

Shell script program questions.

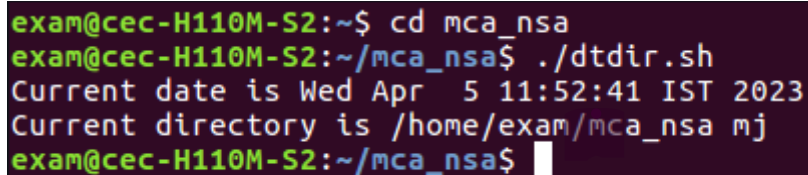
1. Write a script to show current date,time and current directory.

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

```
echo "Current date is `date`"
```

```
echo "Current directory is `pwd`"
```

Output:



```
exam@cec-H110M-S2:~$ cd mca_nsa
exam@cec-H110M-S2:~/mca_nsa$ ./dtdir.sh
Current date is Wed Apr  5 11:52:41 IST 2023
Current directory is /home/exam/mca_nsa mj
exam@cec-H110M-S2:~/mca_nsa$
```

2. Write shell script to find reverse of a number.

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

```
echo enter the no:
```

```
read n
```

```
num=0
```

```
while [ $n -gt 0 ]
```

```
do
```

```
num=$(expr $num \* 10)
```

```
k=$(expr $n % 10)
```

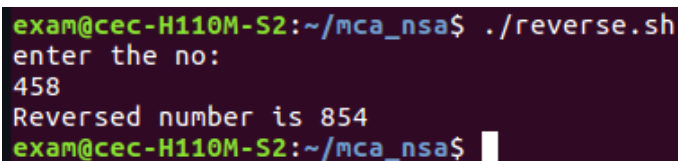
```
num=$(expr $num + $k)
```

```
n=$(expr $n / 10)
```

```
done
```

```
echo Reversed number is $num
```

Output:



```
exam@cec-H110M-S2:~/mca_nsa$ ./reverse.sh
enter the no:
458
Reversed number is 854
exam@cec-H110M-S2:~/mca_nsa$
```

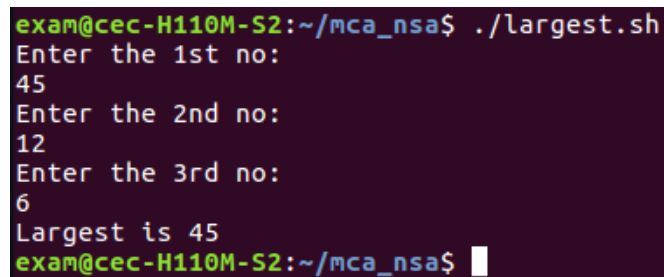
3. Write a script to find largest among three numbers.

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

```
echo Enter the 1st no:
```

```
read m
echo Enter the 2nd no:
read n
echo Enter the 3rd no:
read o
if [ $m -gt $n ] && [ $m -gt $o ]
then
echo Largest is $m
elif [ $n -gt $m ] && [ $n -gt $o ]
then
echo Largest is $n
else
echo Largest is $o
fi
```

Output:

A terminal window with a dark purple background. The prompt is 'exam@cec-H110M-S2:~/mca_nsa\$'. The user enters './largest.sh'. The script prompts for three numbers: 'Enter the 1st no:' (45), 'Enter the 2nd no:' (12), and 'Enter the 3rd no:' (6). It then outputs 'Largest is 45'. The prompt returns to 'exam@cec-H110M-S2:~/mca_nsa\$' with a cursor.

```
exam@cec-H110M-S2:~/mca_nsa$ ./largest.sh
Enter the 1st no:
45
Enter the 2nd no:
12
Enter the 3rd no:
6
Largest is 45
exam@cec-H110M-S2:~/mca_nsa$
```

4. Write a script to check whether a number is armstrong or not.

```
#!/bin/bash
echo "Enter a number: "
read c
x=$c
sum=0
r=0
n=0
while [ $x -gt 0 ]
do
r=`expr $x % 10`
```

```

n=`expr $r \* $r \* $r`
sum=`expr $sum + $n`
x=`expr $x / 10`
done
if [ $sum -eq $c ]
then
echo "It is an Armstrong Number."
else
echo "It is not an Armstrong Number."
fi

```

Output:

```

exam@cec-H110M-S2:~/mca_nsa$ ./armstrong.sh
Enter a number:
153
It is an Armstrong Number.
exam@cec-H110M-S2:~/mca_nsa$ ./armstrong.sh
Enter a number:
412
It is not an Armstrong Number.
exam@cec-H110M-S2:~/mca_nsa$

```

5. Write a script to check password and login.

```

#!/bin/bash
read -p 'Username: ' user
read -sp 'Password: ' pass
if (( $user == "Admin" && $pass == "admin123" ))
then
    echo -e "\nWelcome! You are Sucessfull login\n"
else
    echo -e "\nUnsuccessful login\n"
fi

```

Output:

```

exam@cec-H110M-S2:~/mca_nsa$ ./checkpwd.sh
Username: Admin
Password:
Welcome! You are Sucessfull login

exam@cec-H110M-S2:~/mca_nsa$ ./checkpwd.sh
Username: admin
Password: ./checkpwd.sh: line 6: ((: pass: expression recursion level exceeded (error token is "pass")
Unsuccessful login

exam@cec-H110M-S2:~/mca_nsa$

```

6. Write a script to count the prime numbers in specific range

```
#!/bin/bash
echo "Enter a lower limit"
read i
echo "Enter a upper limit"
read limit
echo "prime numbers upto $limit are :"
while [ $i -le $limit ]
do
    flag=1
    j=2
    while [ $j -lt $i ]
    do
        rem=$(( $i % $j ))
        if [ $rem -eq 0 ]
        then
            flag=0
            break
        fi
        j=$(( $j+1 ))
    done
    if [ $flag -eq 1 ]
    then
        echo "$i"
    fi
    i=$(( $i+1 ))
done
```

Output:

```
exam@CC2-33:~/test$ chmod +x prime.sh
exam@CC2-33:~/test$ ./prime.sh
Enter a limit
20
prime numbers upto 20 are :
1
2
3
5
7
11
13
17
19
exam@CC2-33:~/test$
```

7. Write a script to convert the contents of agiven file from uppercase to lowercase and also count the number of lines,words and characters of the resultant file. Also display the resultant file in descending order.

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

```
getFile()
```

```
{
```

```
    # Reading txtFileName to convert it's content
```

```
    echo -n "Enter File Name:"
```

```
    read filename
```

```
    # Checking if file exist
```

```
    if [ ! -f $filename ]; then
```

```
        echo "File Name $filename does not exists."
```

```
        exit 1
```

```
    fi
```

```
}
```

```
echo "1. Uppercase to Lowercase "
```

```
echo "2. Count the number of characters,words,lines"
```

```
echo "3. Exit "
```

```
echo -n "Enter your Choice(1-3):"
```

```
read ch
```

```
case "$ch" in
```

```
    1)
```

```
# Function Call to get File
getFile
# Converting to lower case if user choose 1
echo "Converting Upper-case to Lower-Case "
tr '[A-Z]' '[a-z]' <$filename
;;

2)
echo Enter the filename
read file
c=`cat $file | wc -c`
w=`cat $file | wc -w`
l=`grep -c "." $file`
echo Number of characters in $file is $c
echo Number of words in $file is $w
echo Number of lines in $file is $l
;;

*) # exiting for all other cases
echo "Exiting..."
exit
;;
esac
```

Output:

```
Enter your Choice(1-3):^Cexam@cec-H110M-S2:~/mca_nsa$ ./fileop.sh
1. Uppercase to Lowercase
2. Count the number of characters,words,lines
3. Exit
Enter your Choice(1-3):1
Enter File Name:sample.txt
Converting Upper-case to Lower-Case
welcome to linux operating system.
this is second line.
this is third line.
exam@cec-H110M-S2:~/mca_nsa$ ./fileop.sh
1. Uppercase to Lowercase
2. Count the number of characters,words,lines
3. Exit
Enter your Choice(1-3):2
Enter the filename
sample.txt
Number of characters in sample.txt is 76
Number of words in sample.txt is 13
Number of lines in sample.txt is 3
exam@cec-H110M-S2:~/mca_nsa$ ./fileop.sh
1. Uppercase to Lowercase
2. Count the number of characters,words,lines
3. Exit
Enter your Choice(1-3):3
Exiting...
exam@cec-H110M-S2:~/mca_nsa$ █
```

8. Write a script to perform following basic math operation as:

Addition, subtraction, multiplication, division

```
#!/bin/sh
```

```
echo "Enter the two numbers to perform arithmetic operations "
```

```
read a b
```

```
val=`expr $a + $b`
```

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

```
val=`expr $a - $b`
```

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

```
val=`expr $a \* $b`
```

```
echo "a * b : $val"
```

```
if [ $a -gt $b ]
```

```
then
```

```
    val=`expr $a / $b`
```

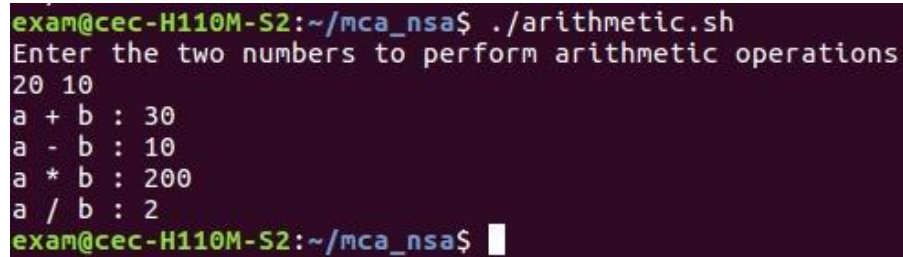
```
    echo "a / b : $val"
```

```
else
```

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

fi

Output:



```
exam@cec-H110M-S2:~/mca_nsa$ ./arithmetic.sh  
Enter the two numbers to perform arithmetic operations  
20 10  
a + b : 30  
a - b : 10  
a * b : 200  
a / b : 2  
exam@cec-H110M-S2:~/mca_nsa$
```

9. Read 3 marks of a student and find the average. Display the grade of the student based on the average. (if..then..elif..fi)

S >= 90%

A < 90%, but >= 80%

B < 80%, but >= 60%

P < 80%, but >= 40%

F < 40%

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

```
read -p "Enter three marks out of 100 each : " m1 m2 m3
```

```
s=$((m1+m2+m3))
```

```
avg=$((s / 3|bc))
```

```
echo -e "Average: $avg"
```

```
if [ $avg -ge 90 ]
```

```
then
```

```
    echo "Grade: S"
```

```
elif [[ $avg -lt 90 && $avg -ge 80 ]]
```

```
then
```

```
    echo "Grade: A"
```

```
elif [[ $avg -lt 80 && $avg -ge 60 ]]
```



```

then
    echo "Grade: B"
elif [[ $avg -lt 80 && $avg -ge 40 ]]
then
    echo "Grade: P"
else
    echo "Grade: F"
fi

```

Output:

```

./avgmarks.sh: line 22: syntax error: unexpected end of file
exam@cec-H110M-S2:~/mca_nsa$ ./avgmarks.sh
Enter three marks out of 100 each : 90 95 82
Average: 89
Grade: A
exam@cec-H110M-S2:~/mca_nsa$ ./avgmarks.sh
Enter three marks out of 100 each : 62 70 68
Average: 66
Grade: B
exam@cec-H110M-S2:~/mca_nsa$ ./avgmarks.sh
Enter three marks out of 100 each : 77 82 72
Average: 77
Grade: B

```

10. Read the name of an Indian state and display the main language according to the table. For other states, the output may be “Unknown”. Use “|” to separate states with same language (*case..esac*)

State	Main Language
Andhra Pradesh	Telugu
Assam	Assamese
Bihar	Hindi
Himachal Pradesh	Hindi
Karnataka	Kannada
Kerala	Malayalam
Lakshadweep	Malayalam
Tamil Nadu	Tamil

```

#!/bin/bash

echo -e "1.andhra pradesh \n2.assam \n3.bihar \n4.karnataka \n5.kerala \n6.tamil nadu \n7.Exit"


read -p "Enter the Indian state: " state

#state=$(echo $state | tr '[:upper:]' '[:lower:]')

```

```
case $state in
    1)
        echo "Language: Telugu"
        ;;
    2)
        echo "Language: Assamese"
        ;;
    3)
        echo "Language: Hindi"
        ;;
    4)
        echo "Language: Kannada"
        ;;
    5)
        echo "Language: Malayalam"
        ;;
    6)
        echo "Language: Tamil"
        ;;
    *)
        echo "Language: Unknown";;
esac
```

Output:

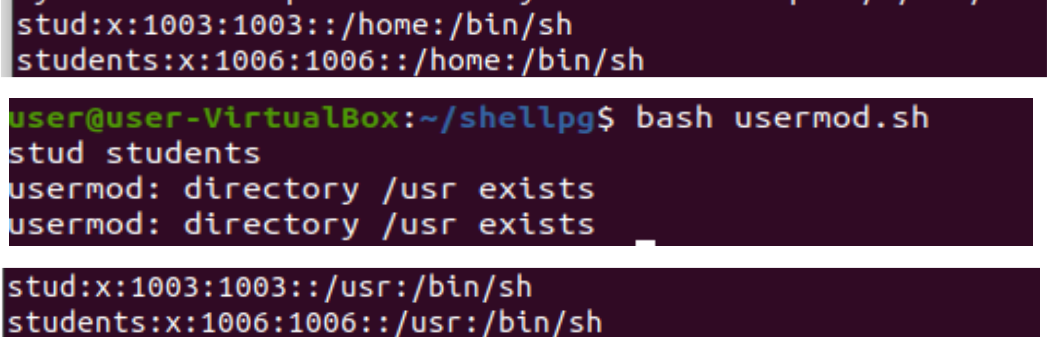


```
exam@cec-H110M-S2:~/mca_nsa$ ./state.sh
1.andhra pradesh
2.assam
3.bihar
4.karnataka
5.kerala
6.tamil nadu
7.Exit
Enter the Indian state: 4
Language: Kannada
```

11. Change the home folder of all users whose name start with stud from /home/username to /usr/username. Also change the password of username to username123 (e.g., /home/stud25 changes to /usr/stud25 and his/her password changes to stud25123) - (Use for .. in)

```
#!/bin/bash
result=$(grep stud* /etc/passwd)
result=$(echo "${result}" | cut -d: -f 1)
echo $result
for f in $result
do
    p="${f}123"
    sudo usermod -p $(echo $p | openssl passwd -1 -stdin) $f
    sudo usermod -m -d /usr $f
done
```

Output:



```
stud:x:1003:1003::/home:/bin/sh
students:x:1006:1006::/home:/bin/sh

user@user-VirtualBox:~/shellpg$ bash usermod.sh
stud students
usermod: directory /usr exists
usermod: directory /usr exists

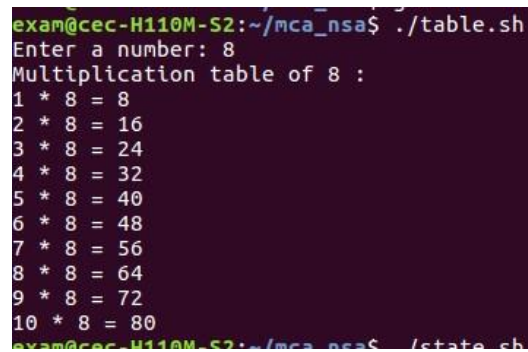
stud:x:1003:1003::/usr:/bin/sh
students:x:1006:1006::/usr:/bin/sh
```

12. Read a number and display the multiplication table of the number up to 10 lines. -
(Use for((..)))

```
#!/bin/bash
read -p "Enter a number: " num
echo "Multiplication table of $num : "
for (( i=1; i<=10; i++))
do
    val=$(( num * i ))
```

```
        echo "$i * $num = $val"
    done
```

Output:

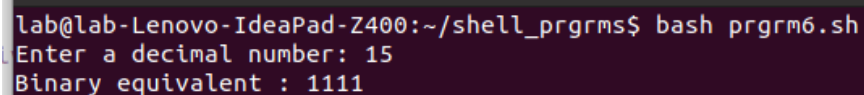


```
exam@cec-H110M-S2:~/mca_nsa$ ./table.sh
Enter a number: 8
Multiplication table of 8 :
1 * 8 = 8
2 * 8 = 16
3 * 8 = 24
4 * 8 = 32
5 * 8 = 40
6 * 8 = 48
7 * 8 = 56
8 * 8 = 64
9 * 8 = 72
10 * 8 = 80
exam@cec-H110M-S2:~/mca_nsa$ ./state.sh
```

13. Read a Decimal number. Convert it to Binary and display the result. -
(Use while)

```
#!/bin/bash
read -p "Enter a decimal number: " n
val=0
power=1
while [ $n -ne 0 ]
do
    r=`expr $n % 2`
    val=`expr $r \* $power + $val`
    power=`expr $power \* 10`
    n=`expr $n \ / 2`
done
echo "Binary equivalent : $val"
```

Output:



```
lab@lab-Lenovo-IdeaPad-Z400:~/shell_prgrms$ bash prgrm6.sh
Enter a decimal number: 15
Binary equivalent : 1111
```

Result:

Shell script program has done successfully and output is verified.