./bubble_sort.sh
Entered array:
10 20 40 -21 -8 16
./bubble_sort.sh: line 12: syntax error: arithmetic expression required
./bubble_sort.sh: line 12: syntax error: `((i = 0; i < 5; i++))'
kpkmc@c99afc4984265e2:~/scripting$ chmod +x bubble_sort.sh
kpkmc@c99afc4984265e2:~/scripting$ nano bubble_sort.sh
kpkmc@c99afc4984265e2:~/scripting$ ./bubble_sort.sh
Entered array:
10 20 40 -21 -8 16
Sorted array:
-21 -8 10 20 40 16
kpkmc@c99afc4984265e2:~/scripting$
```

Q. Pascal's triangle



The image consists of two screenshots of a terminal window. The top screenshot shows the nano text editor editing a file named `pascal_triangle.sh`. The script is a shell script that prints Pascal's triangle. The bottom screenshot shows the terminal after running the script, displaying the Pascal's triangle for 5 rows.

```
GNU nano 7.2 pascal_triangle.sh *
#Shell script to print Pascal Triangle
#!/bin/bash

pastr1()
{
  n=$1
  c=1
  for((i=0;i<n;i++))
  do
    for((s=1;s<n-i;s++))
    do
      echo -n " "
    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 "
    done
    echo
  done

  echo "Enter the number of rows:"
  read n
  pastr1 $n
}

kpkmc99afc4984265e2: ~/scripting
kpkmc99afc4984265e2:~/scripting$ nano pascal_triangle.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x pa
pallindrome.sh pascal_triangle.sh
kpkmc99afc4984265e2:~/scripting$ chmod +x pascal_triangle.sh
kpkmc99afc4984265e2:~/scripting$ ./pascal_triangle.sh
./pascal_triangle.sh: line 29: syntax error: unexpected end of file
kpkmc99afc4984265e2:~/scripting$ nano pascal_triangle.sh
kpkmc99afc4984265e2:~/scripting$ ./pascal_triangle.sh
Enter the number of rows:
5
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
kpkmc99afc4984265e2:~/scripting$
```