Project

1. Shell program to add two integer values. And check if any input is given or not.

Step -1: Created a Script file using touch command as **“Integer.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch integer.sh
root@8b5c7dd85f01583:~# nano integer.sh
root@8b5c7dd85f01583:~# chmod +x integer.sh
```

Program:

```
root@8b5c7dd85f01583: ~
  GNU nano 7.2                                  integer.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"
```

Output:

```
root@8b5c7dd85f01583:~# ./integer.sh
Input1 : 5
Input2 : 7
BC Value : 12
EXPR Value : 5+7
```

2. Simple example of a shell script that prompts the user for their name and greets them.

Step -1: Created a Script file using touch command as **“greet.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch greet.sh
root@8b5c7dd85f01583:~# nano greet.sh
root@8b5c7dd85f01583:~# chmod +x greet.sh
```

Program:

```
root@8b5c7dd85f01583: ~
  GNU nano 7.2                                   greet.sh
echo "Hello! What is your name?"
read name
echo "Hello, $name! Welcome to the world of Shell Scripting"
```

Output:

```
root@8b5c7dd85f01583:~# ./greet.sh
Hello! What is your name?
Pradhisha
Hello, Pradhisha! Welcome to the world of Shell Scripting
```

3. Script for Array sum

Step -1: Created a Script file using touch command as **“array_sum.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch array_sum.sh
root@8b5c7dd85f01583:~# nano array_sum.sh
root@8b5c7dd85f01583:~# chmod +x array_sum.sh
```

Program:

```
root@8b5c7dd85f01583: ~
  GNU nano 7.2                                array_sum.sh
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"
```

Output:

```
root@8b5c7dd85f01583:~# ./array_sum.sh
27
```

4. Script for Palindrome numbers

Step -1: Created a Script file using touch command as **“panlin.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch panlin.sh
root@8b5c7dd85f01583:~# nano panlin.sh
root@8b5c7dd85f01583:~# chmod +x panlin.sh
```

Program:

```
root@8b5c7dd85f01583: ~
  GNU nano 7.2                                panlin.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
```

Output:

```
root@8b5c7dd85f01583:~# ./panlin.sh
Enter the number:
151
151
The number is a palindrome!
```

5. Script for Bubblesort

Step -1: Created a Script file using touch command as **“bubblesort.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch bubblesort.sh
root@8b5c7dd85f01583:~# nano bubblesort.sh
root@8b5c7dd85f01583:~# chmod +x bubblesort.sh
```

Program:

```
 GNU nano 7.2                                  bubblesort.sh
declare -a 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[@]}
```

Output:

```
root@8b5c7dd85f01583:~# ./bubblesort.sh
Entered array:
10 8 20 100 12
Sorted array:
8 10 12 20 100
```

6. Script for Pascal Triangle

Step -1: Created a Script file using touch command as **“pascal.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch pascal.sh
root@8b5c7dd85f01583:~# nano pascal.sh
root@8b5c7dd85f01583:~# chmod +x pascal.sh
```

Program:

```
root@8b5c7dd85f01583: ~
 GNU nano 7.2                                  bubblesort.sh
declare -a 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[@]}
```

Output:

```
root@8b5c7dd85f01583:~# ./pascal.sh
enter the number of rows:
4
      1
    1   1
  1   2   1
1   3   3   1
root@8b5c7dd85f01583:~#
```

7. Script for Reverse Number

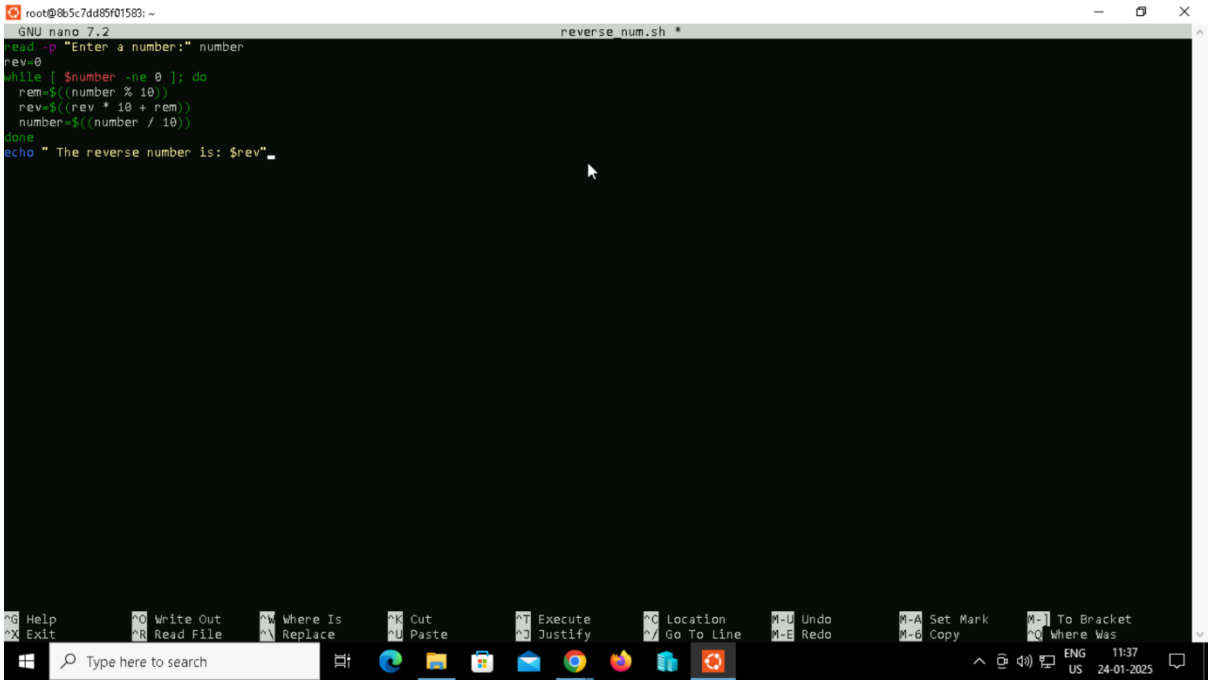
Step -1: Created a Script file using touch command as **“Reverse_num.sh”.**

Step -2: Opened a nano file to write the Script.

Step -3: Changes the file permission.

```
root@8b5c7dd85f01583:~# touch reverse_num.sh
root@8b5c7dd85f01583:~# nano reverse_num.sh
root@8b5c7dd85f01583:~# chmod +x reverse_num.sh
```

Program:



Output:

```
root@8b5c7dd85f01583:~# ./reverse_num.sh
Enter a number:2345
 The reverse number is: 5432
root@8b5c7dd85f01583:~#
```