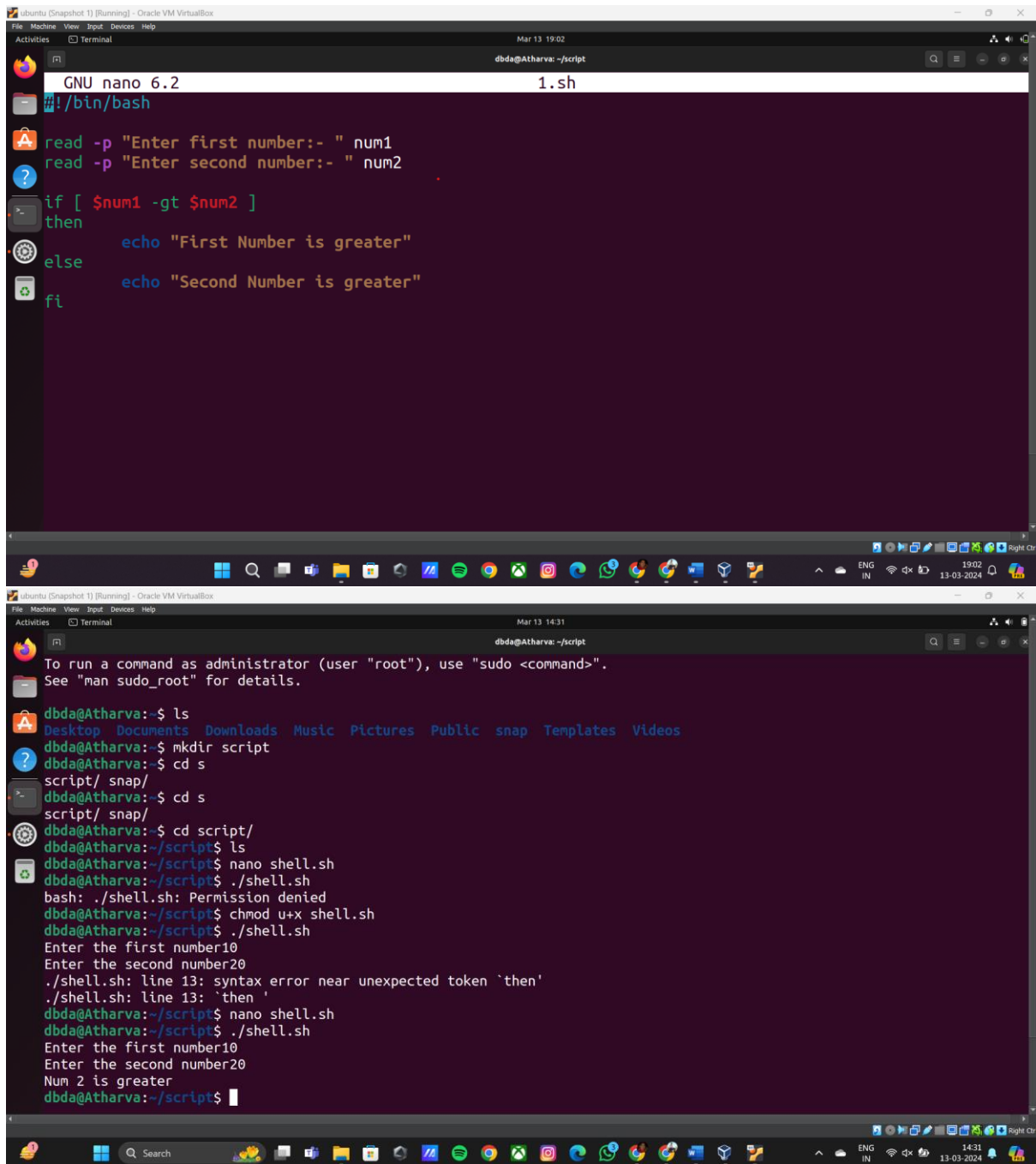


ROLL NO:- 243557

QUES 1:-



The image consists of two screenshots of a Linux terminal window running in Oracle VM VirtualBox. The top screenshot shows the creation of a shell script named '1.sh' using the nano text editor. The script prompts the user to enter two numbers, compares them, and prints a message indicating which is greater. The bottom screenshot shows the execution of the script, including directory navigation, file creation, permission setting, and the final output after user input.

```
GNU nano 6.2 1.sh
#!/bin/bash

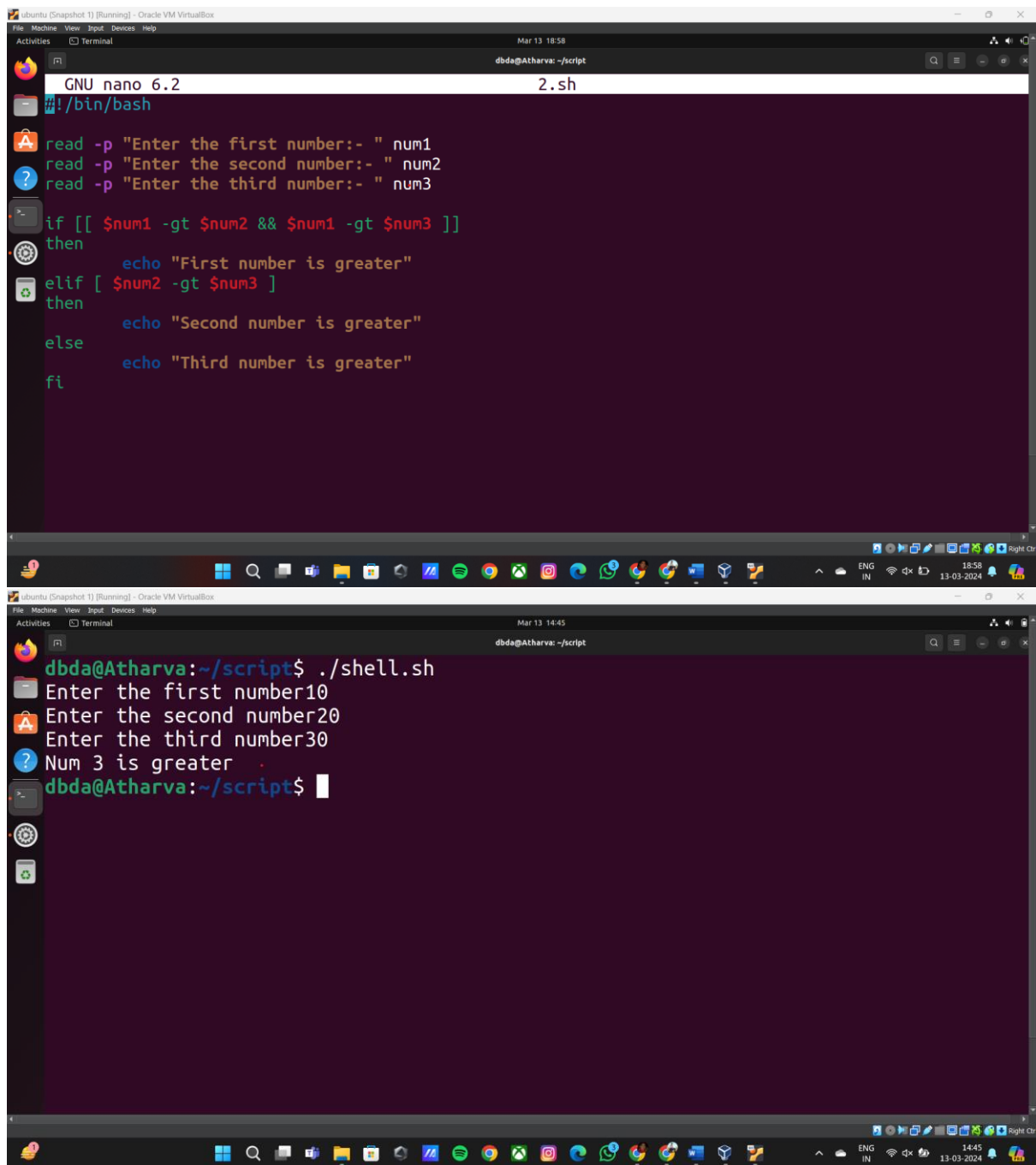
read -p "Enter first number:- " num1
read -p "Enter second number:- " num2

if [ $num1 -gt $num2 ]
then
    echo "First Number is greater"
else
    echo "Second Number is greater"
fi
```

```
dbda@Atharva: ~/script
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

dbda@Atharva:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
dbda@Atharva:~$ mkdir script
dbda@Atharva:~$ cd s
script/ snap/
dbda@Atharva:~$ cd s
script/ snap/
dbda@Atharva:~$ cd script/
dbda@Atharva:~/script$ ls
dbda@Atharva:~/script$ nano shell.sh
dbda@Atharva:~/script$ ./shell.sh
bash: ./shell.sh: Permission denied
dbda@Atharva:~/script$ chmod u+x shell.sh
dbda@Atharva:~/script$ ./shell.sh
Enter the first number10
Enter the second number20
./shell.sh: line 13: syntax error near unexpected token `then'
./shell.sh: line 13: `then '
dbda@Atharva:~/script$ nano shell.sh
dbda@Atharva:~/script$ ./shell.sh
Enter the first number10
Enter the second number20
Num 2 is greater
dbda@Atharva:~/script$
```

QUES 2:-



The image consists of two screenshots of a terminal window running in Oracle VM VirtualBox. The top screenshot shows the creation of a shell script named '2.sh' using the nano text editor. The script prompts the user to enter three numbers and then uses an if-elif-else statement to determine which number is the greatest. The bottom screenshot shows the execution of the script, where the user enters the numbers 10, 20, and 30, and the script correctly outputs 'Num 3 is greater'.

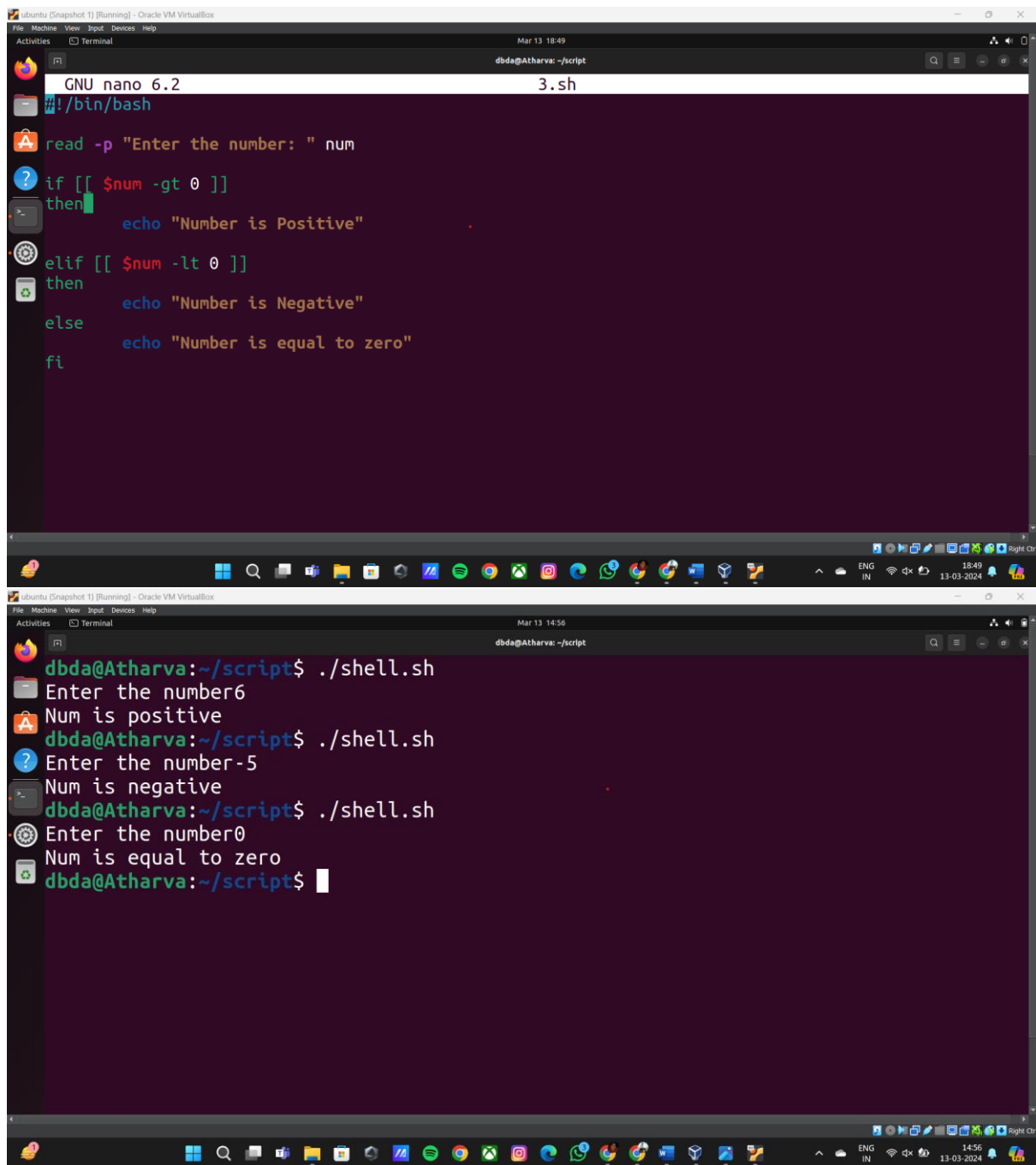
```
GNU nano 6.2 2.sh
#!/bin/bash

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

if [[ $num1 -gt $num2 && $num1 -gt $num3 ]]
then
    echo "First number is greater"
elif [ $num2 -gt $num3 ]
then
    echo "Second number is greater"
else
    echo "Third number is greater"
fi
```

```
dbda@Atharva: ~/script
dbda@Atharva:~/script$ ./shell.sh
Enter the first number10
Enter the second number20
Enter the third number30
Num 3 is greater
dbda@Atharva:~/script$
```

QUES 3:-



The image displays two screenshots of a terminal window within an Oracle VM VirtualBox environment. The top screenshot shows the creation of a shell script named `3.sh` using the `nano` editor. The script contains a `read` command to prompt the user for a number, followed by an `if` statement that checks if the number is greater than, less than, or equal to zero, and prints the corresponding message. The bottom screenshot shows the execution of the script `./shell.sh` three times, with the user entering 6, -5, and 0, and the script outputting "Num is positive", "Num is negative", and "Num is equal to zero" respectively.

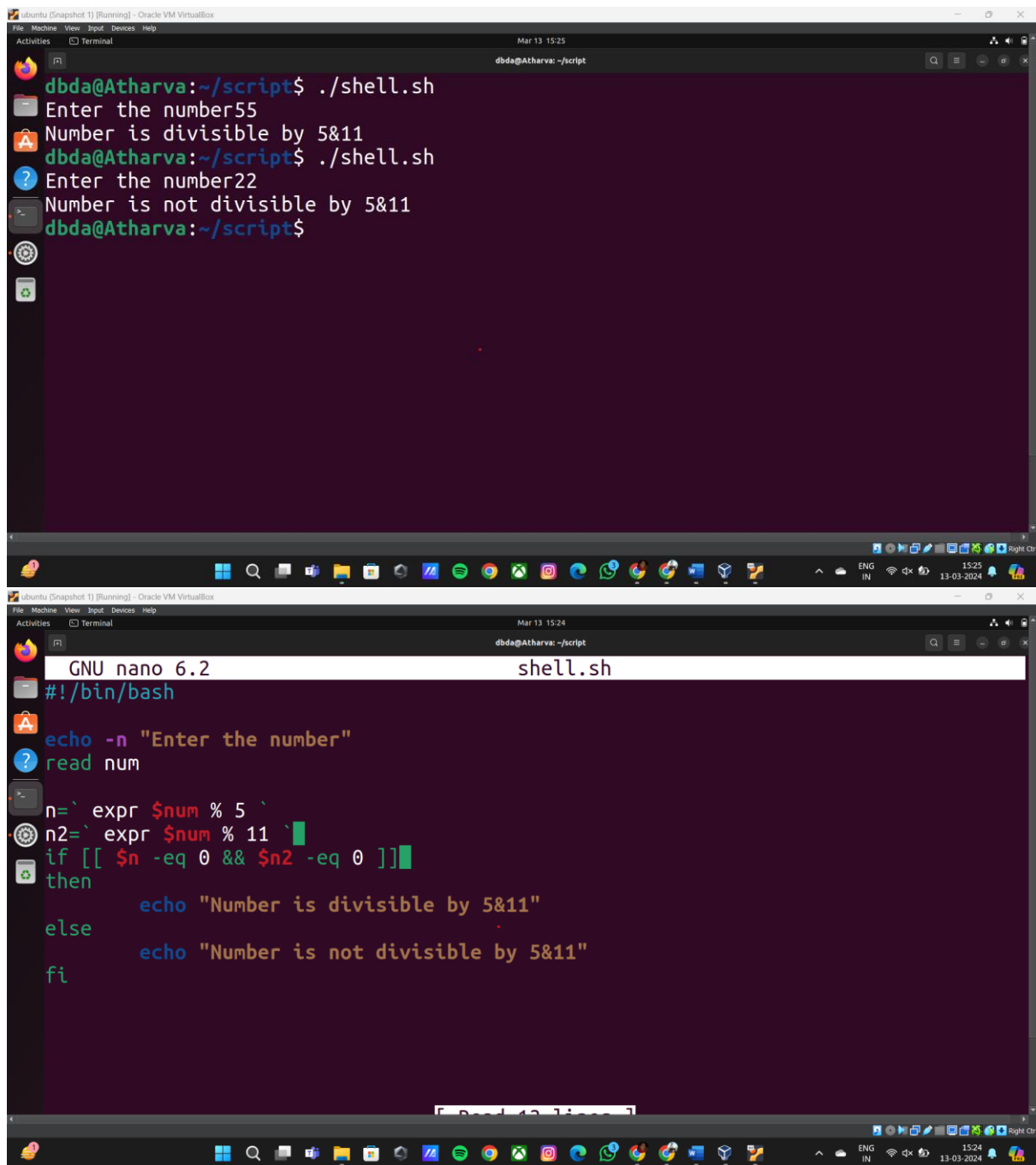
```
GNU nano 6.2 3.sh
#!/bin/bash

read -p "Enter the number: " num

if [[ $num -gt 0 ]]
then
    echo "Number is Positive"
elif [[ $num -lt 0 ]]
then
    echo "Number is Negative"
else
    echo "Number is equal to zero"
fi
```

```
dbda@Atharva: ~/script
dbda@Atharva:~/script$ ./shell.sh
Enter the number6
Num is positive
dbda@Atharva:~/script$ ./shell.sh
Enter the number-5
Num is negative
dbda@Atharva:~/script$ ./shell.sh
Enter the number0
Num is equal to zero
dbda@Atharva:~/script$
```

QUES 4:-



The image consists of two screenshots of a terminal window running inside an Oracle VM VirtualBox. The terminal is titled 'dbda@Atharva: ~/script' and shows the execution of a shell script named 'shell.sh'.

Top Screenshot: The terminal shows the user running `./shell.sh` twice. The first time, the user enters '55', and the script outputs 'Number is divisible by 5&11'. The second time, the user enters '22', and the script outputs 'Number is not divisible by 5&11'.

```
dbda@Atharva:~/script$ ./shell.sh
Enter the number55
Number is divisible by 5&11
dbda@Atharva:~/script$ ./shell.sh
Enter the number22
Number is not divisible by 5&11
dbda@Atharva:~/script$
```

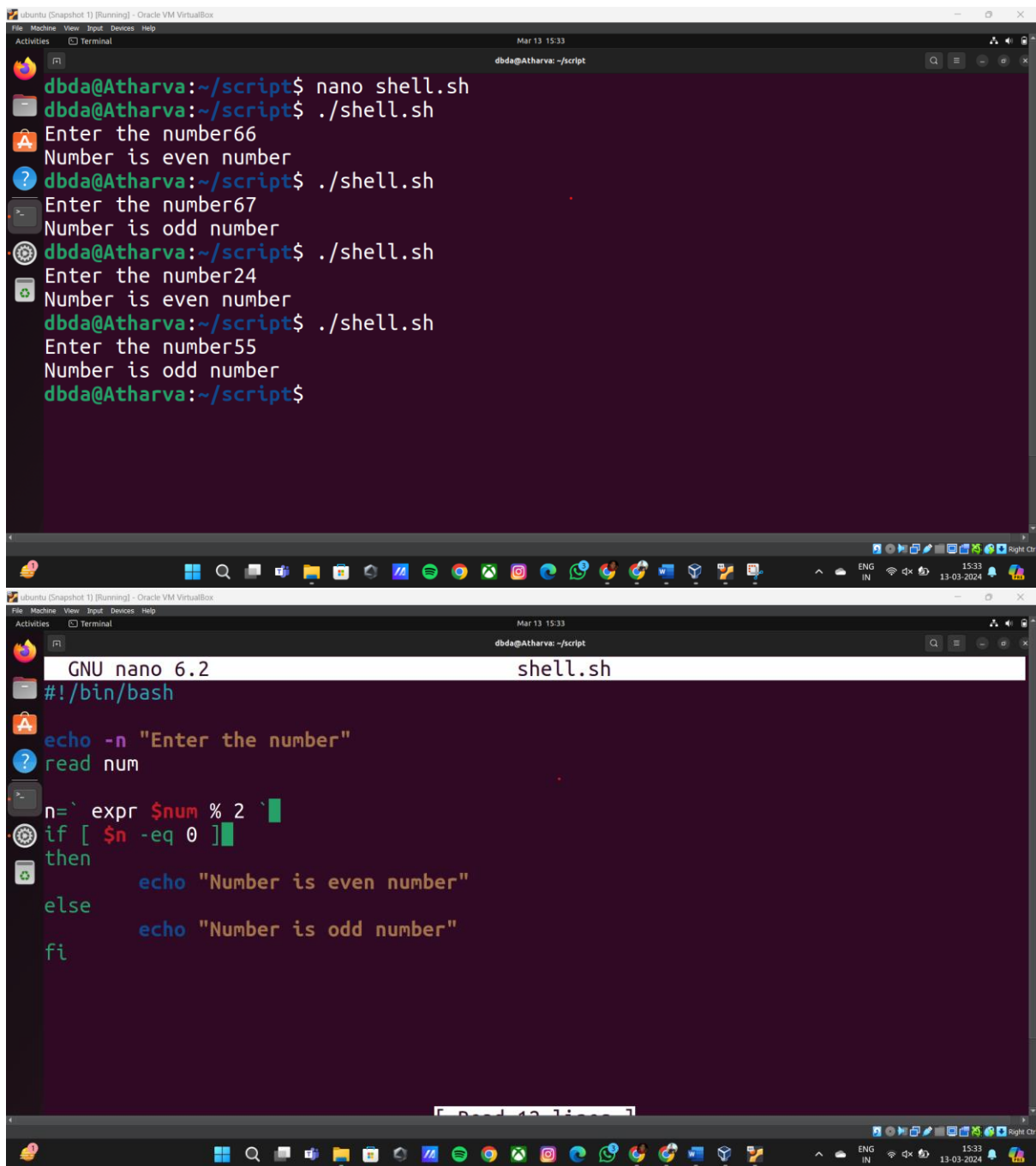
Bottom Screenshot: The terminal shows the source code of the 'shell.sh' script using the GNU nano 6.2 editor. The script prompts the user to enter a number, reads it into the variable 'num', calculates the remainder of 'num' divided by 5 and 11, and then checks if both remainders are zero. If so, it prints 'Number is divisible by 5&11'; otherwise, it prints 'Number is not divisible by 5&11'.

```
GNU nano 6.2 shell.sh
#!/bin/bash

echo -n "Enter the number"
read num

n=`expr $num % 5`
n2=`expr $num % 11`
if [[ $n -eq 0 && $n2 -eq 0 ]]
then
    echo "Number is divisible by 5&11"
else
    echo "Number is not divisible by 5&11"
fi
```

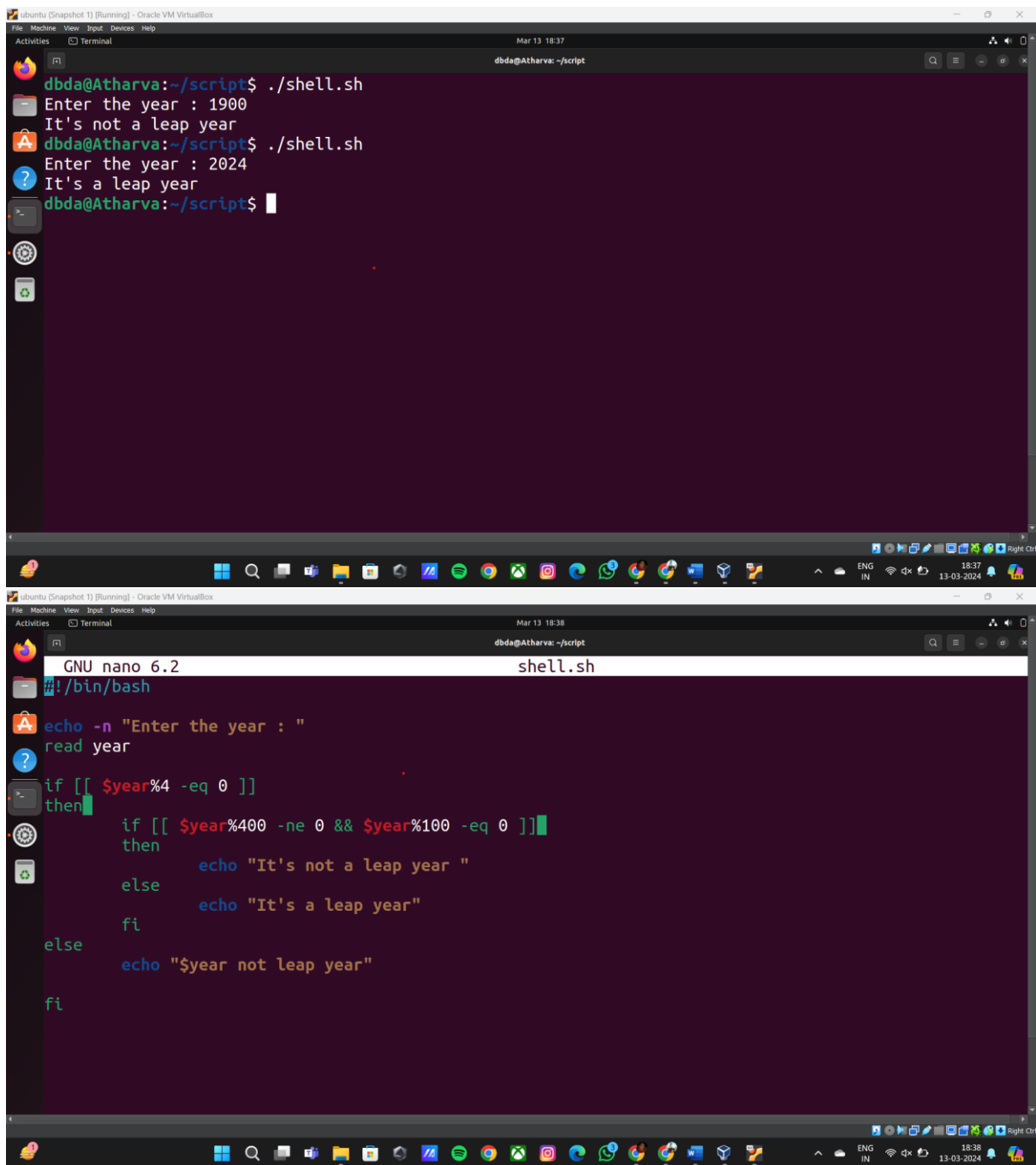
QUES 5:-



```
dbda@Atharva:~/script$ nano shell.sh
dbda@Atharva:~/script$ ./shell.sh
Enter the number66
Number is even number
dbda@Atharva:~/script$ ./shell.sh
Enter the number67
Number is odd number
dbda@Atharva:~/script$ ./shell.sh
Enter the number24
Number is even number
dbda@Atharva:~/script$ ./shell.sh
Enter the number55
Number is odd number
dbda@Atharva:~/script$
```

```
GNU nano 6.2 shell.sh
#!/bin/bash
echo -n "Enter the number"
read num
n=`expr $num % 2`
if [ $n -eq 0 ]
then
    echo "Number is even number"
else
    echo "Number is odd number"
fi
```

QUES 6:-



The image displays two screenshots of a terminal window running within Oracle VM VirtualBox. The terminal is titled 'dbda@Atharva: ~/script' and shows the execution of a shell script named 'shell.sh'.

Top Screenshot: The terminal shows the user running `./shell.sh` twice. The first run uses the year 1900, and the second run uses the year 2024. The script correctly identifies 1900 as not a leap year and 2024 as a leap year.

```
dbda@Atharva:~/script$ ./shell.sh
Enter the year : 1900
It's not a leap year
dbda@Atharva:~/script$ ./shell.sh
Enter the year : 2024
It's a leap year
dbda@Atharva:~/script$
```

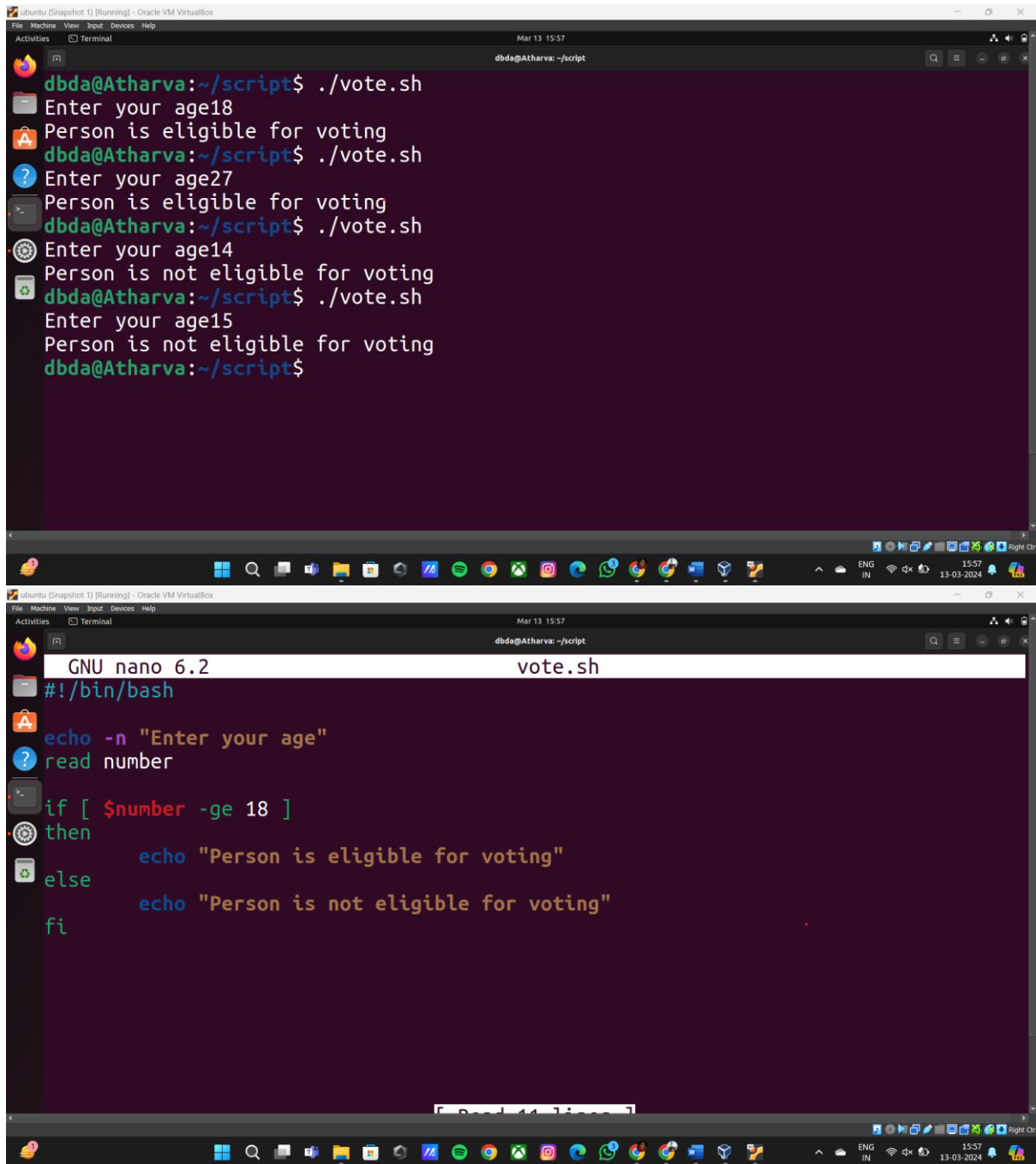
Bottom Screenshot: The terminal shows the source code of the 'shell.sh' script using the GNU nano 6.2 editor. The script uses a series of if-else statements to check if a year is a leap year based on divisibility by 4, 100, and 400.

```
GNU nano 6.2 shell.sh
#!/bin/bash

echo -n "Enter the year : "
read year

if [[ $year%4 -eq 0 ]]
then
    if [[ $year%400 -ne 0 && $year%100 -eq 0 ]]
    then
        echo "It's not a leap year "
    else
        echo "It's a leap year"
    fi
else
    echo "$year not leap year"
fi
```

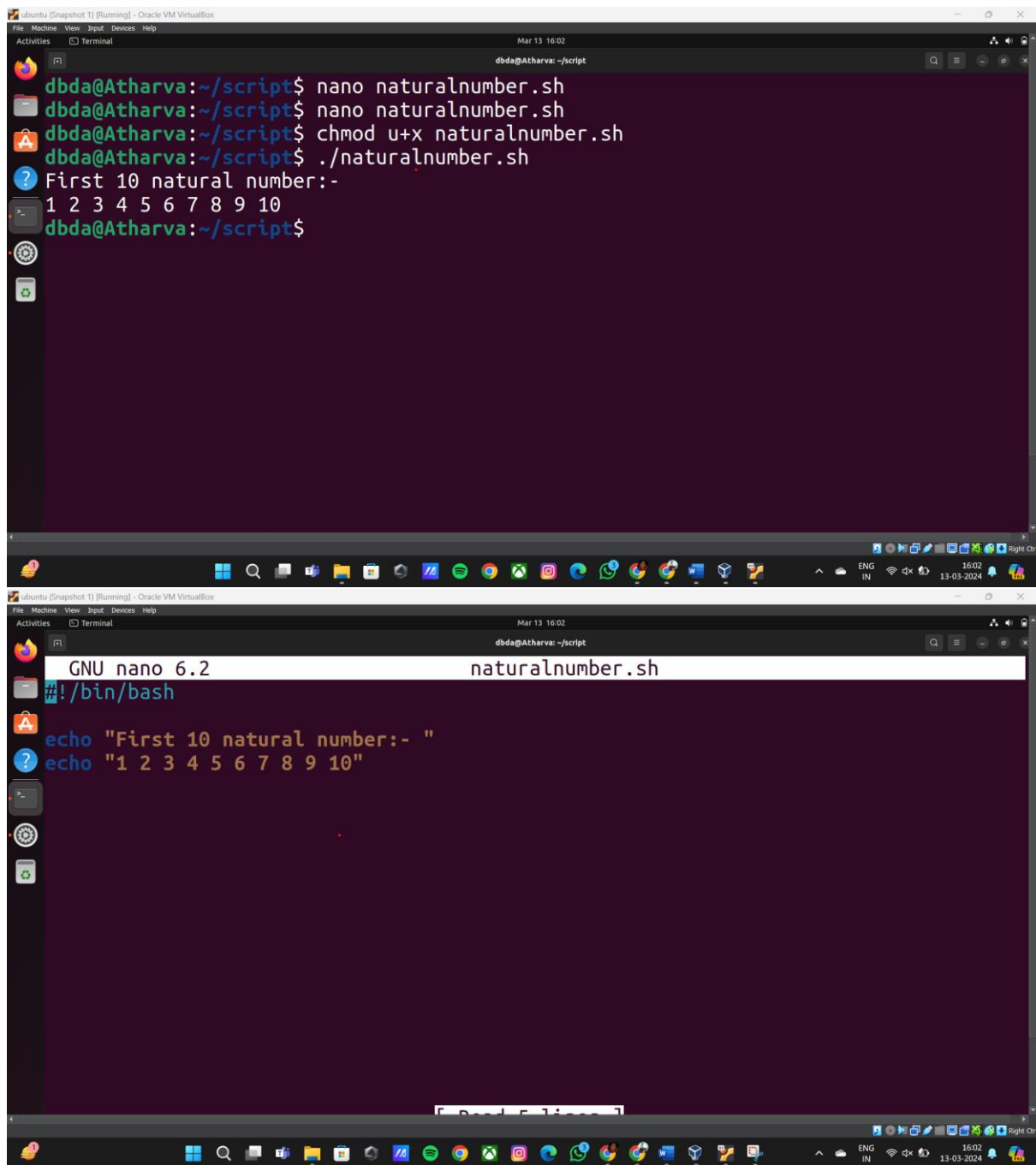
QUES 7:-



The image consists of two screenshots from a Linux virtual machine. The top screenshot shows a terminal window where a script named `vote.sh` is being executed. The user enters ages 18, 27, 14, and 15. The script checks if the age is greater than or equal to 18 and prints "Person is eligible for voting" or "Person is not eligible for voting". The bottom screenshot shows the same terminal window with the `vote.sh` script open in the `nano` text editor. The script content is as follows:

```
GNU nano 6.2 vote.sh
#!/bin/bash
echo -n "Enter your age"
read number
if [ $number -ge 18 ]
then
    echo "Person is eligible for voting"
else
    echo "Person is not eligible for voting"
fi
```

QUES 8:-



The image consists of two screenshots of a terminal window running inside an Oracle VM VirtualBox. The terminal is titled 'ubuntu (Snapshot 1) [Running] - Oracle VM VirtualBox' and shows a user 'dbda' at host 'Atharva' in the directory '~/script'.

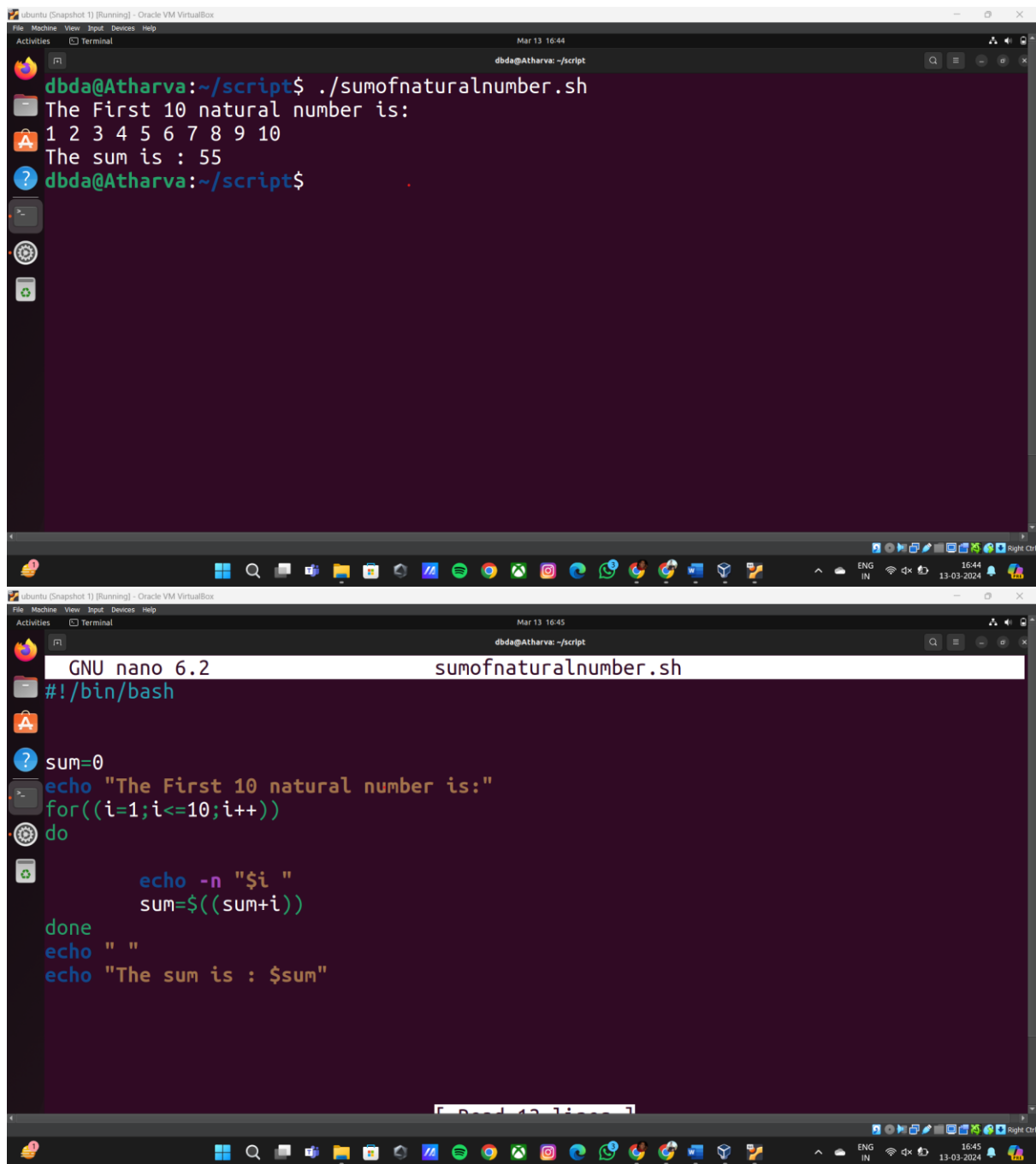
Top Screenshot: The user creates a file named 'naturalnumber.sh' using 'nano', sets permissions with 'chmod u+x naturalnumber.sh', and runs the script with './naturalnumber.sh'. The script outputs 'First 10 natural number:-' followed by the numbers '1 2 3 4 5 6 7 8 9 10'.

```
dbda@Atharva:~/script$ nano naturalnumber.sh
dbda@Atharva:~/script$ nano naturalnumber.sh
dbda@Atharva:~/script$ chmod u+x naturalnumber.sh
dbda@Atharva:~/script$ ./naturalnumber.sh
First 10 natural number:-
1 2 3 4 5 6 7 8 9 10
dbda@Atharva:~/script$
```

Bottom Screenshot: This screenshot shows the 'nano' text editor editing 'naturalnumber.sh'. The file contains two lines: an echo statement for the header and an echo statement for the list of numbers.

```
GNU nano 6.2 naturalnumber.sh
#!/bin/bash
echo "First 10 natural number:- "
echo "1 2 3 4 5 6 7 8 9 10"
```


QUES 9:-



The image displays two screenshots of a terminal window within an Oracle VM VirtualBox environment. The terminal is running Ubuntu (Snapshot 1) and the user is dbda@Atharva.

Top Screenshot: The user has executed the command `./sumofnaturalnumber.sh`. The output shows the first 10 natural numbers and their sum.

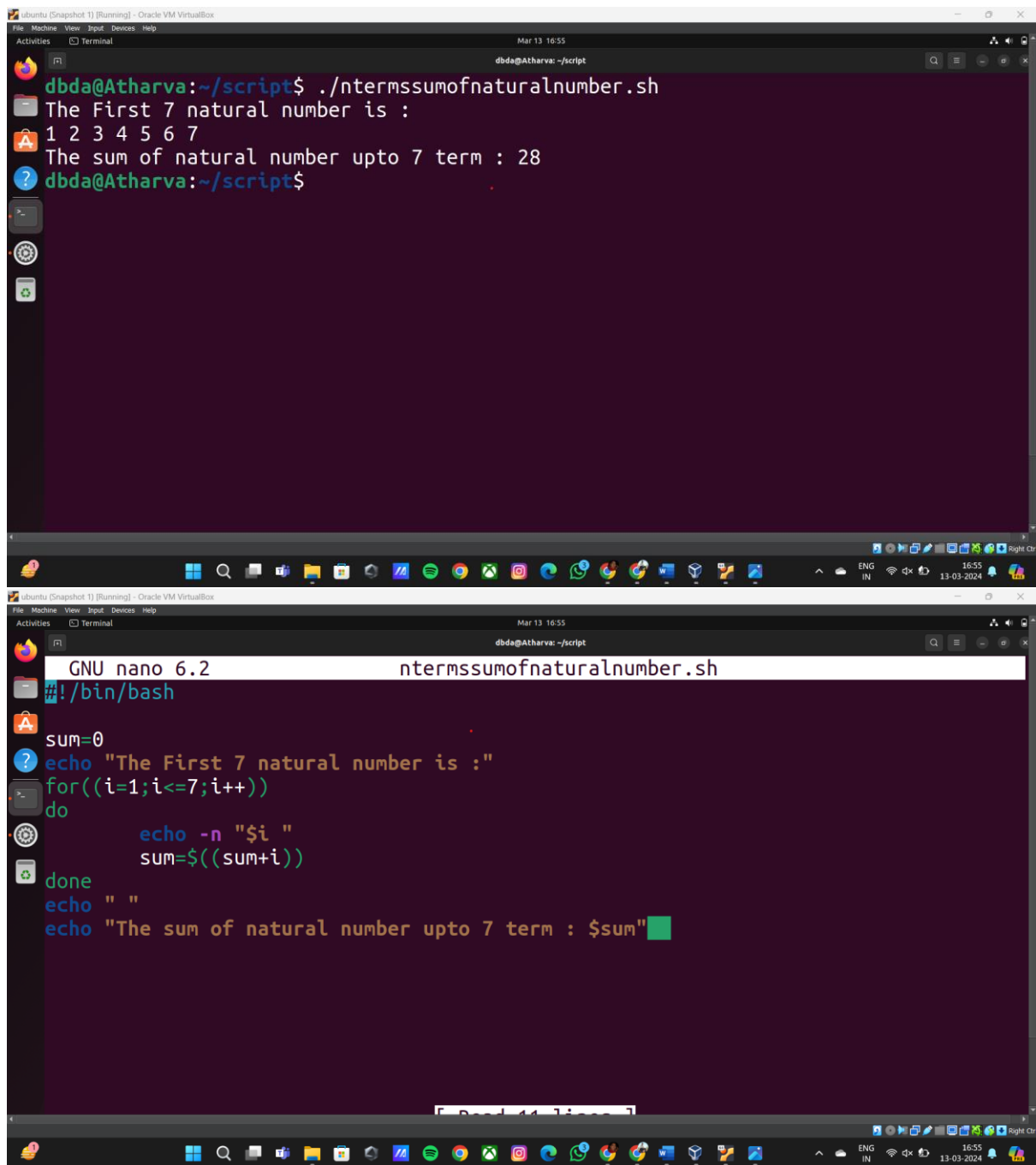
```
dbda@Atharva:~/script$ ./sumofnaturalnumber.sh
The First 10 natural number is:
1 2 3 4 5 6 7 8 9 10
The sum is : 55
dbda@Atharva:~/script$
```

Bottom Screenshot: The user has opened the script file `sumofnaturalnumber.sh` using the `nano` editor. The script's content is as follows:

```
GNU nano 6.2 sumofnaturalnumber.sh
#!/bin/bash

sum=0
echo "The First 10 natural number is:"
for((i=1;i<=10;i++))
do
    echo -n "$i "
    sum=$((sum+i))
done
echo " "
echo "The sum is : $sum"
```

QUES 10:-

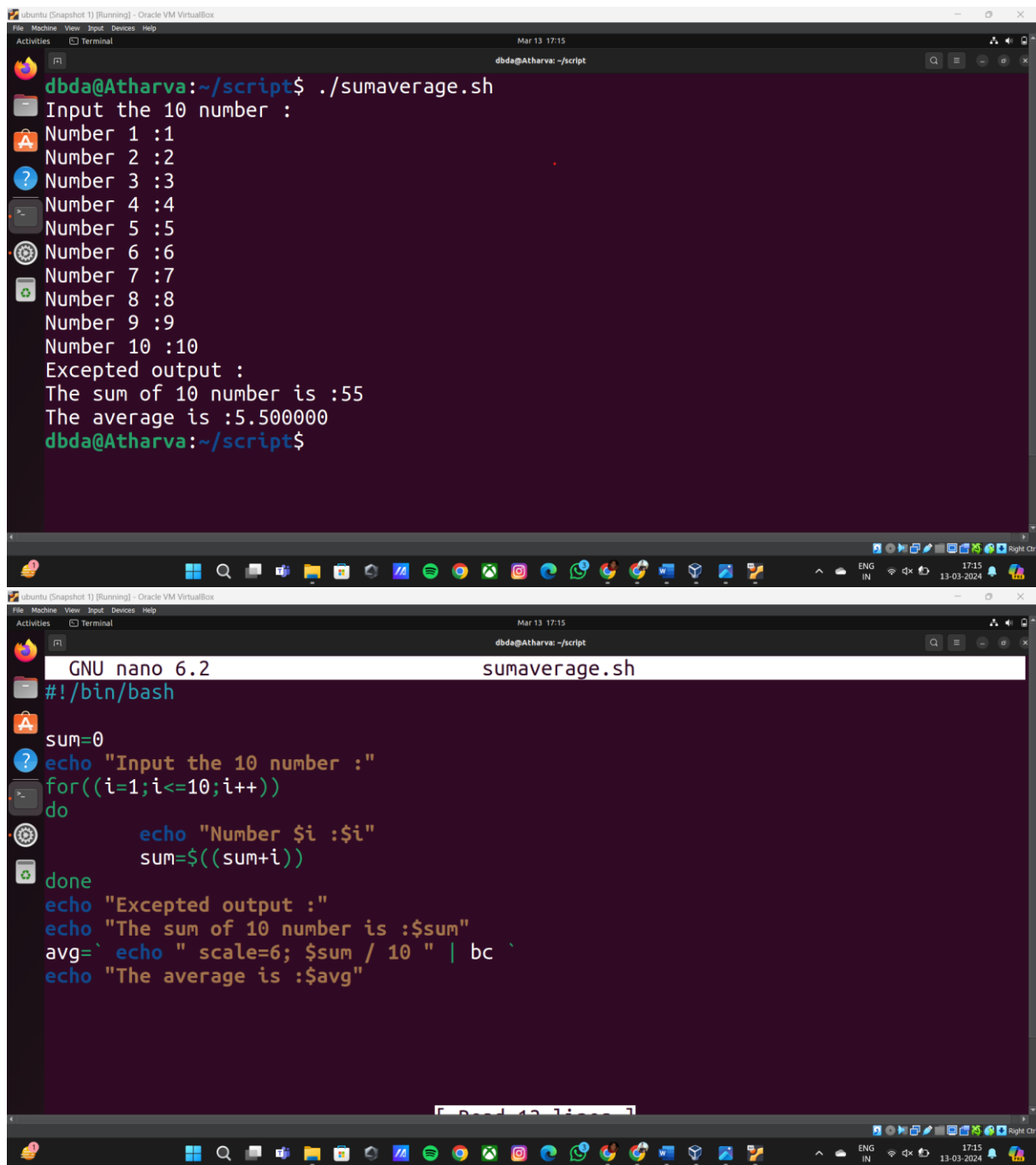


The image consists of two screenshots of a terminal window running on a virtual machine. The top screenshot shows the execution of a script named `ntermssumofnaturalnumber.sh`. The output displays the first 7 natural numbers and their sum, which is 28. The bottom screenshot shows the source code of the script, which uses a `for` loop to calculate the sum of natural numbers from 1 to 7.

```
dbda@Atharva:~/script$ ./ntermssumofnaturalnumber.sh
The First 7 natural number is :
1 2 3 4 5 6 7
The sum of natural number upto 7 term : 28
dbda@Atharva:~/script$
```

```
GNU nano 6.2 ntermssumofnaturalnumber.sh
#!/bin/bash
sum=0
echo "The First 7 natural number is :"
for((i=1;i<=7;i++))
do
    echo -n "$i "
    sum=$((sum+i))
done
echo " "
echo "The sum of natural number upto 7 term : $sum"
```

QUES 11:-



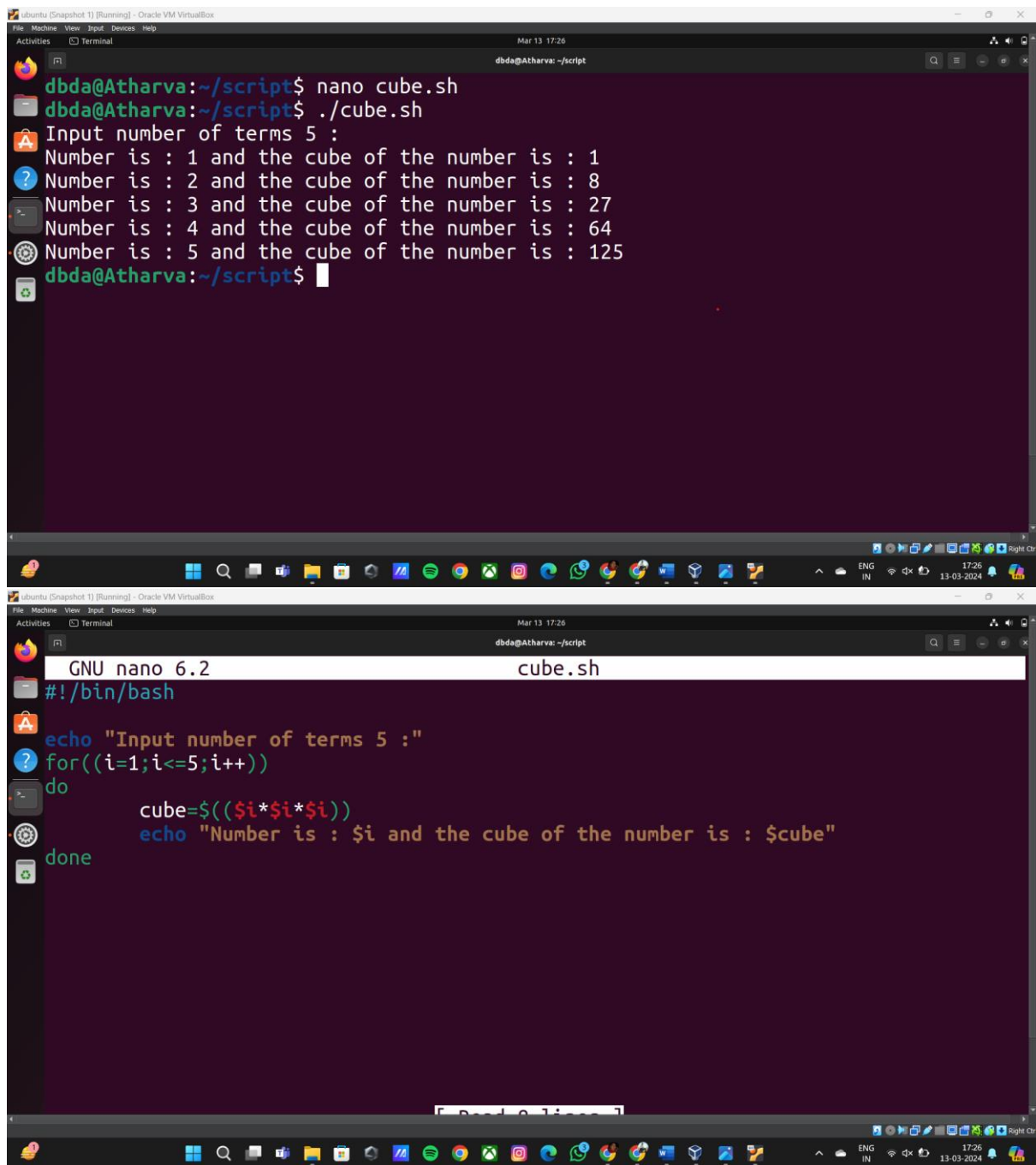
The image consists of two screenshots of a Linux terminal window, likely running on a virtual machine. The top screenshot shows the execution of a script named `sumaverage.sh`. The user `dbda@Atharva` runs the command `./sumaverage.sh`. The script prompts for 10 numbers, which are entered as 1 through 10. It then calculates the sum (55) and the average (5.500000). The bottom screenshot shows the source code of the `sumaverage.sh` script in the `nano` editor. The script uses a `for` loop to calculate the sum of numbers 1 to 10 and then uses `bc` to calculate the average.

```
dbda@Atharva:~/script$ ./sumaverage.sh
Input the 10 number :
Number 1 :1
Number 2 :2
Number 3 :3
Number 4 :4
Number 5 :5
Number 6 :6
Number 7 :7
Number 8 :8
Number 9 :9
Number 10 :10
Excepected output :
The sum of 10 number is :55
The average is :5.500000
dbda@Atharva:~/script$
```

```
GNU nano 6.2 sumaverage.sh
#!/bin/bash

sum=0
echo "Input the 10 number :"
for((i=1;i<=10;i++))
do
    echo "Number $i :$i"
    sum=$((sum+i))
done
echo "Excepected output :"
echo "The sum of 10 number is :$sum"
avg=`echo " scale=6; $sum / 10 " | bc `
echo "The average is :$avg"
```

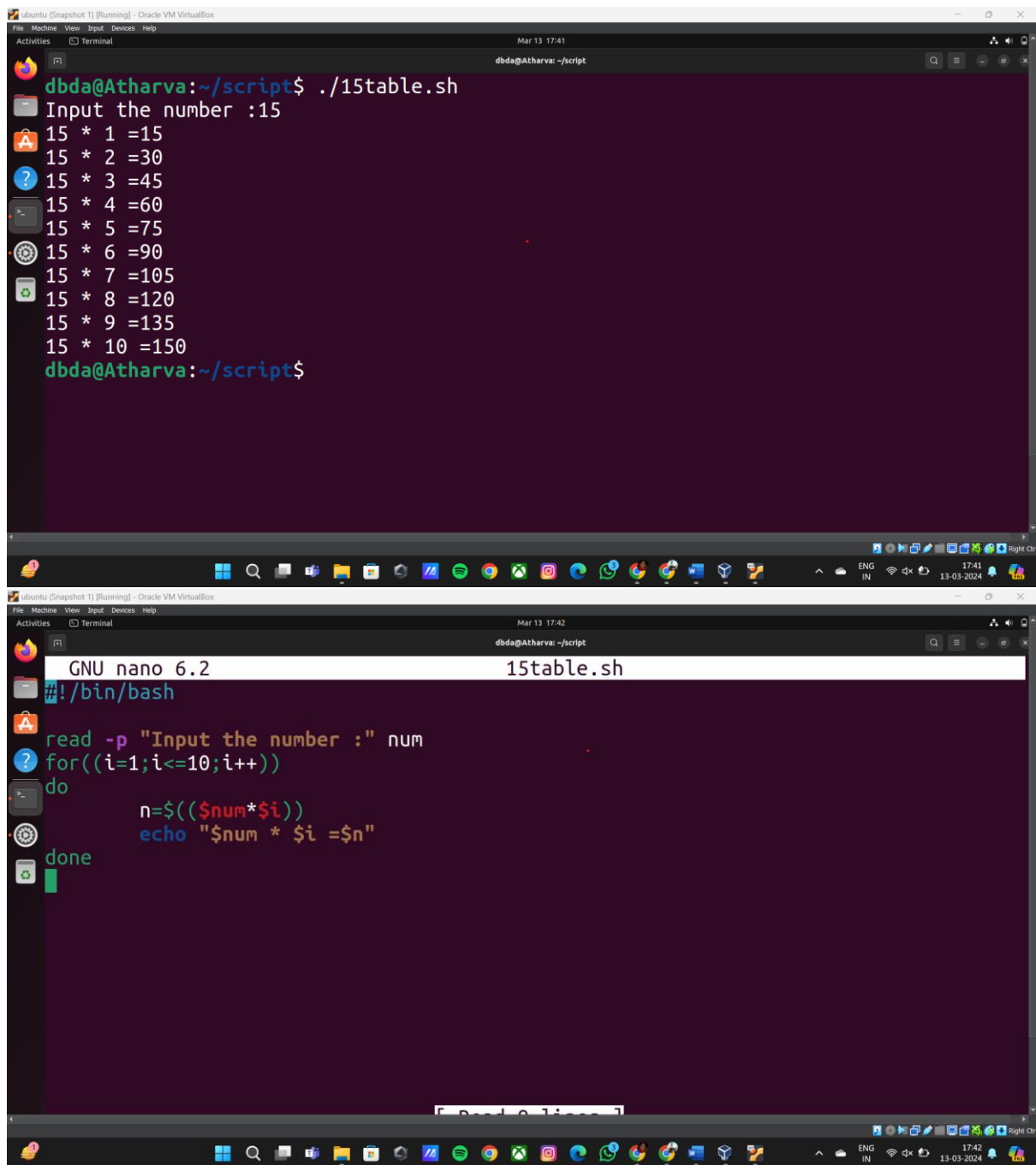
QUES 12:-



```
dbda@Atharva:~/script$ nano cube.sh
dbda@Atharva:~/script$ ./cube.sh
Input number of terms 5 :
Number is : 1 and the cube of the number is : 1
Number is : 2 and the cube of the number is : 8
Number is : 3 and the cube of the number is : 27
Number is : 4 and the cube of the number is : 64
Number is : 5 and the cube of the number is : 125
dbda@Atharva:~/script$
```

```
GNU nano 6.2 cube.sh
#!/bin/bash
echo "Input number of terms 5 :"
for((i=1;i<=5;i++))
do
    cube=$((i*i*i))
    echo "Number is : $i and the cube of the number is : $cube"
done
```

QUES 13:-

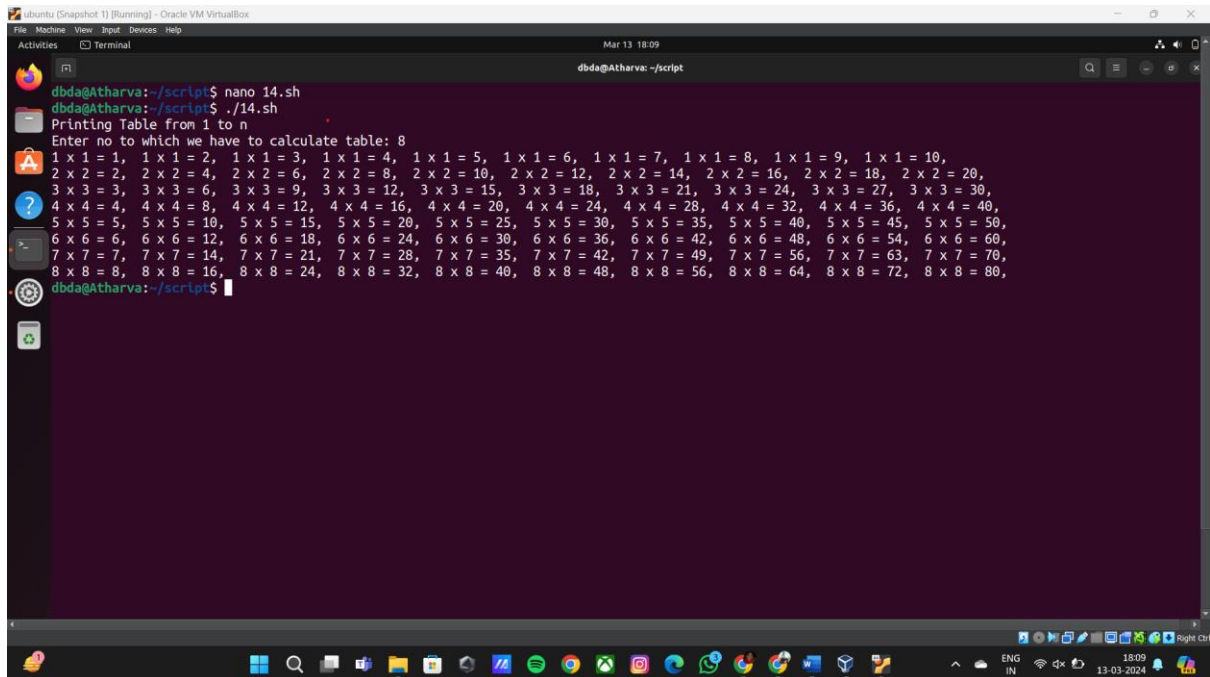


The image consists of two screenshots of a terminal window running in Oracle VM VirtualBox. The top screenshot shows the execution of a script named `15table.sh`. The user `dbda@Atharva` is in the directory `~/script`. The script prompts for an input number, and the user enters `15`. The script then displays the multiplication table for 15, from 1 to 10. The bottom screenshot shows the source code of the `15table.sh` script in the GNU nano 6.2 editor. The script uses `read` to get input, a `for` loop to iterate from 1 to 10, and `echo` to print the multiplication results.

```
dbda@Atharva:~/script$ ./15table.sh
Input the number :15
15 * 1 =15
15 * 2 =30
15 * 3 =45
15 * 4 =60
15 * 5 =75
15 * 6 =90
15 * 7 =105
15 * 8 =120
15 * 9 =135
15 * 10 =150
dbda@Atharva:~/script$
```

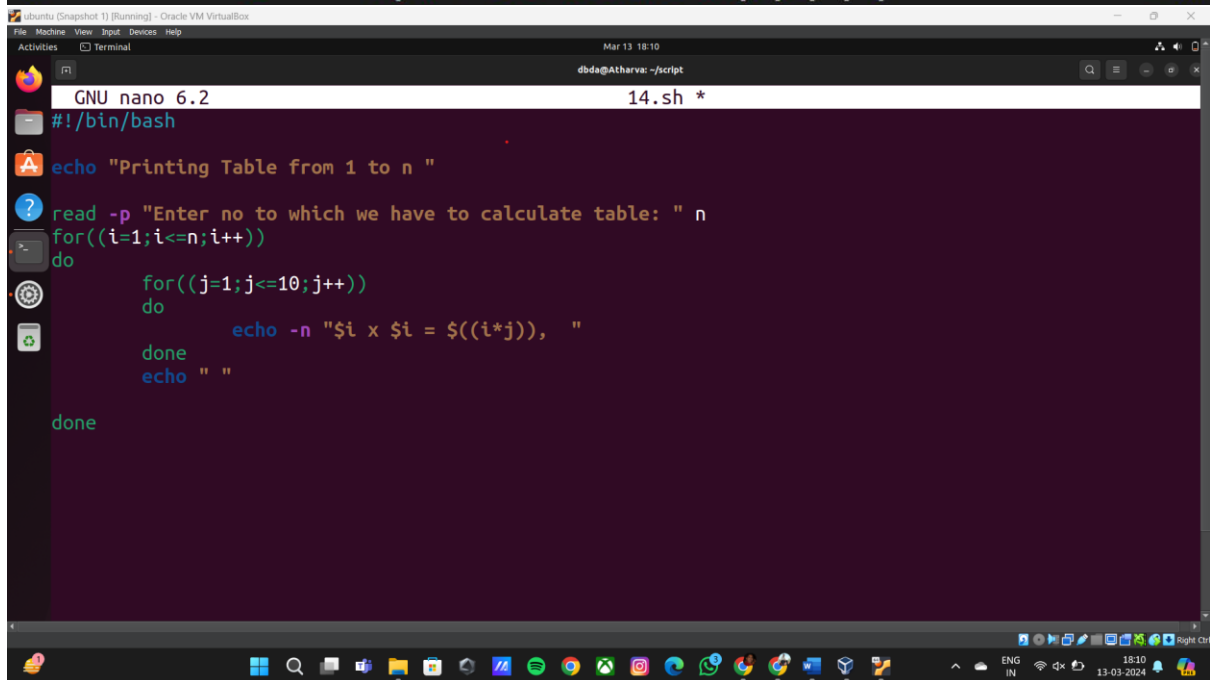
```
GNU nano 6.2 15table.sh
#!/bin/bash
read -p "Input the number :" num
for((i=1;i<=10;i++))
do
    n=$(( $num * $i ))
    echo "$num * $i =$n"
done
```

QUES 14:-



The screenshot shows a terminal window titled 'Terminal' with the prompt 'dbda@Atharva: ~/script'. The user has executed the command 'nano 14.sh' and then './14.sh'. The script prints 'Printing Table from 1 to n' and prompts the user to 'Enter no to which we have to calculate table:'. The user has entered '8'. The script then prints a multiplication table for numbers 1 through 8. The table is displayed as a single line of text with each multiplication result separated by a comma.

```
dbda@Atharva:~/script$ nano 14.sh
dbda@Atharva:~/script$ ./14.sh
Printing Table from 1 to n
Enter no to which we have to calculate table: 8
1 x 1 = 1, 1 x 1 = 2, 1 x 1 = 3, 1 x 1 = 4, 1 x 1 = 5, 1 x 1 = 6, 1 x 1 = 7, 1 x 1 = 8, 1 x 1 = 9, 1 x 1 = 10,
2 x 2 = 2, 2 x 2 = 4, 2 x 2 = 6, 2 x 2 = 8, 2 x 2 = 10, 2 x 2 = 12, 2 x 2 = 14, 2 x 2 = 16, 2 x 2 = 18, 2 x 2 = 20,
3 x 3 = 3, 3 x 3 = 6, 3 x 3 = 9, 3 x 3 = 12, 3 x 3 = 15, 3 x 3 = 18, 3 x 3 = 21, 3 x 3 = 24, 3 x 3 = 27, 3 x 3 = 30,
4 x 4 = 4, 4 x 4 = 8, 4 x 4 = 12, 4 x 4 = 16, 4 x 4 = 20, 4 x 4 = 24, 4 x 4 = 28, 4 x 4 = 32, 4 x 4 = 36, 4 x 4 = 40,
5 x 5 = 5, 5 x 5 = 10, 5 x 5 = 15, 5 x 5 = 20, 5 x 5 = 25, 5 x 5 = 30, 5 x 5 = 35, 5 x 5 = 40, 5 x 5 = 45, 5 x 5 = 50,
6 x 6 = 6, 6 x 6 = 12, 6 x 6 = 18, 6 x 6 = 24, 6 x 6 = 30, 6 x 6 = 36, 6 x 6 = 42, 6 x 6 = 48, 6 x 6 = 54, 6 x 6 = 60,
7 x 7 = 7, 7 x 7 = 14, 7 x 7 = 21, 7 x 7 = 28, 7 x 7 = 35, 7 x 7 = 42, 7 x 7 = 49, 7 x 7 = 56, 7 x 7 = 63, 7 x 7 = 70,
8 x 8 = 8, 8 x 8 = 16, 8 x 8 = 24, 8 x 8 = 32, 8 x 8 = 40, 8 x 8 = 48, 8 x 8 = 56, 8 x 8 = 64, 8 x 8 = 72, 8 x 8 = 80,
```



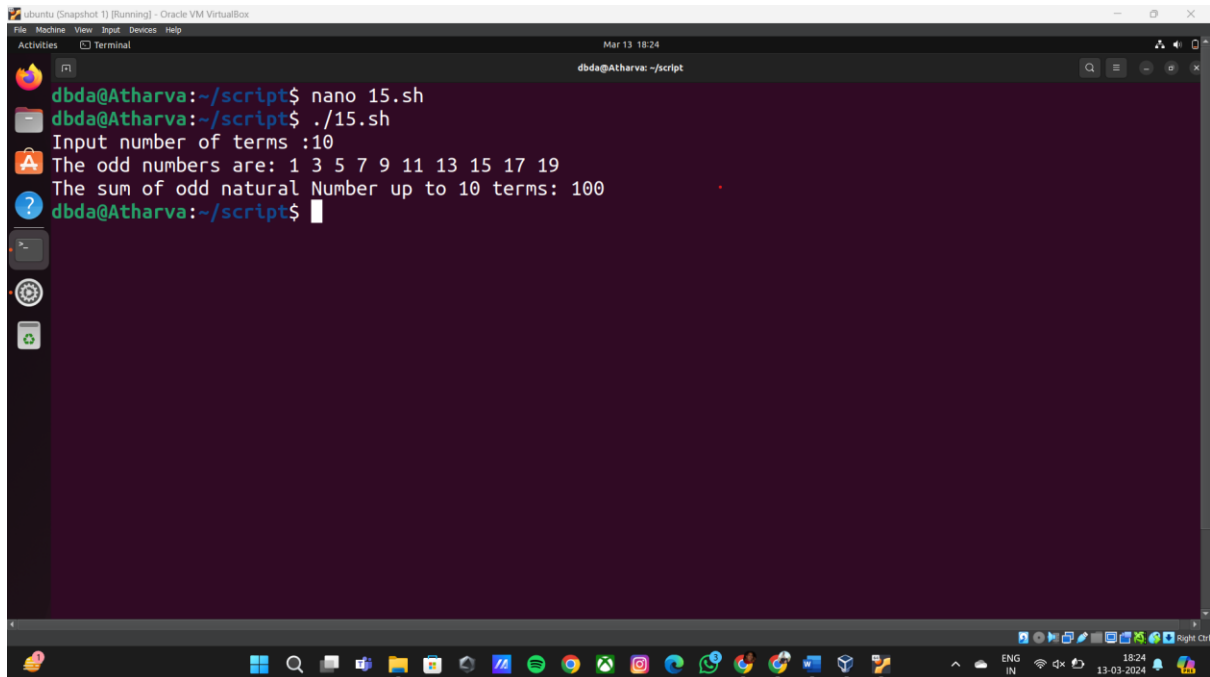
The screenshot shows the same terminal window with the script '14.sh' open in the nano editor. The code is as follows:

```
GNU nano 6.2 14.sh *
#!/bin/bash

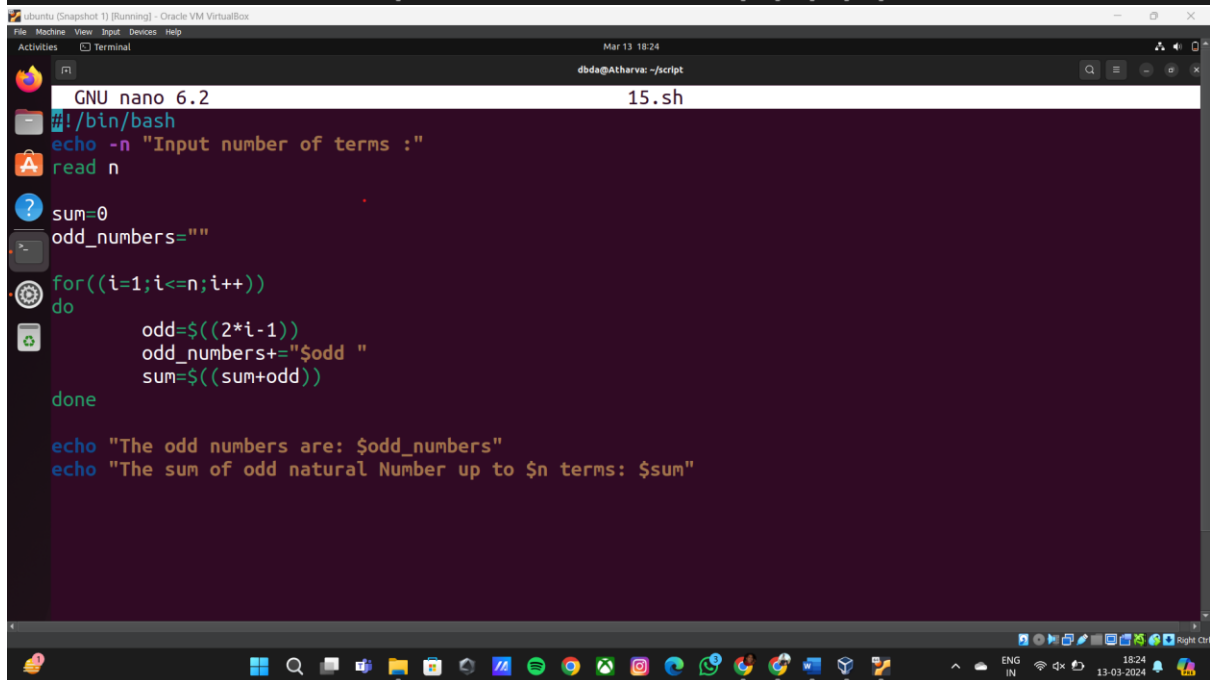
echo "Printing Table from 1 to n "

read -p "Enter no to which we have to calculate table: " n
for((i=1;i<=n;i++))
do
    for((j=1;j<=10;j++))
    do
        echo -n "$i x $i = $((i*j)), "
    done
    echo " "
done
```

QUES 15:-



```
dbda@Atharva:~/script$ nano 15.sh
dbda@Atharva:~/script$ ./15.sh
Input number of terms :10
The odd numbers are: 1 3 5 7 9 11 13 15 17 19
The sum of odd natural Number up to 10 terms: 100
dbda@Atharva:~/script$
```



```
GNU nano 6.2 15.sh
#!/bin/bash
echo -n "Input number of terms : "
read n
sum=0
odd_numbers=""
for((i=1;i<=n;i++))
do
    odd=$((2*i-1))
    odd_numbers+="$odd "
    sum=$((sum+odd))
done
echo "The odd numbers are: $odd_numbers"
echo "The sum of odd natural Number up to $n terms: $sum"
```