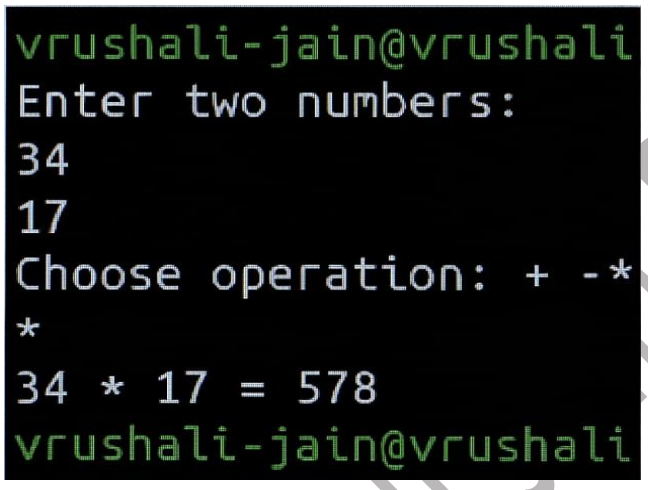
ASSIGNMENT 3

To create shell scripts for the following questions

1. To find Largest of Three Numbers

```
#!/bin/bash
read -p "Enter first number: " a
read -p "Enter second number: "  b
read -p "Enter third number: " c

if (( a >= b && a >= c )); then
  echo "$a is the largest."
elif (( b >= a && b >= c )); then
   echo "$b is the largest."
else
   echo "$c is the largest."
Fi
```



2. To find a year is leap year or not.

```
#!/bin/bash
read -p "Enter a year: " year

if (( (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) )); then
   echo "$year is a leap year."
else
   echo "$year is not a leap year."
Fi
```

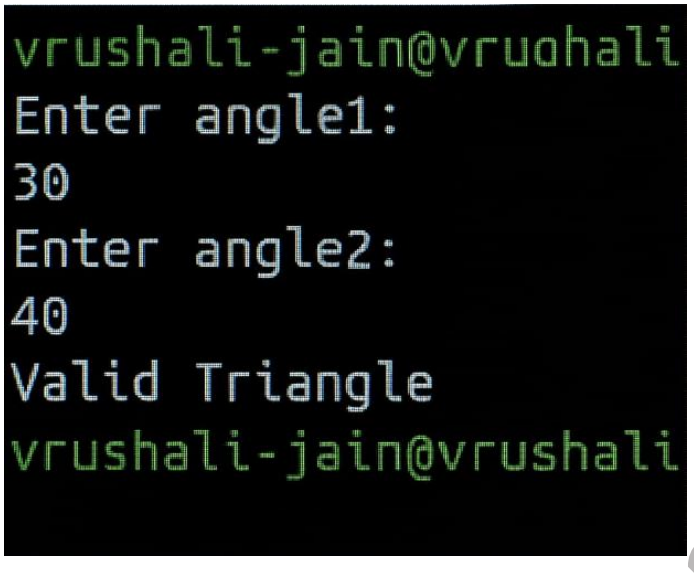
3. To input angles of a triangle and find out whether it is valid triangle or not

```
#!/bin/bash
read -p "Enter angle1: " a
read -p "Enter angle2: " b
read -p "Enter angle3: " c

sum=$((a + b + c))

if (( sum == 180 && a > 0 && b > 0 && c > 0 )); then
   echo "Valid Triangle"
```

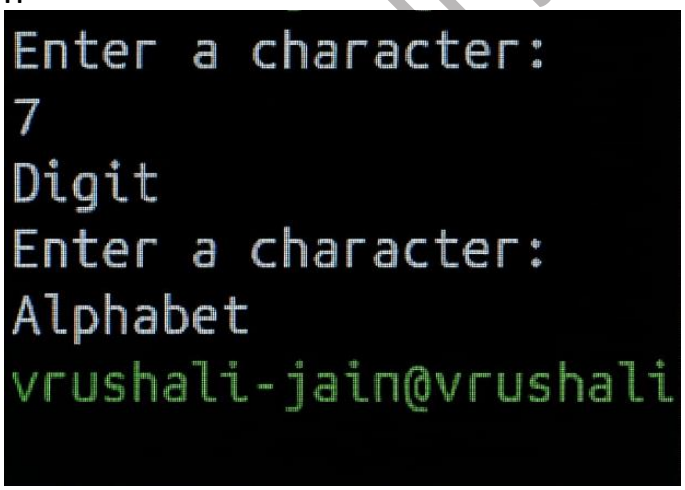
```
else
   echo "Invalid Triangle"
fi
```



4. To check whether a character is alphabet, digit or special character.

```
#!/bin/bash
read -p "Enter a character: " ch

if [[ $ch =~ [A-Za-z] ]]; then
   echo "Alphabet"
elif [[ $ch =~ [0-9] ]]; then
   echo "Digit"
else
   echo "Special Character"
Fi
```

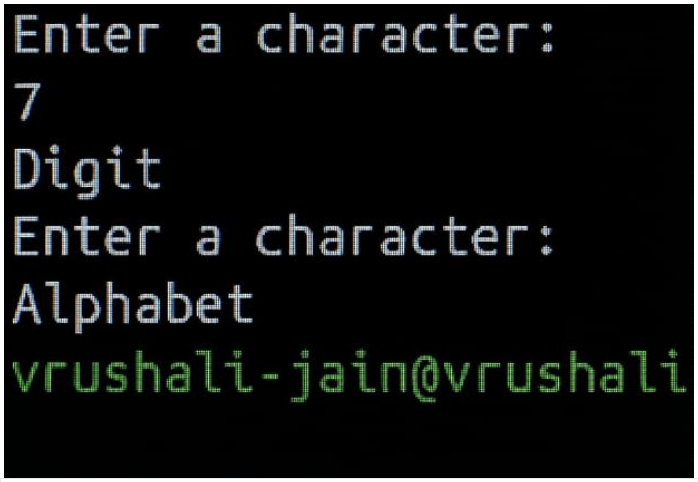


5. To calculate profit or loss

```
#!/bin/bash
read -p "Enter a character: " ch

if [[ $ch =~ [A-Za-z] ]]; then
   echo "Alphabet"
elif [[ $ch =~ [0-9] ]]; then
```

```
   echo "Digit"
else
   echo "Special Character"
Fi
```

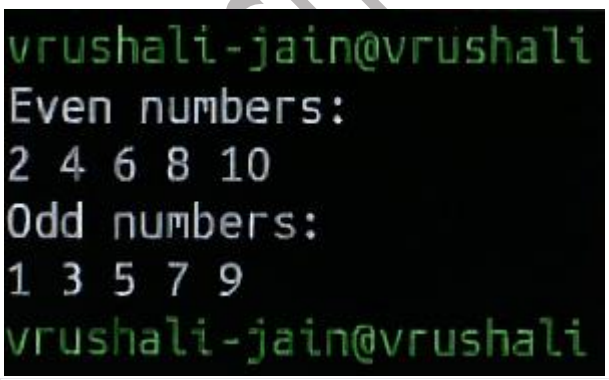


6. To print all even and odd number from 1 to 10

```
#!/bin/bash
echo "Even numbers:"
for i in {1..10}; do
   if (( i % 2 == 0 )); then
      echo -n "$i "
   fi
done

echo -e "\nOdd numbers:"
for i in {1..10}; do
   if (( i % 2 != 0 )); then
      echo -n "$i "
   fi
done
```

Echo



7.To print table of a given number

```
#!/bin/bash
read -p "Enter a number: " n
for i in {1..10}; do
   echo "$n x $i = $((n * i))"
Done
```

```
Enter a number:
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 1 = 40
vrushali-jain@vrushali
```

8.To find factorial of a given integer

```
#!/bin/bash
read -p "Enter a number: " num
fact=1

for ((i=1; i<=num; i++)); do
   fact=$((fact * i))
done

echo "Factorial of $num is $fact"
```

```
vrushali-jain@vrushali-jain:
Enter a number: 7
Factorial of 7 is 5040
vrushali-jain@vrushali-jain:

Factorial of 7 is 5040
```

9.To print sum of all even numbers from 1 to 10.

```
#!/bin/bash
sum=0
for i in {1..10}; do
   if (( i % 2 == 0 )); then
      sum=$((sum + i))
   fi
done
echo "Sum of even numbers from 1 to 10 is $sum"
```

```
vrushali-jain@vrushali-jain
Sum of even numbers from
1 to 10 is 30
vrushali-jain@vrushali-jain
```

10.To print sum of digit of any number.

```
#!/bin/bash
read -p "Enter a number: " num
sum=0

while (( num > 0 )); do
   digit=$((num % 10))
   sum=$((sum + digit))
   num=$((num / 10))

done
echo "Sum of digits is $sum"
```

```
Enter a number: 4372
Sum of digits is 16
vrushali-jain@vrushali:
```

11.To make a basic calculator which performs addition, subtraction, Multiplication, division

```
#!/bin/bash
echo "Enter two numbers:"
read a
read b

echo "Choose operation: + - * /"
read op

case $op in
   +) echo "$a + $b = $((a + b))" ;;
   -) echo "$a - $b = $((a - b))" ;;
   \*) echo "$a * $b = $((a * b))" ;;
   /) echo "$a / $b = $((a / b))" ;;
   *) echo "Invalid operation" ;;
Esac
```

```
vrushali-jain@vrushali
Enter two numbers:
34
17
Choose operation: + -*
*
34 * 17 = 578
vrushali-jain@vrushali
```

12.To print days of a week.

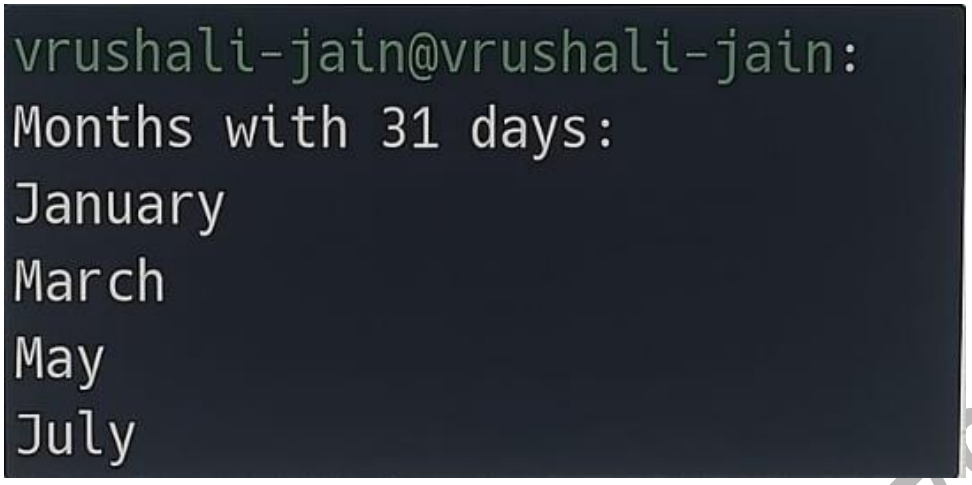
```
#!/bin/bash
days=("Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday" "Saturday")

for day in "${days[@]}"; do
   echo "$day"
Done
```

```
vrushali-jain@vrushali:
Sunday
Monday
Tuesday
Wednesday
Thursday
Saturday
vrushali-jain@vrushali:
```

13.To print starting 4 months having 31 days.

```
#!/bin/bash
months=("January" "March" "May" "July")
echo "Months with 31 days:"
for month in "${months[@]}"; do
   echo "$month"
Done
```



14.Using functions,

a. To find given number is Amstrong number or not

```
#!/bin/bash
is_armstrong() {
   num=$1
   sum=0
   temp=$num

   while (( temp > 0 )); do
      digit=$((temp % 10))
      sum=$((sum + digit * digit * digit))
      temp=$((temp / 10))
   done

   if (( sum == num )); then
      echo "$num is an Armstrong number."
   else
      echo "$num is not an Armstrong number."
   fi
}

read -p "Enter a number: " n
is_armstrong $n
```

b. To find whether a number is palindrome or not

```
#!/bin/bash
is_palindrome() {
   num=$1
   rev=0
   temp=$num

   while (( temp > 0 )); do
      digit=$((temp % 10))
      rev=$((rev * 10 + digit))
      temp=$((temp / 10))
   done
```

```
   if (( rev == num )); then
      echo "$num is a palindrome."
   else
      echo "$num is not a palindrome."
   fi
}

read -p "Enter a number: " n
is_palindrome $n
```

c. To print Fibonacci series upto n terms

```
#!/bin/bash
fibonacci() {
   n=$1
   a=0
   b=1
   echo "Fibonacci series up to $n terms:"
   for (( i=0; i<n; i++ )); do
      echo -n "$a "
      fn=$((a + b))
      a=$b
      b=$fn
   done
   echo
}

read -p "Enter number of terms: " n
fibonacci $n
```

d. To find given number is prime or composite

```
#!/bin/bash
is_prime() {
   num=$1
   if (( num <= 1 )); then
      echo "$num is neither prime nor composite."
      return
   fi

   for (( i=2; i*i<=num; i++ )); do
      if (( num % i == 0 )); then
         echo "$num is composite."
         return
      fi
   done
   echo "$num is prime."
}

read -p "Enter a number: " n
is_prime $n
```

e. To convert a given decimal number to binary equivalent

```
#!/bin/bash
decimal_to_binary() {
   num=$1
   binary=""
   while (( num > 0 )); do
      binary=$((num % 2))$binary
      num=$((num / 2))
   done
   echo "Binary: $binary"
}

read -p "Enter decimal number: " n
decimal_to_binary $n
```