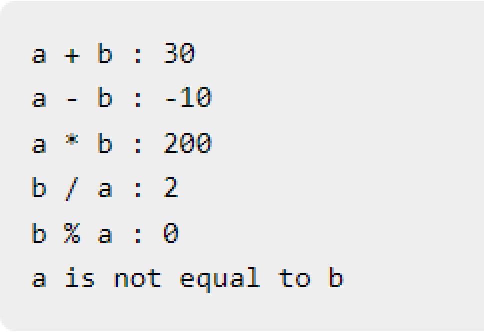
POS
MODULE 5

7. **Write a shell script for performing the arithmetic operations.**

Code

```
#!/bin/sh

a=10
b=20

val=`expr $a + $b`
echo "a + b : $val"

val=`expr $a - $b`
echo "a - b : $val"

val=`expr $a \* $b`
echo "a * b : $val"

val=`expr $b / $a`
echo "b / a : $val"

val=`expr $b % $a`
echo "b % a : $val"

if [ $a == $b ]
then
  echo "a is equal to b"
fi

if [ $a != $b ]
then
  echo "a is not equal to b"
fi
```

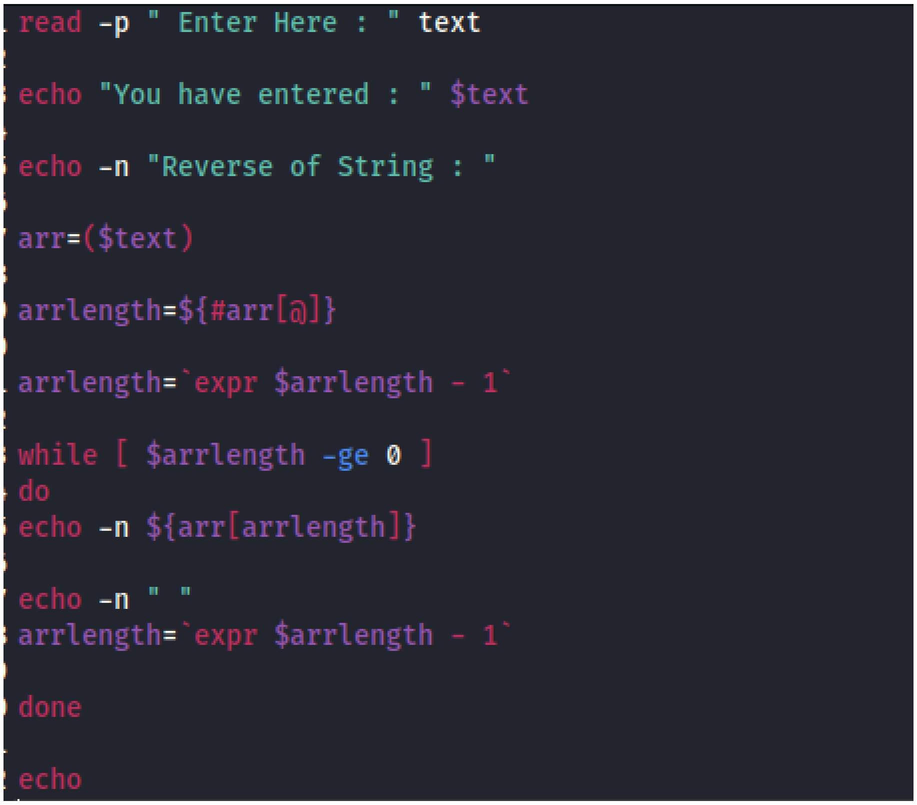
Output



```
a + b : 30
a - b : -10
a * b : 200
b / a : 2
b % a : 0
a is not equal to b
```

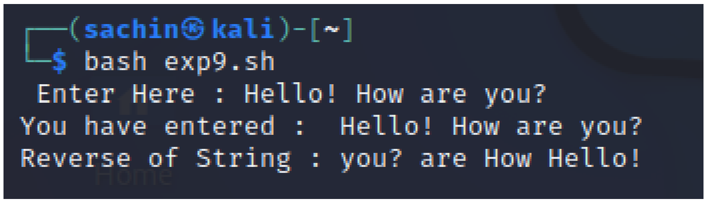
POS
MODULE 5

8. a) Write a shell script to print the reverse sting on the screen



```
read -p " Enter Here : " text

echo "You have entered : " $text

echo -n "Reverse of String : "

arr=($text)

arrlength=${#arr[@]}

arrlength=`expr $arrlength - 1`

while [ $arrlength -ge 0 ]
do
echo -n ${arr[arrlength]}

echo -n " "
arrlength=`expr $arrlength - 1`

done

echo
```

Output



```
┌──(sachin㉿kali)-[~]
└─$ bash exp9.sh
 Enter Here : Hello! How are you?
You have entered :  Hello! How are you?
Reverse of String : you? are How Hello!
```

b) Explain about the features of C shell

Features of C shell are as follows:-

I. Wildcard substitution in file names (pattern-matching): Carries out commands on a group of files by specifying a pattern to match, rather than specifying an actual file name.

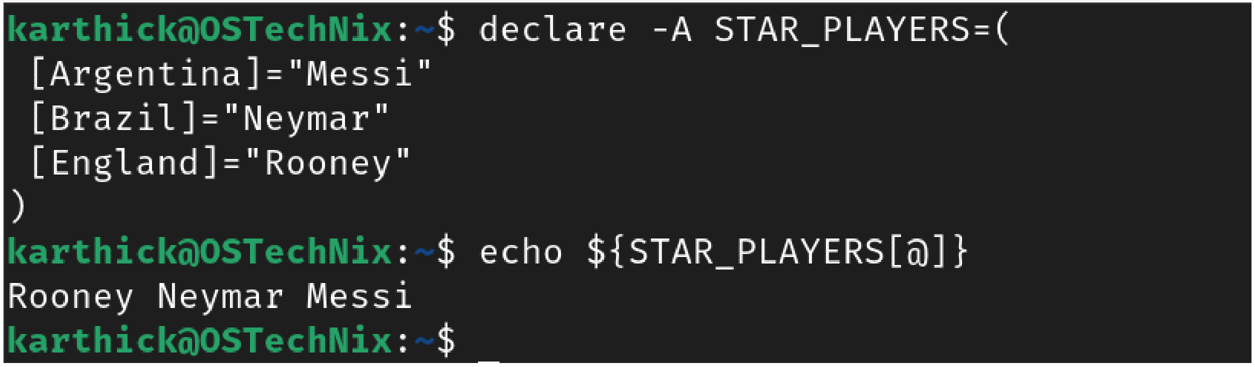
II. Background processing: Sets up lengthy tasks to run in the background, freeing the terminal for concurrent interactive processing.

III. Command aliasing: Gives an alias name to a command or phrase. When the shell encounters an alias on the command line or in a shell script, it substitutes the text to which the alias refers.

IV. Command history: Records the commands you enter in a history file. You can use this file to easily access, modify, and reissue any listed command.

V. File name substitution: Automatically produces a list of file names on a command line using pattern-matching characters.

```
$ echo ${STAR_PLAYERS[@]}

$ echo ${STAR_PLAYERS[*]}
```



```
karthick@OSTechNix:~$ declare -A STAR_PLAYERS=(
 [Argentina]="Messi"
 [Brazil]="Neymar"
 [England]="Rooney"
)
karthick@OSTechNix:~$ echo ${STAR_PLAYERS[@]}
Rooney Neymar Messi
karthick@OSTechNix:~$ _
```

10. **Write a shell script for finding the GCD between the given numbers.**

Code

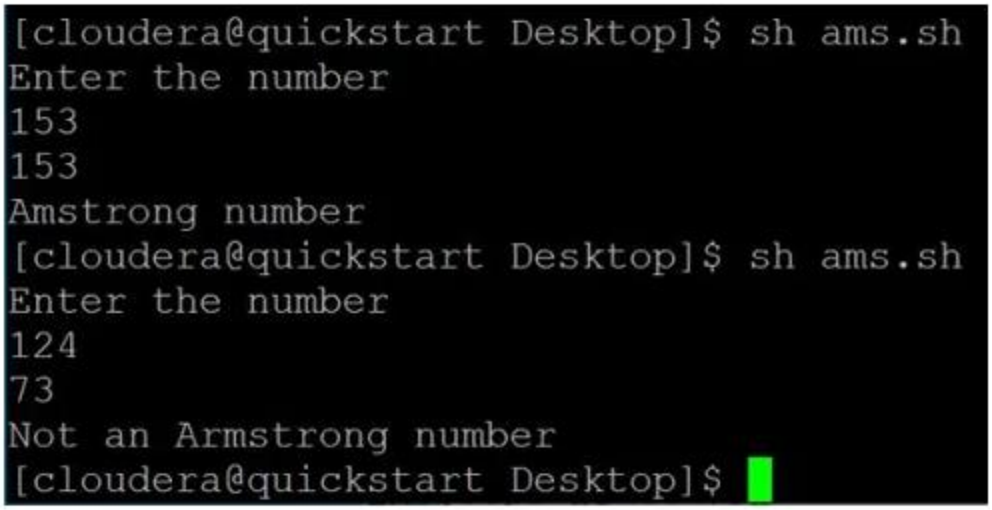
```
echo Enter two numbers with space in between
read a b
m=$a
if [ $b -lt $m ]
then
m=$b
fi
while [ $m -ne 0 ]
do
x=`expr $a % $m`
y=`expr $b % $m`
if [ $x -eq 0 -a $y -eq 0 ]
then
echo gcd of $a and $b is $m
break
fi
m=`expr $m - 1`
done
```

POS
MODULE 5

11. Write a shell script to find whether the given no is amstrong no or not.

```
echo "Enter the number"
read n
function ams
{
t=$n
s=0
b=0
c=10
while [ $n -gt $b ]
do
r=$((n % c))
i=$((r * r * r))
s=$((s + i))
n=$((n / c))
done
echo $s
if [ $s == $t ]
then
echo "Amstrong number"
else
echo "Not an Armstrong number"
fi
}
result=`ams $n`
echo "$result"
```

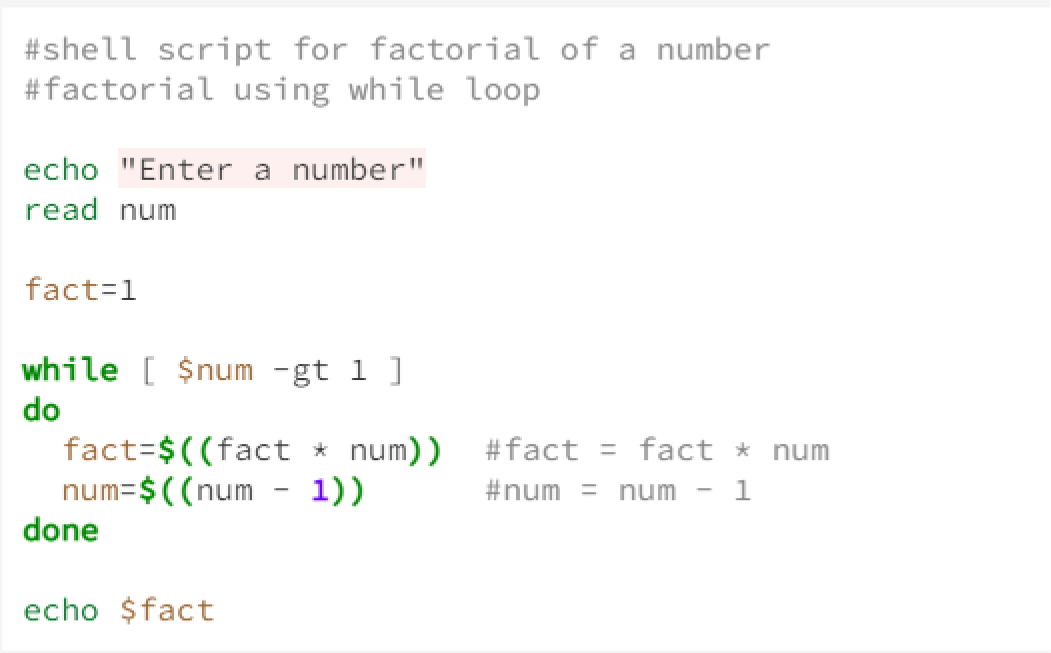
Output



```
[cloudera@quickstart Desktop]$ sh ams.sh
Enter the number
153
153
Amstrong number
[cloudera@quickstart Desktop]$ sh ams.sh
Enter the number
124
73
Not an Armstrong number
[cloudera@quickstart Desktop]$
```

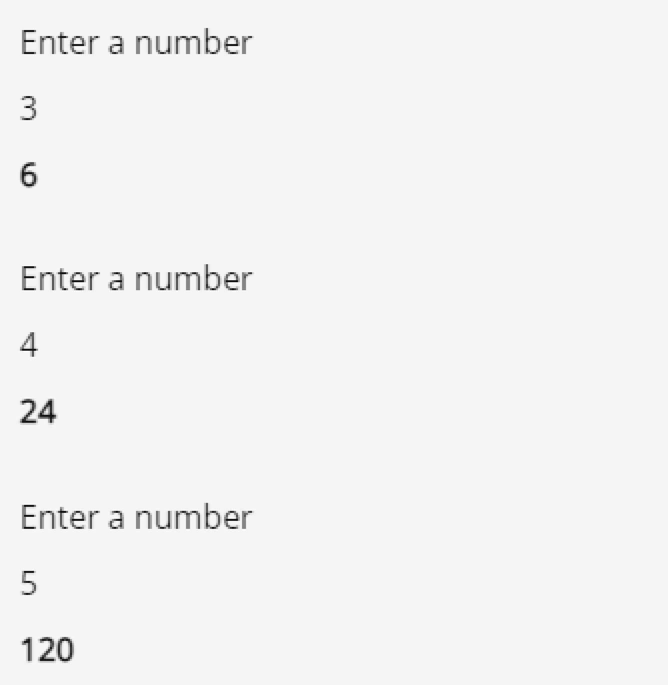
POS
MODULE 5

12. Write a shell script to find the factorial of given number.



```
#shell script for factorial of a number
#factorial using while loop

echo "Enter a number"
read num

fact=1

while [ $num -gt 1 ]
do
  fact=$((fact * num))  #fact = fact * num
  num=$((num - 1))      #num = num - 1
done

echo $fact
```

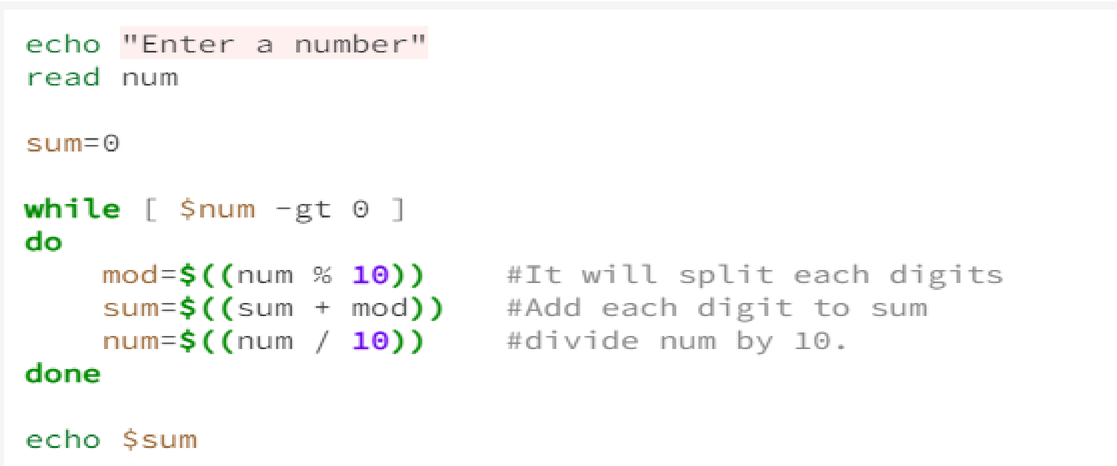
Output



```
Enter a number

3

6


Enter a number

4

24


Enter a number

5

120
```

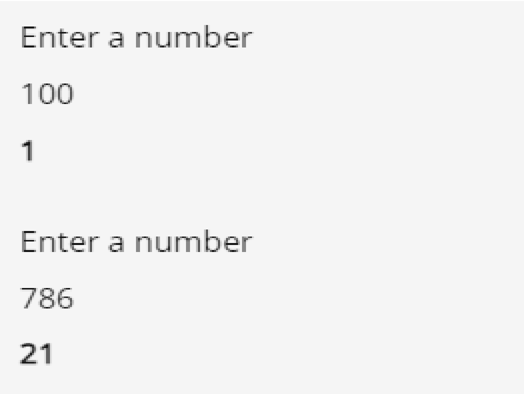
POS
MODULE 5

13. a) Write a shell script to find the sum of digits of given number.



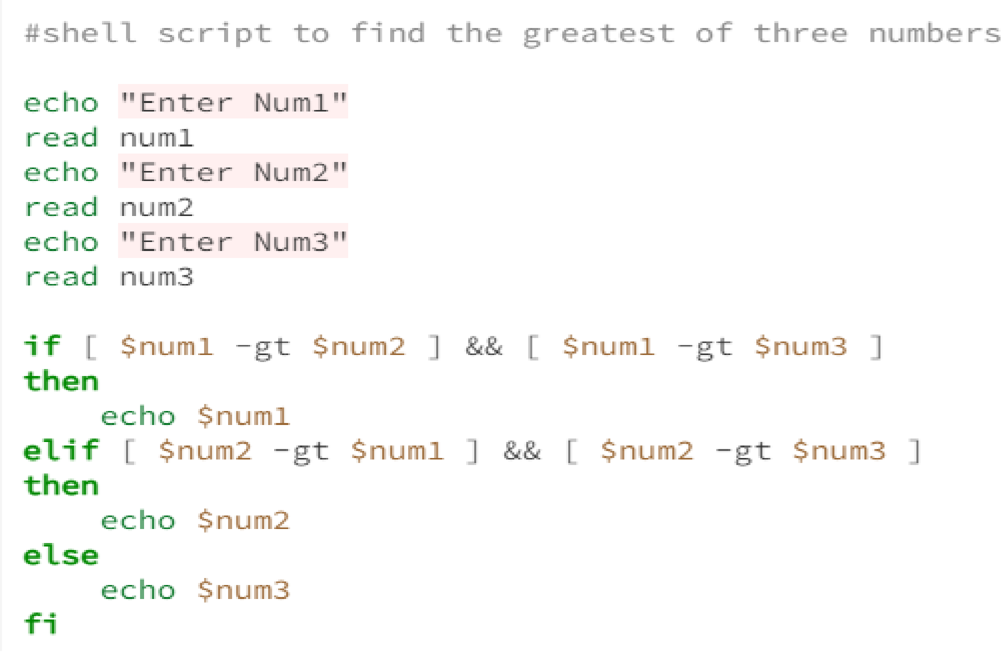
```
echo "Enter a number"
read num

sum=0

while [ $num -gt 0 ]
do
    mod=$((num % 10))    #It will split each digits
    sum=$((sum + mod))   #Add each digit to sum
    num=$((num / 10))    #divide num by 10.
done

echo $sum
```

Output



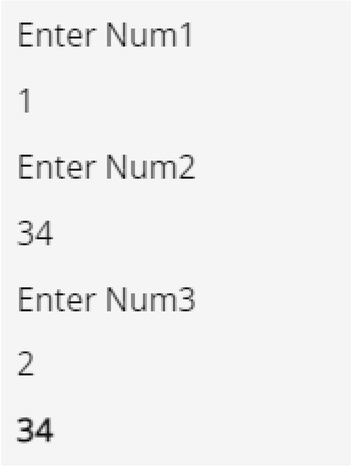
```
Enter a number
100
1

Enter a number
786
21
```

14. Write a shell script to find greatest of three given number



```
#shell script to find the greatest of three numbers

echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo $num2
else
    echo $num3
fi
```

Output



```
Enter Num1

1

Enter Num2

34

Enter Num3

2

34
```

15. **You need to print a given Number say 10572, in reverse order using a Shell script such that the input is provided using command Line Argument only. If the input data is not provided as Command Line Argument, it should throw and error and should suggest, how to use the script. Write the script but before that tell me the algorithm that needs to be implemented here.**

Solution:

create a file called ‘**numbers.sh**‘ and add the following given code it.

```
#!/bin/bash
if [ $# -ne 1 ]
then
    echo "Usage: $0   number"
    echo "       I will find reverse of given number"
    echo "       For eg. $0 0123, I will print 3210"
    exit 1
fi

n=$1
rev=0
sd=0
while [ $n -gt 0 ]
do
    sd=`expr $n % 10`
    rev=`expr $rev \* 10  + $sd`
    n=`expr $n / 10`
done
    echo  "Reverse number is $rev"
```

OUTPUT

```
$ ./numbers.sh 10572

Reverse number is 27501
```