PROJECT

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

Step 1: Create an "**integer.sh"** script file using touch command

Step 2: Create a nano file to write the code

```
root@serverid: ~
root@serverid:~# touch integer.sh
root@serverid:~# nano integer.sh
root@serverid:~# chmod +x integer.sh
root@serverid:~# ./integer.sh
```

Step 3: Write the Code in integer script file

```
root@serverid: ~
  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"
```

Step 4: Output

```
root@serverid:~# ./integer.sh
Input1: 10
Input2: 20
BC value: 30
EXPR Value: 10+20
root@serverid:~#
```

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

Step1: Create an "**hello.sh"** script file using touch command

Step 2: Create a nano file to write the code

```
root@serverid:~# touch hello.sh
root@serverid:~# nano hello.sh
```

Step3 : Write the Code in nano hello.sh script file

```
root@serverid: ~
  GNU nano 7.2                                  hello.sh
#! /bin/bash
#Prompt the user for their name
echo "Hello! What's your name?"
read name
#Greet the user with a personalized message
echo "Hello, $name! Welcome to the world of shell scripting!"
```

Step 4: Output

```
root@serverid:~# ./hello.sh
Hello! What's your name?
Mamatha
Hello, Mamatha! Welcome to the world of shell scripting!
root@serverid:~#
```

3. Shell Program to perform array sum

Step 1: Create an "**arraysum.sh"** script file using touch command

Step 2: Create a nano file to write the code

```
root@serverid:~# touch arraysum.sh
root@serverid:~# nano arraysum.sh
root@serverid:~# chmod +x arraysum.sh
```

Step 3: Write the Code in nano arraysum.sh script file

```
root@serverid: ~
  GNU nano 7.2                                arraysum.sh *
#!/bin/bash

# Prompt the user to input the array
echo "Enter numbers for the array"
read -a arr
sum=0
for (( i=0; i < ${#arr[*]}; i++ )); do
  if (( arr[i] > 0 )); then
     sum=$(expr $sum + ${arr[i]})
  fi
done
echo "Sum is: $sum"
```

Step 4: Output

```
root@serverid:~# ./arraysum.sh
Enter numbers for the array
2 4 -5 -8 9 12
Sum is: 27
root@serverid:~#
```

4. Shell Program to verify the number is palindrome number or not?

Step 1: Create an "**palindrome.sh"** script file using touch command

Step 2: Create a nano file to write the code

```
root@serverid:~# touch palindrome.sh
root@serverid:~# nano palindrome.sh
root@serverid:~# chmod +x palindrome.sh
```

Step 3: Write the Code in nano palindrome.sh script file

```
root@serverid: ~
  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
```

Step 4: Output

```
root@serverid:~# ./palindrome.sh
Enter the number:
121
121
The number is a palindrome!
root@serverid:~# ./palindrome.sh
Enter the number:
256
652
The number is not a palindrome number!
root@serverid:~#
```

5. Shell Program to perform Bubble sort?

Step 1: Create an "**bubblesort.sh"** script file using touch command

Step 2: Create a nano file to write the code

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

Step 3: Write the Code in nano bubblesort.sh script file

```
root@serverid: ~
 GNU nano 7.2                                  bubblesort.sh
#bubble sort in shell with static array
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[@]}
```

Step 4: Output

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


6. Shell Program to perform Pascals triangle?

Step 1: Create an "**pascaltriangle.sh"** script file using touch command

Step 2: Create a nano file to write the code

```
root@serverid:~# touch pascaltriangle.sh
root@serverid:~# nano pascaltriangle.sh
root@serverid:~# nano pascaltriangle.sh
root@serverid:~# chmod +x pascaltriangle.sh
```

Step 3: Write the Code in nano pascaltriangle.sh script file

```
root@serverid: ~
 GNU nano 7.2                                pascaltriangle.sh *
#Shell program to print Pascal triangle
pastri()
{
r=$1
c=1
for((i=0;i<r;i++))
{
for((s=1;s<r-i;s++))
{
echo -n "  "
}
for((j=0;j<=i;j++))
{
if [ $j -eq 0 -o $i -eq 0 ]
then
c=1;
else
c=$((c*(i-j+1)/j))
fi
echo -n $c "  "
}
echo
}
}
echo "enter the number of rows:"
read r
pastri $r
```

Step 4: Output

```
root@serverid:~# ./pascaltriangle.sh
enter the number of rows:
5
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
root@serverid:~#
```

7. Shell Program to perform Reverse of a Number

Step 1: Create an "**reversenums.sh"** script file using touch command

Step 2: Create a nano file to write the code

Step 3: Write the nano reversenums.sh script file

Step 4: Output