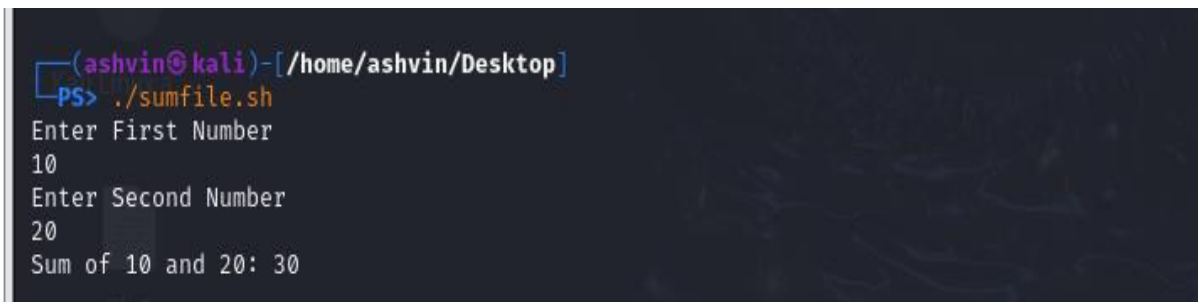


Practical 5: Write a shell Program to do the sum of two inputted numbers.

Code:

- Create a file called sumfile . Go to the nano sumfile and write the following script:

```
#!/bin/bash echo "Enter 1st number: "; read first_number  
echo "Enter 2nd number: "; read second_number  
sum=$((first_number + second_number)) echo "Sum of  
$first_number and $second_number: $sum";
```
- Now run the script ./sumfile.sh
- If permissions denied make a file executable by chmod +x sumfile.sh
- **OutPut:**



```
(ashvin@kali)-[/home/ashvin/Desktop]  
PS> ./sumfile.sh  
Enter First Number  
10  
Enter Second Number  
20  
Sum of 10 and 20: 30
```

Practical 6: An employee basic salary is input through keyboard where DA is 40% of basic salary and HRA is 20% of basic salary. write a program to calculate gross salary.

Script:

```
#!/bin/bash  
  
# Prompt the user to enter the basic salary  
  
echo "Enter the basic salary:";  
  
read basic_salary  
# Calculate DA as 40% of basic salary  
  
da=$((basic_salary * 40 / 100))
```

```

# Calculate HRA as 20% of basic salary

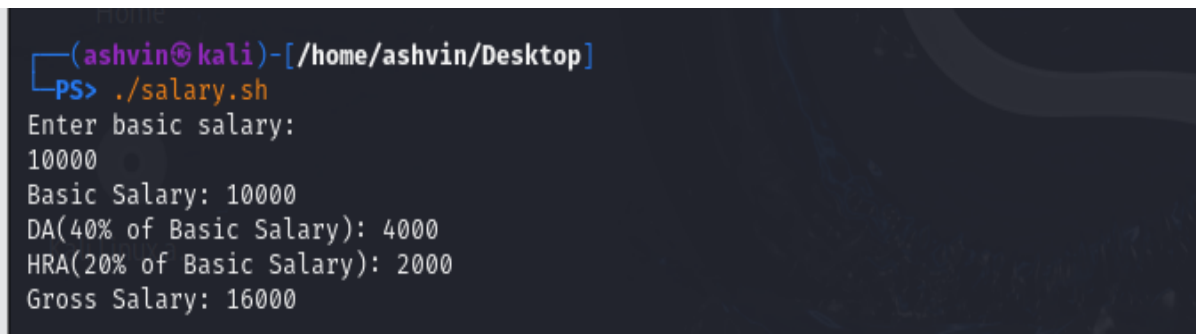
hra=$((basic_salary * 20 / 100))

# Calculate Gross Salary gross_salary=$((basic_salary +
da + hra))

# Display the results echo "Basic Salary:
$basic_salary"; echo "DA (40% of Basic Salary): $da";
echo "HRA (20% of Basic Salary): $hra"; echo "Gross
Salary: $gross_salary";

```

Output:



```

(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./salary.sh
Enter basic salary:
10000
Basic Salary: 10000
DA(40% of Basic Salary): 4000
HRA(20% of Basic Salary): 2000
Gross Salary: 16000

```

Practical 7: Write a shell Program to find maximum and minimum from three values.

Code:

```

#!/bin/bash

# Prompt the user to enter three numbers

echo "Enter the first number:"; read
num1

echo "Enter the second number:";
read num2

echo "Enter the third number:"; read
num3

# Find the maximum number
max=$num1

if [ $num2 -gt $max ]; then

```

```

        max=$num2
    fi

    if [ $num3 -gt $max ]; then
        max=$num3
    fi

# Find the minimum number min=$num1

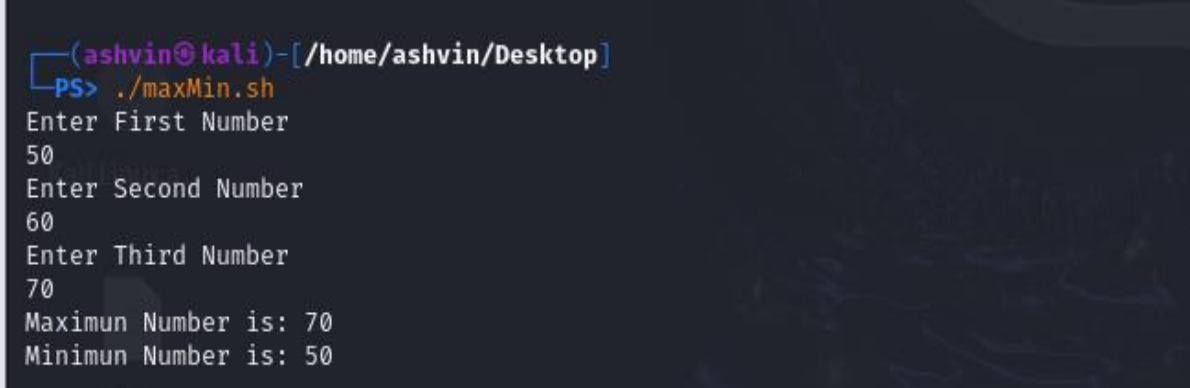
    if [ $num2 -lt $min ]; then
        min=$num2
    fi

    if [ $num3 -lt $min ]; then
        min=$num3
    fi

# Display the results echo "The maximum
value is: $max"; echo "The minimum value is:
$min";

```

Output:



```

(ashvin@kali) - [ /home/ashvin/Desktop ]
PS> ./maxMin.sh
Enter First Number
50
Enter Second Number
60
Enter Third Number
70
Maximun Number is: 70
Minimun Number is: 50

```

Practical 8: Write a shell script in Linux to find given number is even or odd?

Code:

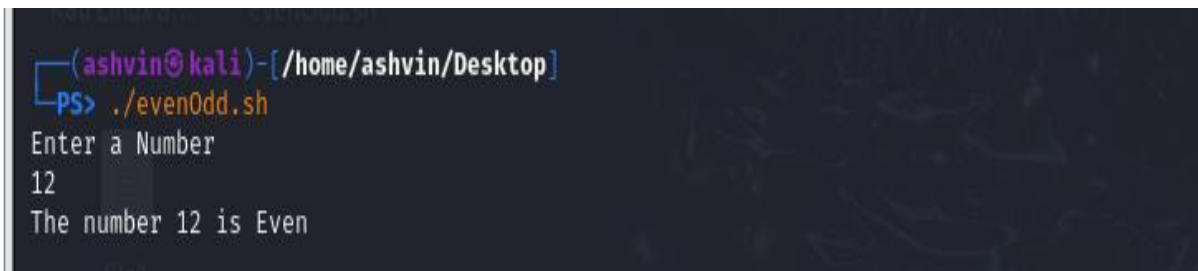
```
#!/bin/bash
```

```
# Prompt the user to enter a number

echo "Enter a number:"; read
number

# Check if the number is even or odd if [
$((number % 2)) -eq 0 ]; then echo
"The number $number is Even"; else
echo "The number $number is Odd." fi
```

Output:



```
(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./evenOdd.sh
Enter a Number
12
The number 12 is Even
```

Practical 9: Write a shell script in Linux Find Given year Is Leap year Or Not.

Code:

```
#!/bin/bash

# Prompt user to enter a year read -
p "Enter a year: "year

# Check if the year is divisible by 4, but not divisible by 100 unless divisible by
400

if [[ ( $((year % 4)) -eq 0 ) && ( $((year % 100)) -ne 0 || $((year %
400)) -eq 0 ) ]] then

echo "The year $year is a Leap year.";
else
echo "The year $year is not a Leap year.";
```

Output:

```
(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./leap.sh
Enter a year:
2000
The year 2000 is a Leap Year
```

Practical 10: Write a shell program to find out how many terminals have users logged in.

Code:

```
#!/bin/bash

logged_in_terminals=$(who | awk '{print $2}' | sort | uniq)

terminal_count=$(echo "$logged_in_terminals" | wc -l) echo "Number of
terminals with users logged in: $terminal_count";
```

Output:

```
(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./terminalCount.sh
Number of terminals with users logged in: 1
```

Practical 11: Write a script to count no of line, words and characters of a inputted file.

Code:

```
#!/bin/bash

echo &" Enter the filename:"; read
filename

if [[ -f "$filename" ]]; then #Tests if the file exists and is a regular file
(not a directory or other type of file).

echo "$filename is regular file (not a directory or other type of file)"

wc "$filename"; else

echo" File does not

exist."; fi
```

Output:

```
(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./f1.sh
Enter the filename:
evenOdd.sh
evenOdd.sh is a regular file (not a directory or other type of file)
 9 28 156 evenOdd.sh
```

Practical 12: Write a shell Program to find length of a given string.

Code: #!/bin/bash echo "Enter a string:"; read
input_string # Find the length of the string
length=\${#input_string} echo "Length of the
string is: \$length";

Output:

```
(ashvin@kali)-[/home/ashvin/Desktop]
PS> ./find.sh
Enter a string:
Hello World
Length of the string is: 11
```

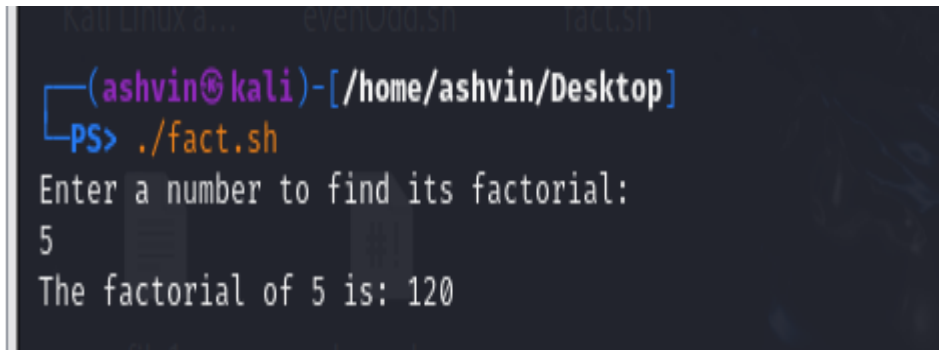
Practical 13: Write a shell Program to find out the factorial of input.

Code: #!/bin/bash

```
echo "Enter a number to find its factorial:";
read num
if [ $num -lt 0 ]; then
echo "Factorial is not defined for negative numbers."; exit 1
fi
factorial=1 i=1
while [ $i -le $num ]
```

```
factorial=$((factorial * i)) && i=$((i + 1))  
echo "The factorial of $num is: $factorial"; echo "The  
factorial of $num is: $factorial";
```

Output:

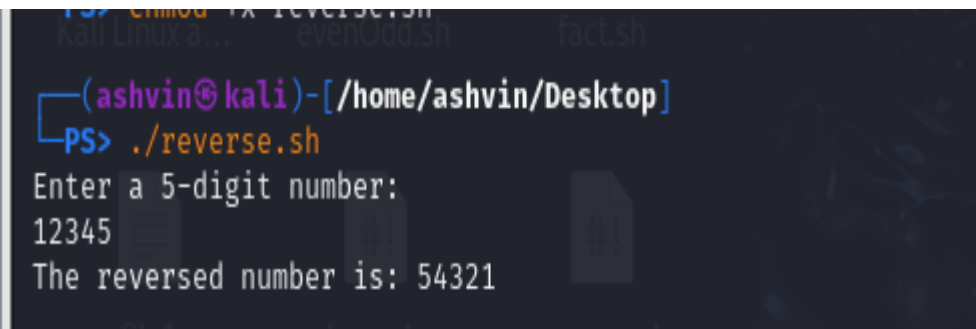
A terminal window with a dark background. The prompt is (ashvin@kali)-[/home/ashvin/Desktop]. The user enters ./fact.sh. The script prompts "Enter a number to find its factorial:" and the user enters 5. The script outputs "The factorial of 5 is: 120".

```
(ashvin@kali)-[/home/ashvin/Desktop]  
PS> ./fact.sh  
Enter a number to find its factorial:  
5  
The factorial of 5 is: 120
```

Practical 14: Write a shell program to reverse a given 5 digit number.

```
#!/bin/bash  
echo "Enter a 5-digit number:"; read  
num  
reversed=$(echo $num | rev) echo "The  
reversed number is: $reversed";
```

Output

A terminal window with a dark background. The prompt is (ashvin@kali)-[/home/ashvin/Desktop]. The user enters ./reverse.sh. The script prompts "Enter a 5-digit number:" and the user enters 12345. The script outputs "The reversed number is: 54321".

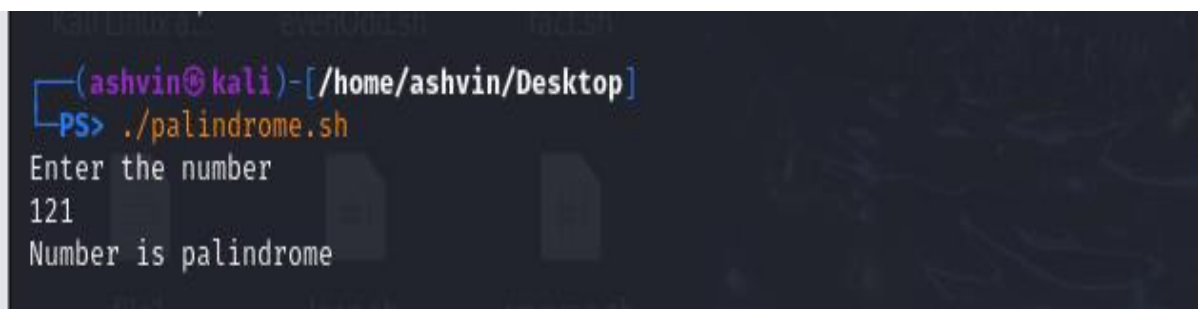
```
(ashvin@kali)-[/home/ashvin/Desktop]  
PS> ./reverse.sh  
Enter a 5-digit number:  
12345  
The reversed number is: 54321
```

Practical 15: Write a shell program to check inputted number is palindrome or not?.

```
echo "Enter the number"; read  
n  
function pal
```

```
{  
number=$n reverse=0  
while [ $n -gt 0 ] do a=`expr $n % 10`  
`n=`expr $n / 10 ` reverse=`expr  
$reverse \* 10 + $a` done echo  
$reverse if [ $number -eq $reverse ]  
then  
echo "Number is palindrome"; else  
echo "Number is not palindrome"; fi  
}  
r=`pal $n` echo  
"$r";
```

Output



A terminal window with a dark background. The prompt is `(ashvin@kali)-[/home/ashvin/Desktop]`. The user enters `PS> ./palindrome.sh`. The script prompts "Enter the number" and the user enters "121". The script outputs "Number is palindrome".