EXPERIMENT:- 03

AIM:- UNIX / LINUX Shell Programming Basics on Bourne Shell, Bash Shell, C Shell.

1. Write a shell script to perform arithmetic operations using expr in Bourne Shell.

```
shankar@Shankar ~> touch expr_bourne.sh
shankar@Shankar ~> gedit expr_bourne.sh
```

```
# Reading the inputs
read -p "Enter first number: " a
read -p "Enter Second number: " b

# Performing Arithmetic Operations
echo "Sum: $(expr $a + $b)"
echo "Difference: $(expr $a - $b)"
echo "Product: $(expr $a \* $b)"
echo "Division: $(expr $a / $b)"
echo "Modulus: $(expr $a % $b)"
```

```
shankar@Shankar ~> sh expr_bourne.sh
Enter first number: 75
Enter Second number: 15
Sum: 90
Difference: 60
Product: 1125
Division: 5
Modulus: 0
```

2. Write a shell script to perform arithmetic operations without using expr in Bash Shell.

```
shankar@Shankar ~> touch arithmetic.sh
shankar@Shankar ~> gedit arithmetic.sh
```

```
# Added a shebang  to explicitly specify that the script
# should be executed using the Bash shell.

#!/bin/bash

# Reading the numbers
read -p "Enter first number: " a
read -p "Enter second number: " b

# Performing arithmetic operations without expr
echo "a + b = $(($a + $b))"
echo "a - b = $(($a - $b))"
echo "a * b = $(($a * $b))"
echo "a / b = $(($a / $b))"
echo "a % b = $(($a % $b))"
```

```
shankar@Shankar ~> bash arithmetic.sh
Enter first number: 23
Enter second number: 8
a + b = 31
a - b = 15
a * b = 184
a / b = 2
a % b = 7
```

3. Write a shell script program to find the Maximum three numbers.

```
shankar@Shankar ~> touch greater.sh
shankar@Shankar ~> gedit greater.sh
```

```
read -p "Enter first number: " num1
read -p "Enter second number: " num2
read -p "Enter third number: " num3

greatest=0

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]; then
    greatest=$num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]; then
    greatest=$num2
else
    greatest-$num3
fi

echo "The greatest number is: $greatest"
```

```
shankar@Shankar ~> sh greater.sh
Enter first number: 57
Enter second number: 89
Enter third number: 43
The greatest number is: 89
```

4. Write a script for comparison of strings value of username and password variables will be taken from standard input. If both values match, then the output will be “valid user,” otherwise the output will be “invalid user.”

```
shankar@Shankar ~> touch authentication.sh
shankar@Shankar ~> gedit authentication.sh
```

```
read -p "Enter UserName: " user_name
read -p "Enter Password: " pw

echo    # To get a new line

if [ "$user_name" = "Shankar" ] && [ "$pw" = "150602" ]; then
        echo "Valid User"
else
        echo "Invalid User "
fi
```

```
shankar@Shankar ~> sh authentication.sh
Enter UserName: Shankar
Enter Password: 150602

Valid User
```

5. Write a shell script to perform Arithmetic operation using CASE.

```
shankar@Shankar ~> touch case_arithmetic.sh
shankar@Shankar ~> gedit case_arithmetic.sh
```

```
read -p "Enter first number: " num1
read -p "Enter second number: " num2

echo "Choose an arithmetic operation:-"
echo "1. Addition(+)"
echo "2. Substraction(-)"
echo "3. Multiplication(*)"
echo "4. Division(/)"

read -p "Enter Choice: " choice

case "$choice" in
    1)
        echo "The result of addition is: $((num1 + num2))"
        ;;
    2)
        echo "The result of subtraction is: $((num1 - num2))"
        ;;
    3)
        echo "The result of multiplication is: $((num1 * num2))"
        ;;
    4)
        if [ $num2 -eq 0 ]; then
            echo "Error: Division by zero is not allowed"
        else
            echo "The result of division is: $((num1 / num2))"
        fi
        ;;
    *)
        echo "Invalid choice"
        ;;
esac
```

```
shankar@Shankar ~> sh case_arithmetic.sh
Enter first number: 89
Enter second number: 23
Choose an arithmetic operation:-
1. Addition(+)
2. Substraction(-)
3. Multiplication(*)
4. Division(/)
Enter Choice: 4
The result of division is: 3
```

6. Write a shell script to print the AP series 1,2,3…1000 using loop.

```
shankar@Shankar ~> touch ap.sh
shankar@Shankar ~> gedit ap.sh
```

```
read -p "Enter a number: " num

# Use a for loop to print the AP series
if [ $num -gt 0 ] && [ $num -lt 1001 ]; then
        for (( i = 1; i <= $num; i++ )); do
                echo $i
        done
fi
```

```
shankar@Shankar ~> bash ap.sh
Enter a number: 15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

7. Write a shell script to calculate the factorial value of a number.

```
shankar@Shankar ~> touch facto.sh
shankar@Shankar ~> gedit facto.sh
```

```
read -p "Enter a number: " num

f=1
for((i=1; i<=$num; i++)); do
    f=`expr $f \* $i`
done
echo "Factorial Value=" $f
```

```
shankar@Shankar ~> bash facto.sh
Enter a number: 7
Factorial Value= 5040
```

8. Write a shell program to generate Fibonacci series.

```
shankar@Shankar ~> touch fibonaci.sh
shankar@Shankar ~> gedit fibonaci.sh
```

```
read -p "Enter the number to generate Fibonacci: " num

term1=0
term2=1
count=2

echo "Fibonacci series up to $num terms:-"
# Print the first 2 number
echo -n "$term1 $term2 "

until [ $count -ge $num ]; do
        next_term=$((term1 + term2))
        # print the next number
        echo -n "$next_term "
        # update values
        term1=$term2
        term2=$next_term
        # increment the count
        count=$((count + 1))
done
#printing new line character to end the output
echo
```

```
shankar@Shankar ~> sh fibonaci.sh
Enter the number to generate Fibonacci: 12
Fibonacci series up to 12 terms:-
0 1 1 2 3 5 8 13 21 34 55 89
```

9. Write shell program to print sum of digit of a given number and its reverse.

```
shankar@Shankar ~> touch sum_of_digit.sh
shankar@Shankar ~> gedit sum_of_digit.sh
```

```
# Read a number from the user
read -p "Enter a number: " number

# Initialize variables
original_number=$number
sum_of_digits=0
reversed_number=0

# Calculate the sum of digits
while [ $number -gt 0 ]; do
    digit=$((number % 10))
    sum_of_digits=$((sum_of_digits + digit))
    number=$((number / 10))
done

# Calculate the reverse of the number
number=$original_number
while [ $number -gt 0 ]; do
    digit=$((number % 10))
    reversed_number=$((reversed_number * 10 + digit))
    number=$((number / 10))
done

# Display the results
echo "Sum of digits of $original_number is:- $sum_of_digits"
echo "Reverse of $original_number is:- $reversed_number"
```

```
shankar@Shankar ~> sh sum_of_digit.sh
Enter a number: 2101020758
Sum of digits of 2101020758 is:- 26
Reverse of 2101020758 is:- 8570201012
```

10. Write shell program to print the following series

```
   *
  ***
 *****
*******
```

```
shankar@Shankar ~> touch asterik_series.sh
shankar@Shankar ~> gedit asterik_series.sh
```

```
# Loop to print the pattern
for ((i = 1; i <= 4; i++)); do
    # Print spaces for indentation
    for ((j = i; j < 4; j++)); do
        echo -n " "
    done

    # Print asterisks
    for ((k = 1; k <= 2 * i - 1; k++)); do
        echo -n "*"
    done

    # Move to the next line
    echo
done
```

```
shankar@Shankar ~> bash asterik_series.sh
   *
  ***
 *****
*******
```