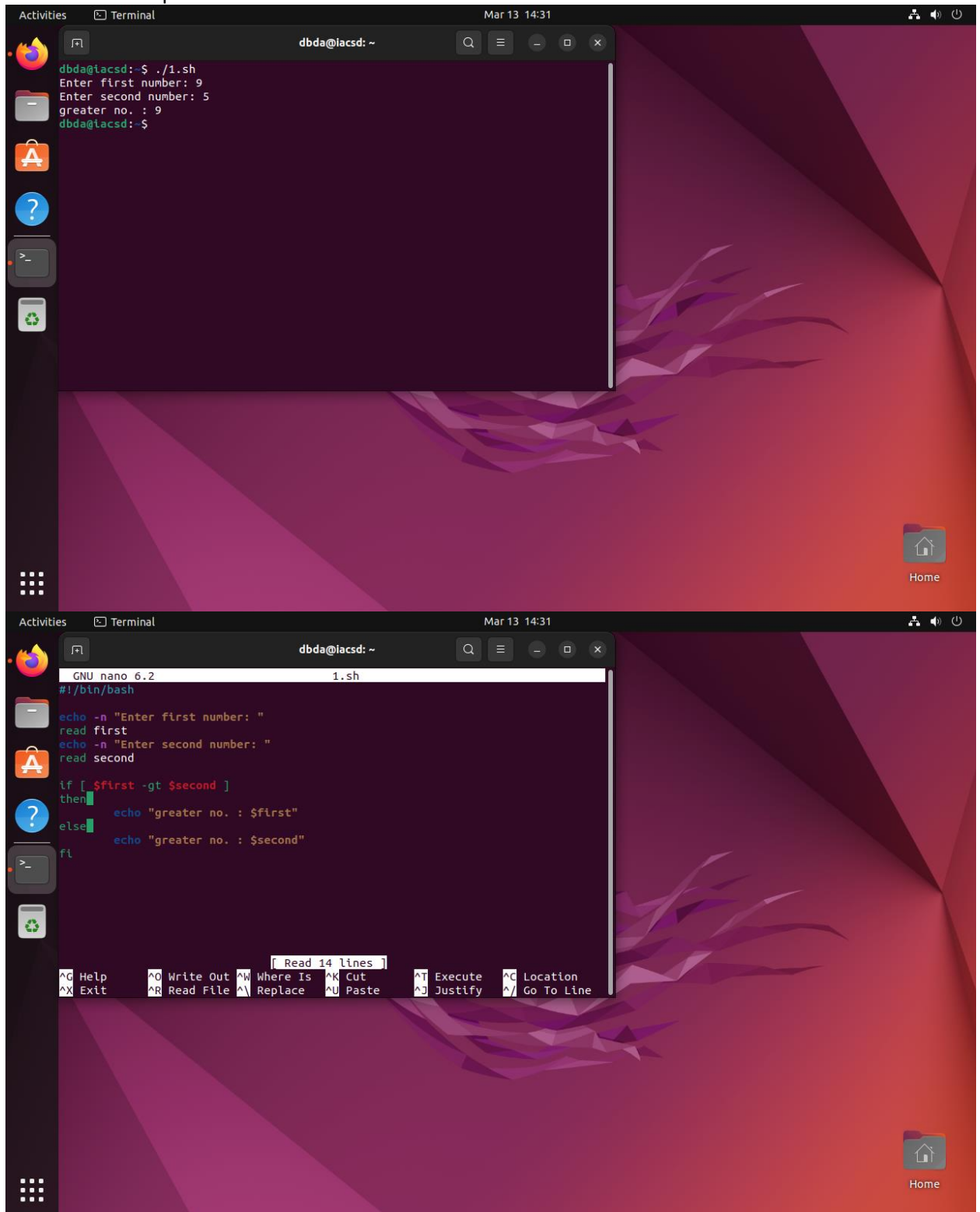


Assignment-3

1. Write a Shell Script to find maximum between two numbers.



The image displays two screenshots of a Linux desktop environment, specifically the Ubuntu 18.04 LTS version, showing a terminal window. The desktop background is a purple and red geometric pattern. The terminal window is titled "dbda@lacs: ~" and shows the execution of a shell script.

Top Screenshot: The terminal shows the execution of a script named `1.sh`. The user enters the command `./1.sh`, and the script prompts for two numbers. The first number entered is 9, and the second number entered is 5. The script outputs "greater no. : 9".

```
dbda@lacs:~$ ./1.sh
Enter first number: 9
Enter second number: 5
greater no. : 9
dbda@lacs:~$
```

Bottom Screenshot: The terminal shows the same script being edited in the `nano` editor. The script content is as follows:

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

echo -n "Enter first number: "
read first
echo -n "Enter second number: "
read second

if [ $first -gt $second ]
then
    echo "greater no. : $first"
else
    echo "greater no. : $second"
fi
```

The bottom screenshot also shows the `nano` editor's status bar at the bottom, indicating "Read 14 lines" and providing various keyboard shortcuts for editing and navigation.

2. Write a Shell Script to find maximum between three numbers.

The image displays two sequential screenshots of a Linux desktop environment, specifically Ubuntu, with a terminal window open. The desktop background is a purple and pink geometric pattern. The top-left dock contains icons for the Dash, Firefox, Files, Application Store, Help, and System Monitor. The top-right system tray shows the date and time as 'Mar 13 14:47' and icons for network, sound, and power.

Top Screenshot: The terminal window, titled 'dbda@iacsd: ~', shows the GNU nano 6.2 text editor editing a file named '2.sh'. The script content is as follows:

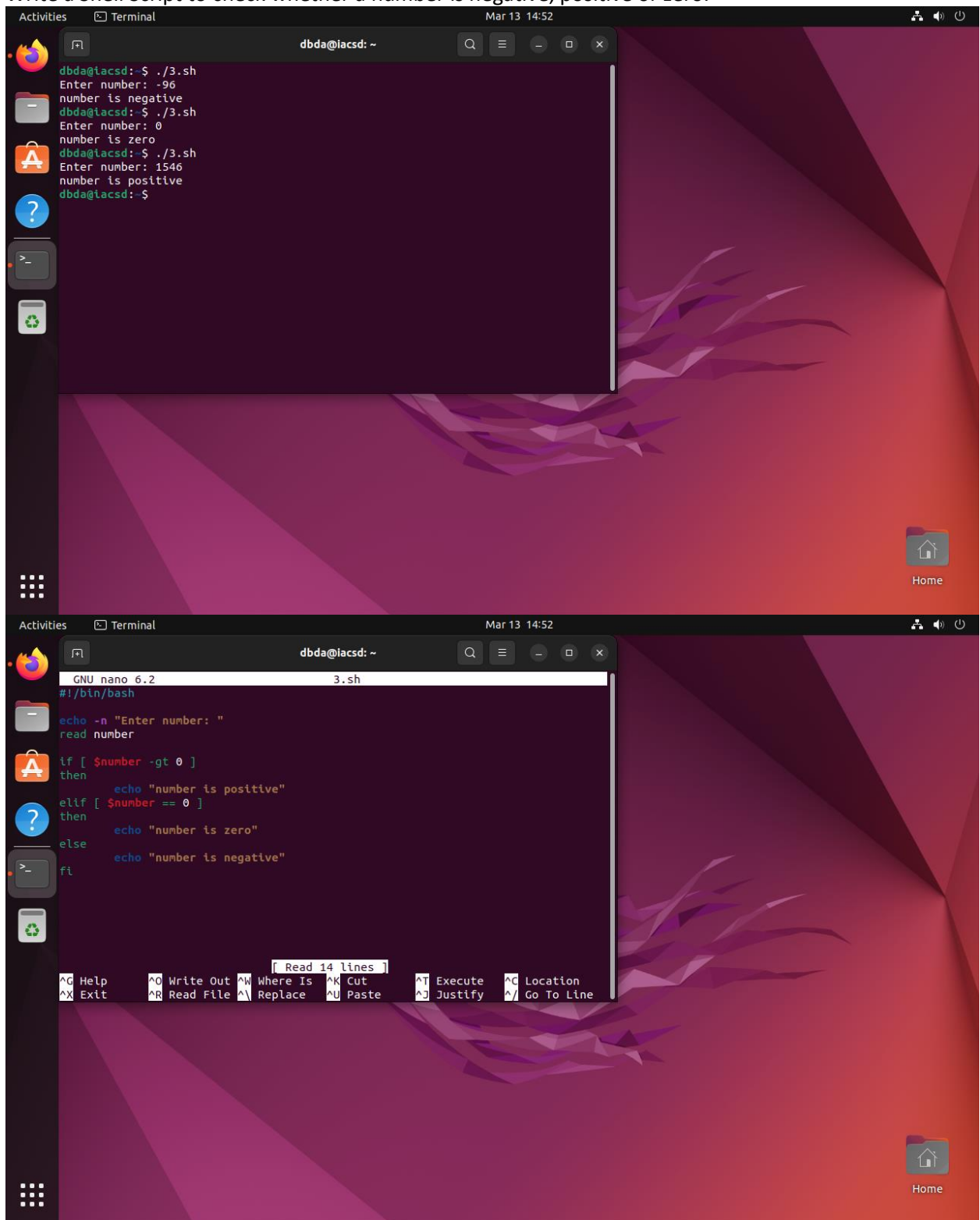
```
#!/bin/bash
echo -n "Enter first number: "
read first
echo -n "Enter second number: "
read second
echo -n "Enter third number: "
read third
if [ $first -gt $second ]
then
    if [ $first -gt $third ]
    then
        echo "$first is maximum"
    else
        echo "$third is maximum"
    fi
elif [ $second -gt $third ]
then
    echo "$second is maximum"
else
    echo "$third is maximum"
fi
```

The bottom status bar of the nano editor indicates 'Wrote 22 lines' and provides keyboard shortcuts for various actions like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line.

Bottom Screenshot: The terminal window shows the execution of the script. The user has entered the command './2.sh'. The output of the script is as follows:

```
dbda@iacsd:~$ ./2.sh
Enter first number: -53
Enter second number: 8
Enter third number: 37
37 is maximum
dbda@iacsd:~$
```

3. Write a Shell Script to check whether a number is negative, positive or zero.



The image consists of two screenshots of a Linux desktop environment, specifically the GNOME interface, showing a terminal window. The desktop background is a purple and red geometric pattern. The terminal window title is "dbda@lacsds: ~".

The top screenshot shows the execution of a shell script named `3.sh`. The user enters the command `./3.sh` three times with different inputs: `-96`, `0`, and `1546`. The script outputs "number is negative", "number is zero", and "number is positive" respectively.

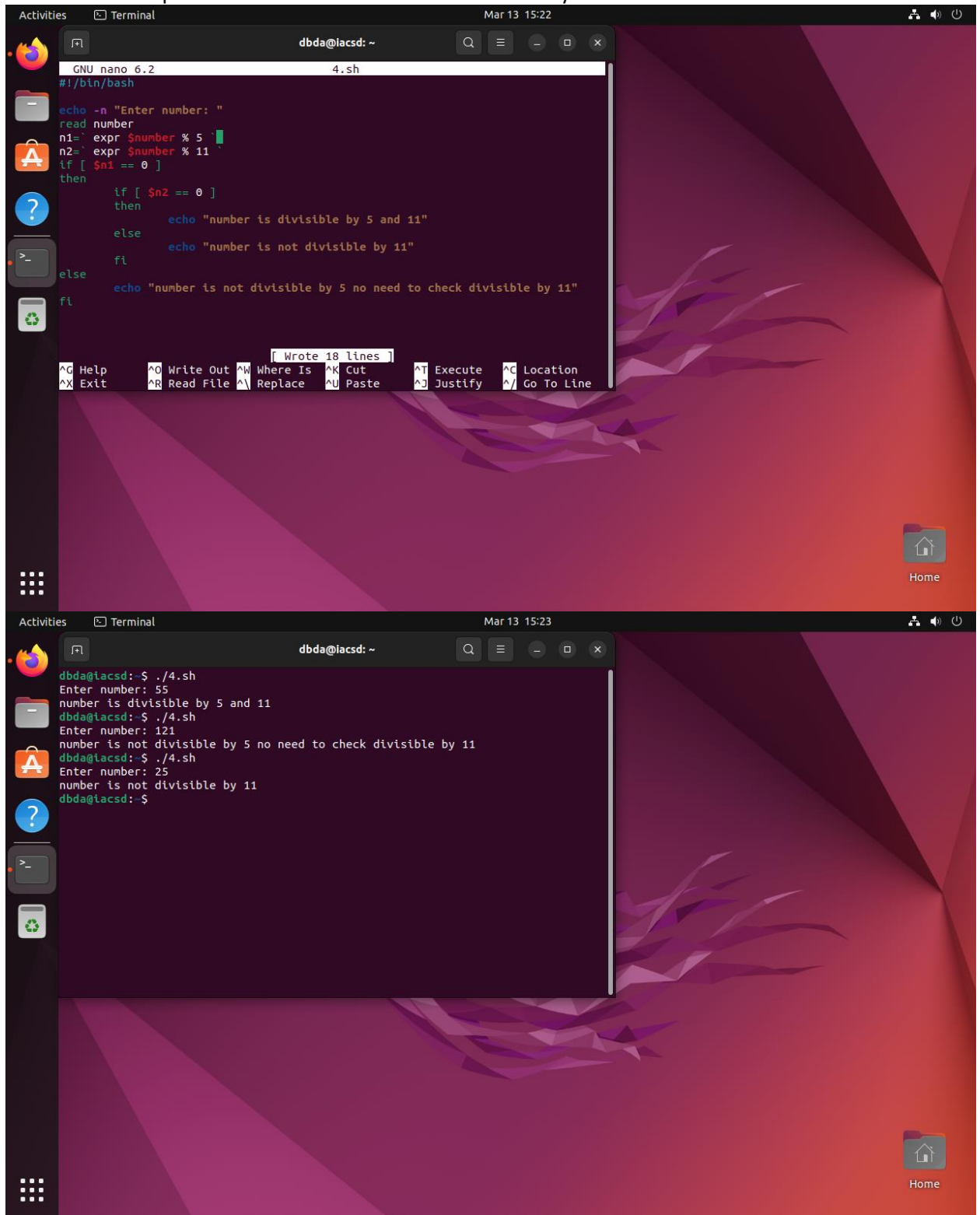
The bottom screenshot shows the same terminal window with the `3.sh` script open in the `nano` text editor. The script content is as follows:

```
#!/bin/bash
echo -n "Enter number: "
read number

if [ $number -gt 0 ]
then
    echo "number is positive"
elif [ $number == 0 ]
then
    echo "number is zero"
else
    echo "number is negative"
fi
```

At the bottom of the terminal window, there is a status bar with various keyboard shortcuts for the `nano` editor, such as `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^M Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.

4. Write a Shell Script to check whether a number is divisible by 5 and 11 or not.



The image consists of two screenshots of a Linux desktop environment, specifically the GNOME desktop, showing a terminal window. The desktop background is a purple and red geometric pattern. The terminal window is titled 'Terminal' and shows the user 'dbda@iacsd: ~'.

The top screenshot shows the user editing a file named '4.sh' using the 'GNU nano 6.2' editor. The script content is as follows:

```
#!/bin/bash
echo -n "Enter number: "
read number
n1=`expr $number % 5`
n2=`expr $number % 11`
if [ $n1 == 0 ]
then
    if [ $n2 == 0 ]
    then
        echo "number is divisible by 5 and 11"
    else
        echo "number is not divisible by 11"
    fi
else
    echo "number is not divisible by 5 no need to check divisible by 11"
fi
```

The bottom screenshot shows the user executing the script with the command `./4.sh`. The output is as follows:

```
dbda@iacsd:~$ ./4.sh
Enter number: 55
number is divisible by 5 and 11
dbda@iacsd:~$ ./4.sh
Enter number: 121
number is not divisible by 5 no need to check divisible by 11
dbda@iacsd:~$ ./4.sh
Enter number: 25
number is not divisible by 11
dbda@iacsd:~$
```

5. Write a Shell Script to check whether a number is even or odd.

The image consists of two screenshots of a Linux desktop environment, specifically the Ubuntu 18.04 LTS version, showing a terminal window. The desktop background is a purple and red geometric pattern. The terminal window is titled 'Terminal' and shows the user 'dbda@lacsds' at the prompt.

The top screenshot shows the user running a script named `5.sh` multiple times with different inputs. The script checks if the input number is even or odd and prints the result. The inputs and outputs are as follows:

```
dbda@lacsds:~$ ./5.sh
Enter number: 5
number is odd
dbda@lacsds:~$ ./5.sh
Enter number: 2
number is even
dbda@lacsds:~$ ./5.sh
Enter number: 0
number is even
dbda@lacsds:~$ ./5.sh
Enter number: 3
number is odd
dbda@lacsds:~$ ./5.sh
Enter number: -95
number is odd
dbda@lacsds:~$ ./5.sh
Enter number: -202
number is even
dbda@lacsds:~$
```

The bottom screenshot shows the user editing the script `5.sh` using the `nano` text editor. The script content is as follows:

```
#!/bin/bash

echo -n "Enter number: "
read number
n=`expr $number % 2`
if [ $n == 0 ]
then
    echo "number is even"
else
    echo "number is odd"
fi
```

The terminal window also displays a status bar at the bottom with various keyboard shortcuts for the `nano` editor, such as `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^U Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.

6. Write a Shell Script to check whether a year is leap year or not.

The image displays two screenshots of a Linux desktop environment, likely Ubuntu, with a purple and orange geometric wallpaper. The desktop features a sidebar on the left with icons for Firefox, Files, App Store, Help, Dash, and a trash can. The top panel shows the 'Activities' button, a 'Terminal' window icon, and the system clock displaying 'Mar 13 15:56'.

The first screenshot shows a terminal window titled 'dbda@lacsds: ~' with a search bar and window controls. Inside the terminal, a script named '6.sh' is being edited using 'GNU nano 6.2'. The script's content is as follows:

```
#!/bin/bash

echo -n "Enter year: "
read year
n=`expr $year % 4 `
m=`expr $year % 100 `
z=`expr $year % 400 `
if [[ $n -eq 0 ]] && [[ $m -ne 0 ]] || [ $z -eq 0 ]
then
    echo "year is leap year"
else
    echo "year is not leap year"
fi
```

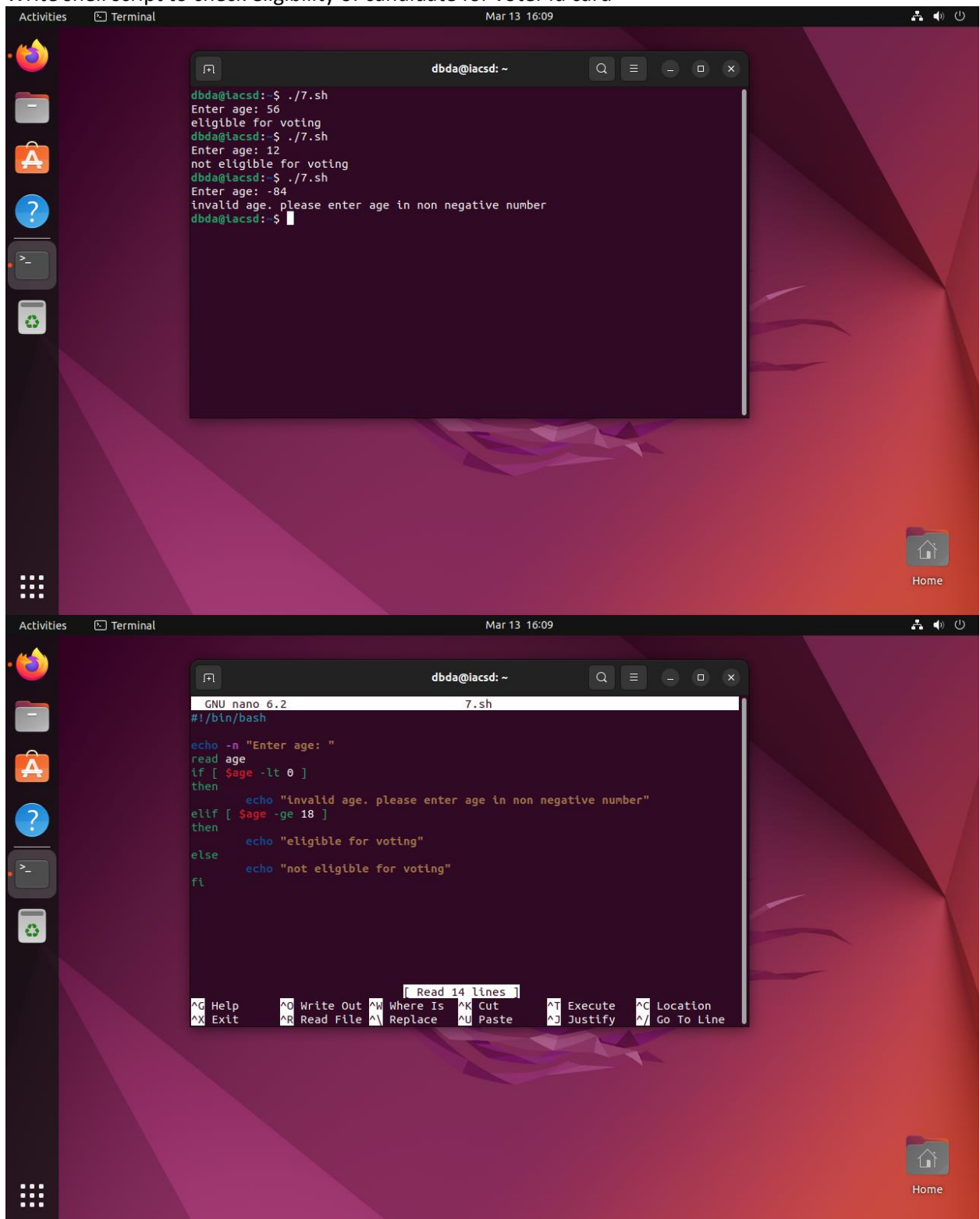
A status bar at the bottom of the terminal window indicates 'Wrote 14 lines'. Below the script, a row of keyboard shortcuts is visible: ^C Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^A Replace, ^U Paste, ^D Justify, and ^_ Go To Line.

The second screenshot shows the same terminal window after the script has been executed. The output of the script is shown for four different years:

```
dbda@lacsds:~$ ./6.sh
Enter year: 2024
year is leap year
dbda@lacsds:~$ ./6.sh
Enter year: 2005
year is not leap year
dbda@lacsds:~$ ./6.sh
Enter year: 2000
year is leap year
dbda@lacsds:~$ ./6.sh
Enter year: 1996
year is leap year
dbda@lacsds:~$
```

The system clock in the top panel now displays 'Mar 13 15:57'.

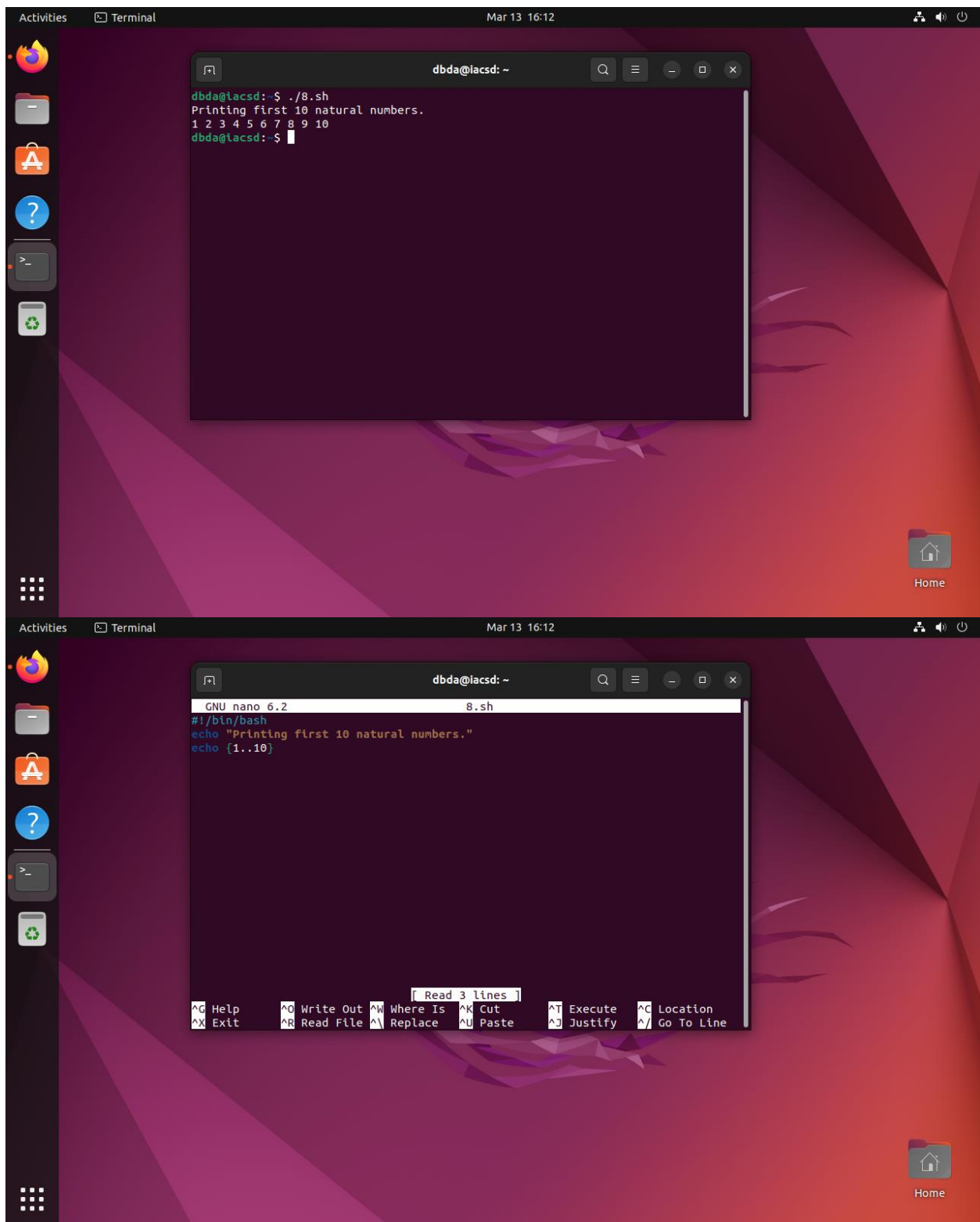
7. Write shell script to check eligibility of candidate for voter id card



8. Shell Script to display the first 10 natural numbers.

Expected Output :

1 2 3 4 5 6 7 8 9 10



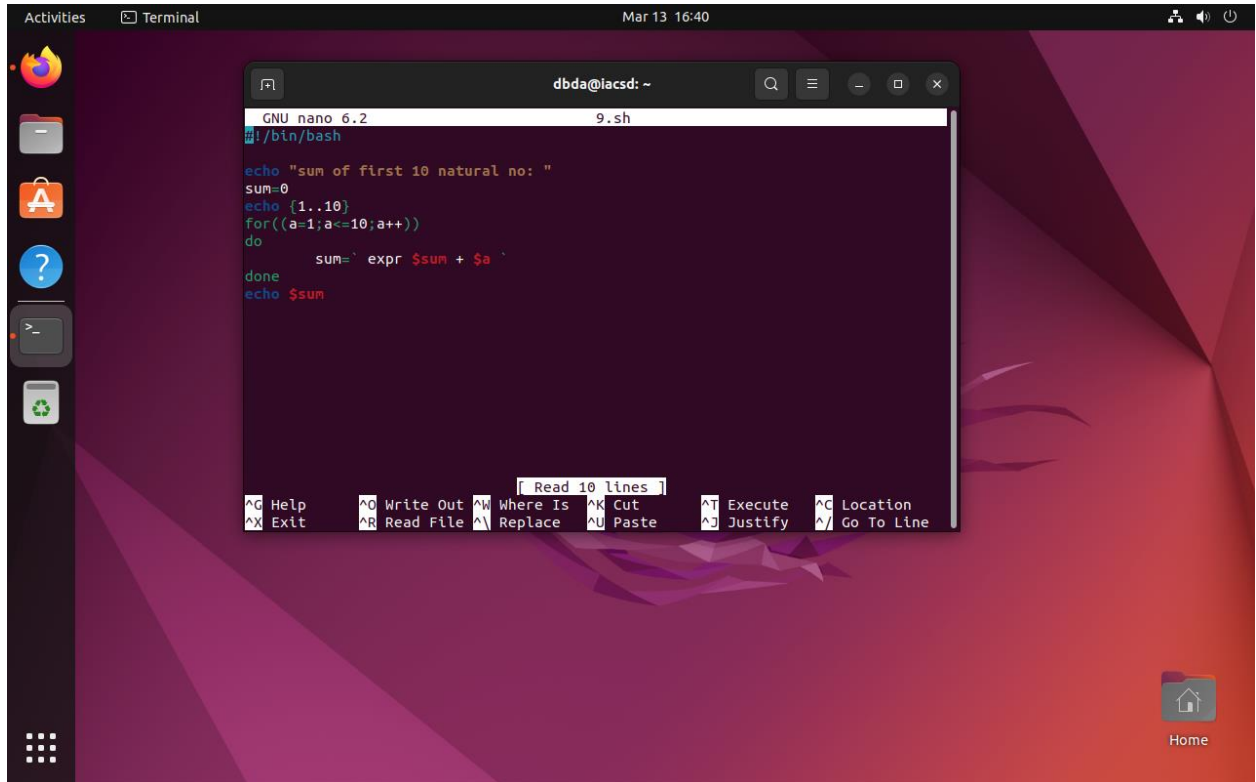
9. Shell Script to compute the sum of the first 10 natural numbers.

Expected Output :

The first 10 natural number is :

1 2 3 4 5 6 7 8 9 10

The Sum is : 55



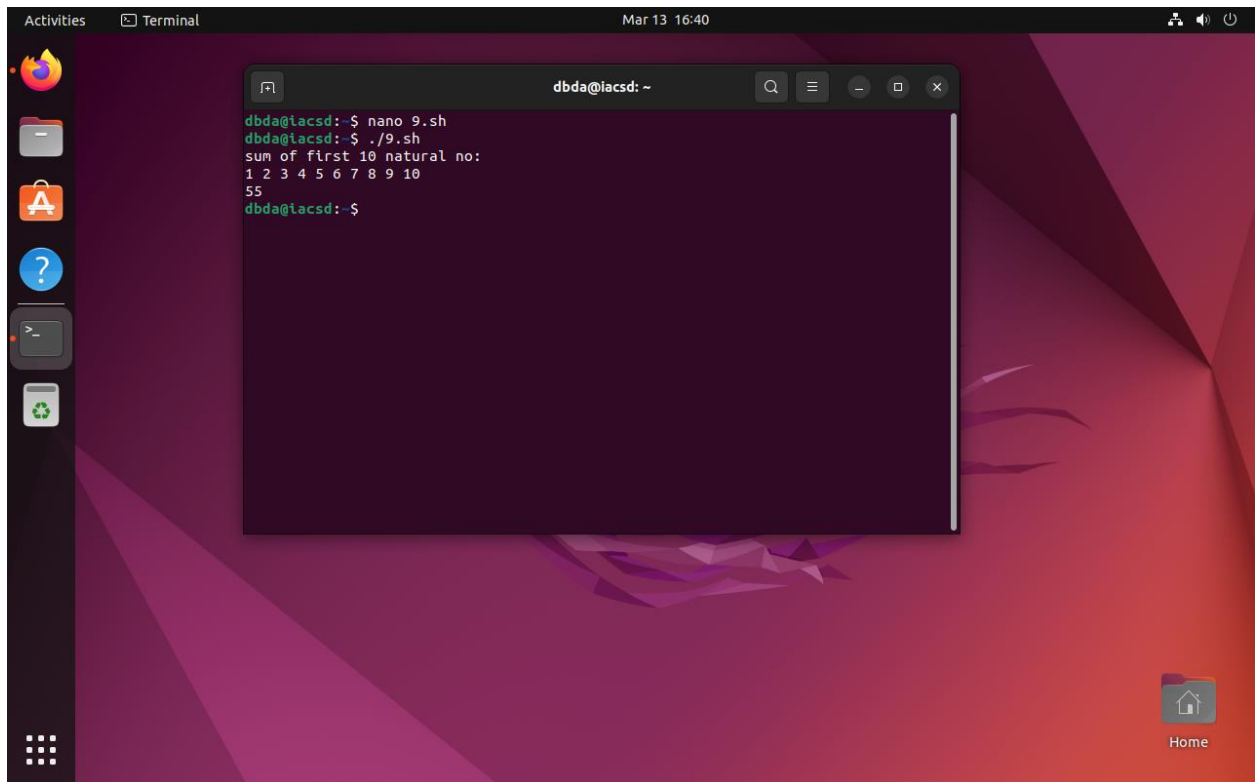
The screenshot shows a Linux desktop with a dark theme. A terminal window titled 'dbda@lacs: ~' is open, displaying a shell script in nano 6.2. The script calculates the sum of the first 10 natural numbers. The desktop background is a red and purple geometric pattern. A 'Home' folder icon is visible in the bottom right corner.

```
dbda@lacs: ~
GNU nano 6.2 9.sh
#!/bin/bash

echo "sum of first 10 natural no: "
sum=0
echo {1..10}
for((a=1;a<=10;a++))
do
    sum=`expr $sum + $a `
done
echo $sum
```

Read 10 lines

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line



The screenshot shows a Linux desktop with a dark purple and red background. A terminal window is open, displaying the following commands and output:

```
dbda@lacs: ~  
dbda@lacs: $ nano 9.sh  
dbda@lacs: $ ./9.sh  
sum of first 10 natural no:  
1 2 3 4 5 6 7 8 9 10  
55  
dbda@lacs: $
```

The terminal window has a title bar that says "dbda@lacs: ~". The desktop has a sidebar on the left with icons for Activities, Terminal, Files, Applications, Help, and a Dash icon. A "Home" icon is visible in the bottom right corner.

10. Shell Script to display n terms of natural numbers and their sum.

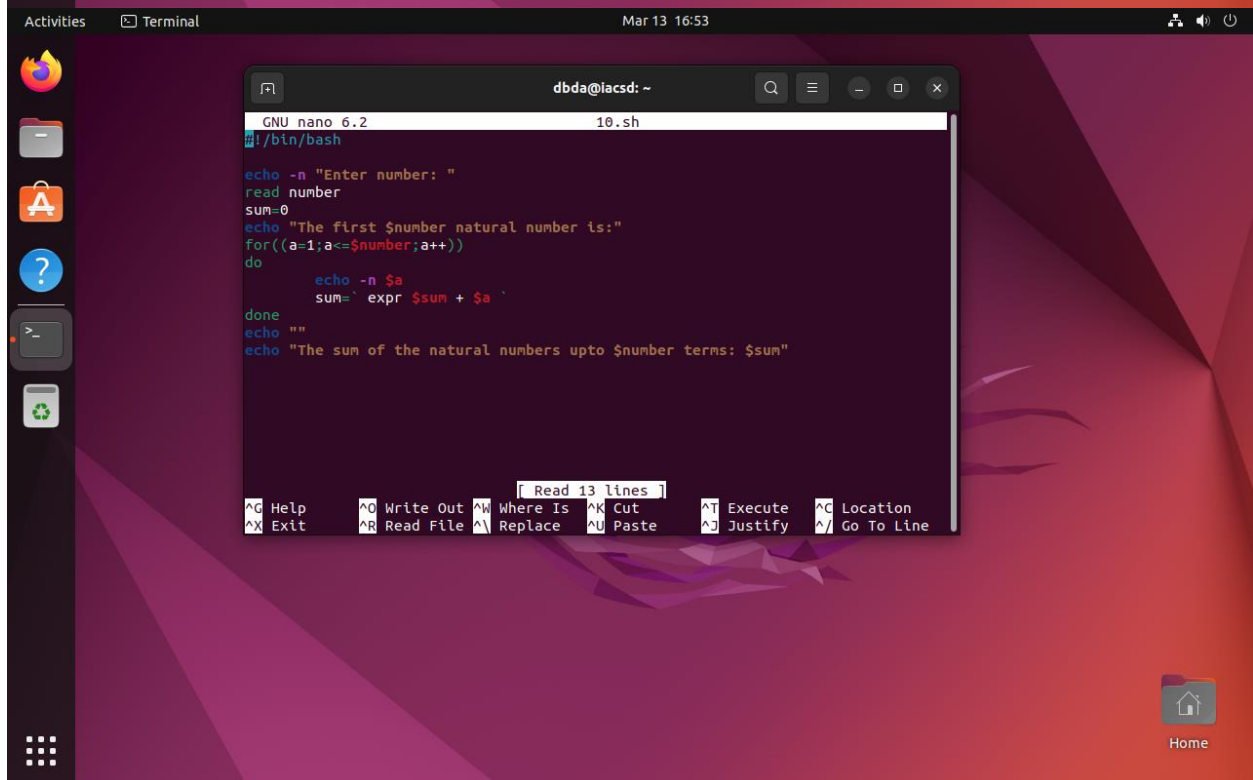
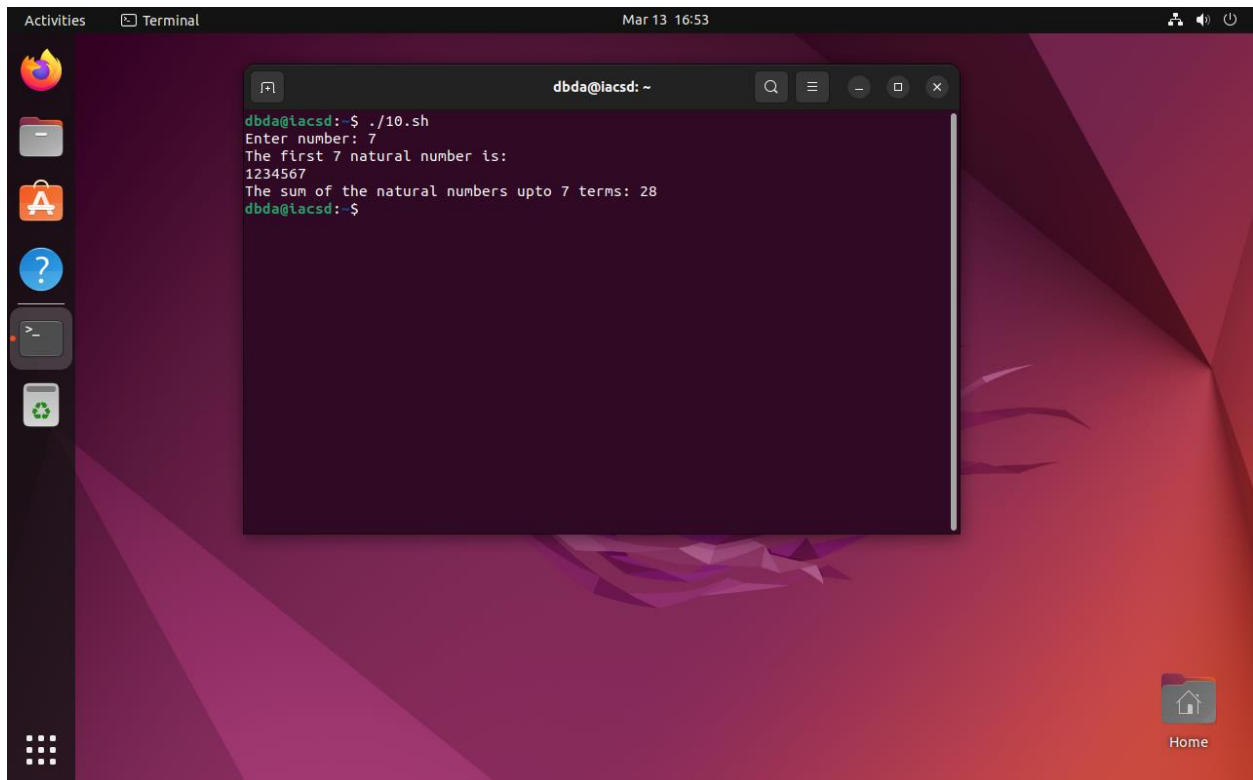
Test Data : 7

Expected Output :

The first 7 natural number is :

1 2 3 4 5 6 7

The Sum of Natural Number upto 7 terms : 28



11. Shell Script to read 10 numbers from the keyboard and find their sum and average.

Test Data :

Input the 10 numbers :

Number-1 :2

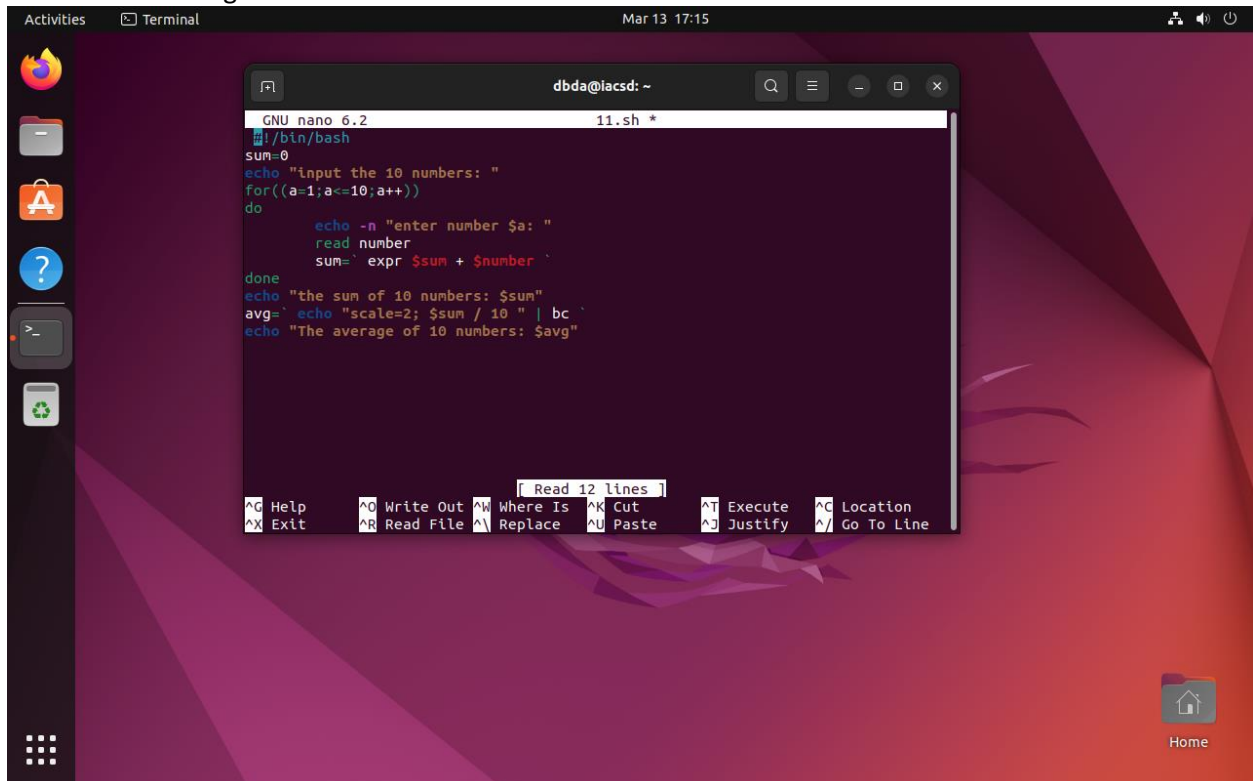
...

Number-10 :2

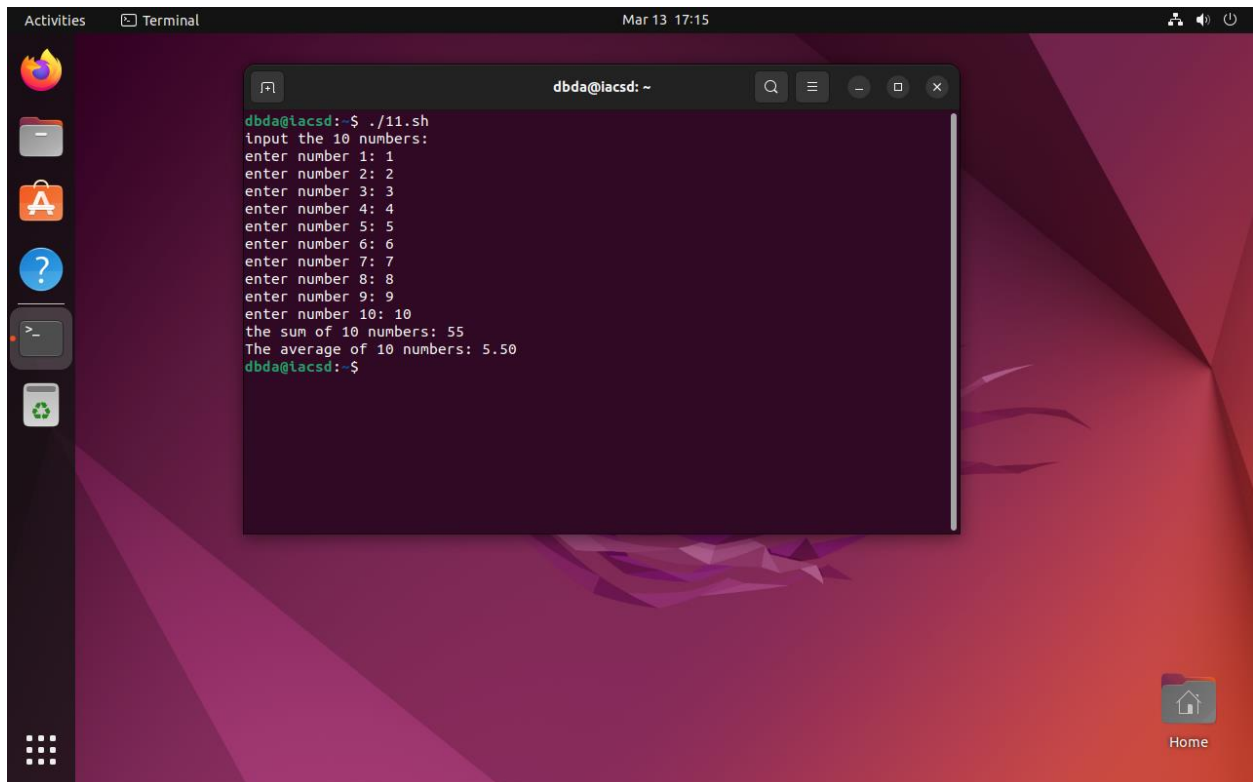
Expected Output :

The sum of 10 no is : 55

The Average is : 5.500000



```
dbda@lacsds: ~  
GNU nano 6.2 11.sh *  
#!/bin/bash  
sum=0  
echo "input the 10 numbers: "  
for((a=1;a<=10;a++))  
do  
    echo -n "enter number $a: "  
    read number  
    sum=`expr $sum + $number`  
done  
echo "the sum of 10 numbers: $sum"  
avg=`echo "scale=2; $sum / 10 " | bc`  
echo "The average of 10 numbers: $avg"  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^M Replace   ^U Paste     ^D Justify   ^_ Go To Line
```



The screenshot shows a Linux desktop with a dark purple and red background. A terminal window is open, displaying the execution of a shell script. The script prompts for 10 numbers, calculates their sum (55) and average (5.50). The desktop has a sidebar with application icons (Firefox, Files, App Store, Help, Terminal, Recycle Bin) and a 'Home' button in the bottom right corner. The top bar shows 'Activities', 'Terminal', and the date/time 'Mar 13 17:15'.

```
dbda@lacsda: ~  
dbda@lacsda: $ ./11.sh  
input the 10 numbers:  
enter number 1: 1  
enter number 2: 2  
enter number 3: 3  
enter number 4: 4  
enter number 5: 5  
enter number 6: 6  
enter number 7: 7  
enter number 8: 8  
enter number 9: 9  
enter number 10: 10  
the sum of 10 numbers: 55  
The average of 10 numbers: 5.50  
dbda@lacsda: $
```

12. Shell Script to display the cube of the number up to an integer.

Test Data :

Input number of terms : 5

Expected Output :

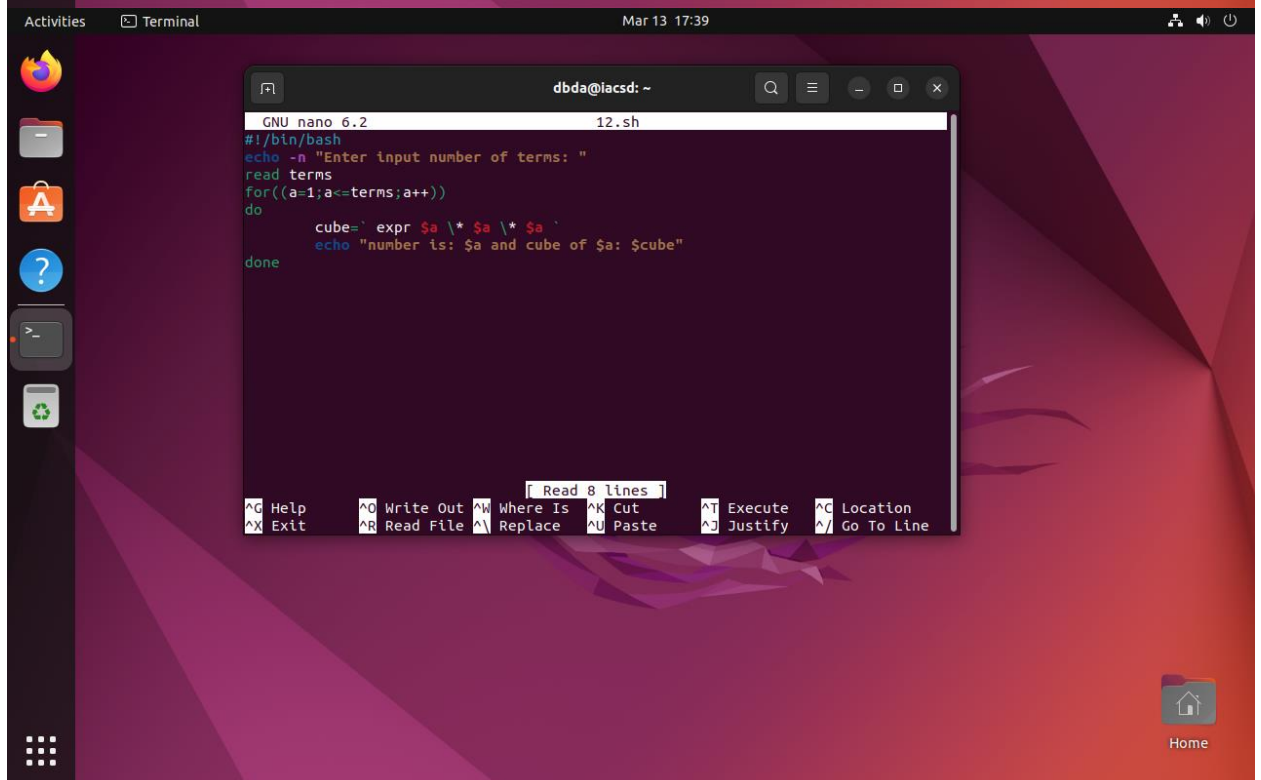
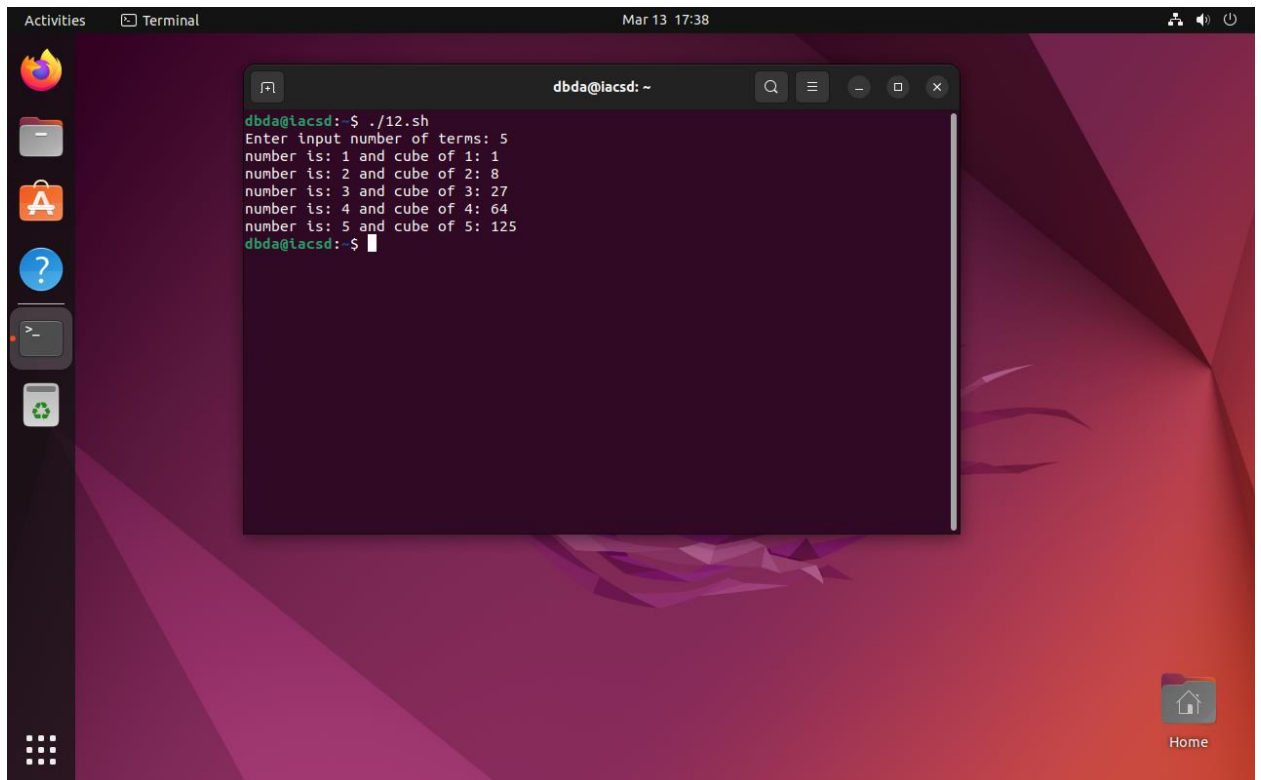
Number is : 1 and cube of the 1 is :1

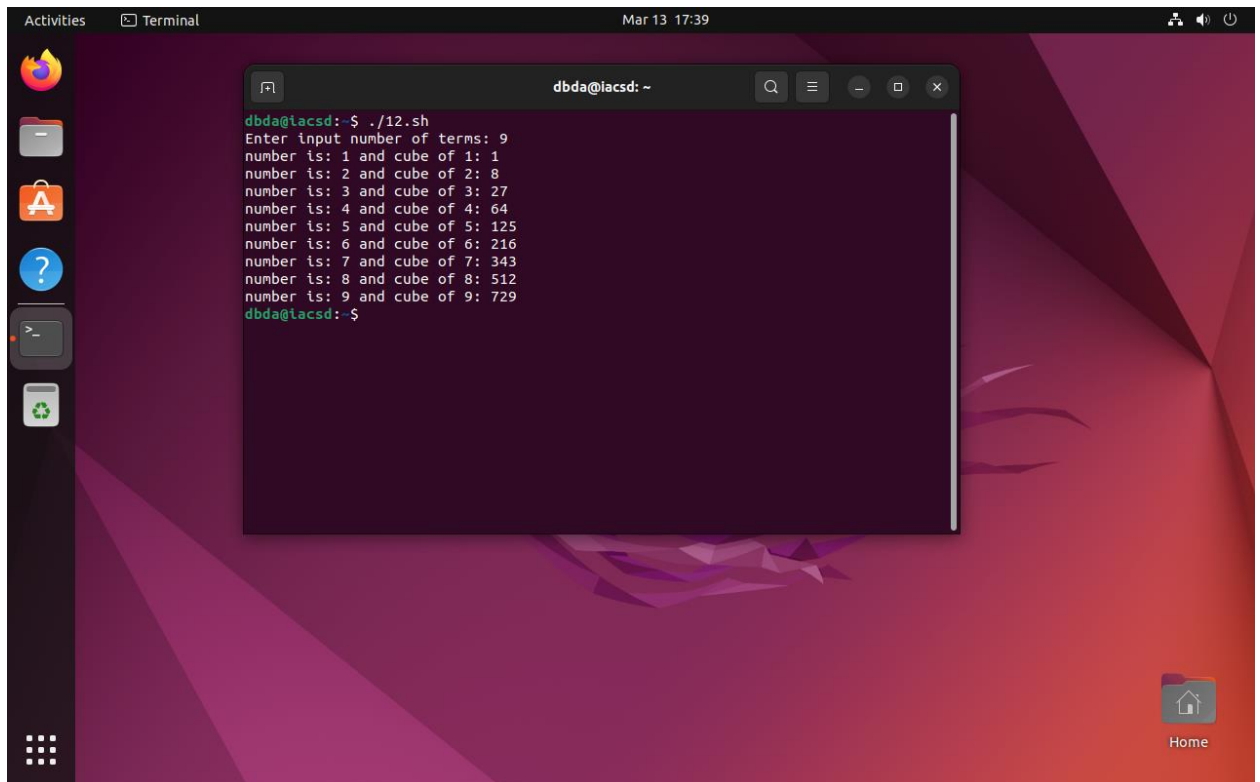
Number is : 2 and cube of the 2 is :8

Number is : 3 and cube of the 3 is :27

Number is : 4 and cube of the 4 is :64

Number is : 5 and cube of the 5 is :125





13. Shell Script to display the multiplication table for a given integer.

Test Data :

Input the number (Table to be calculated) : 15

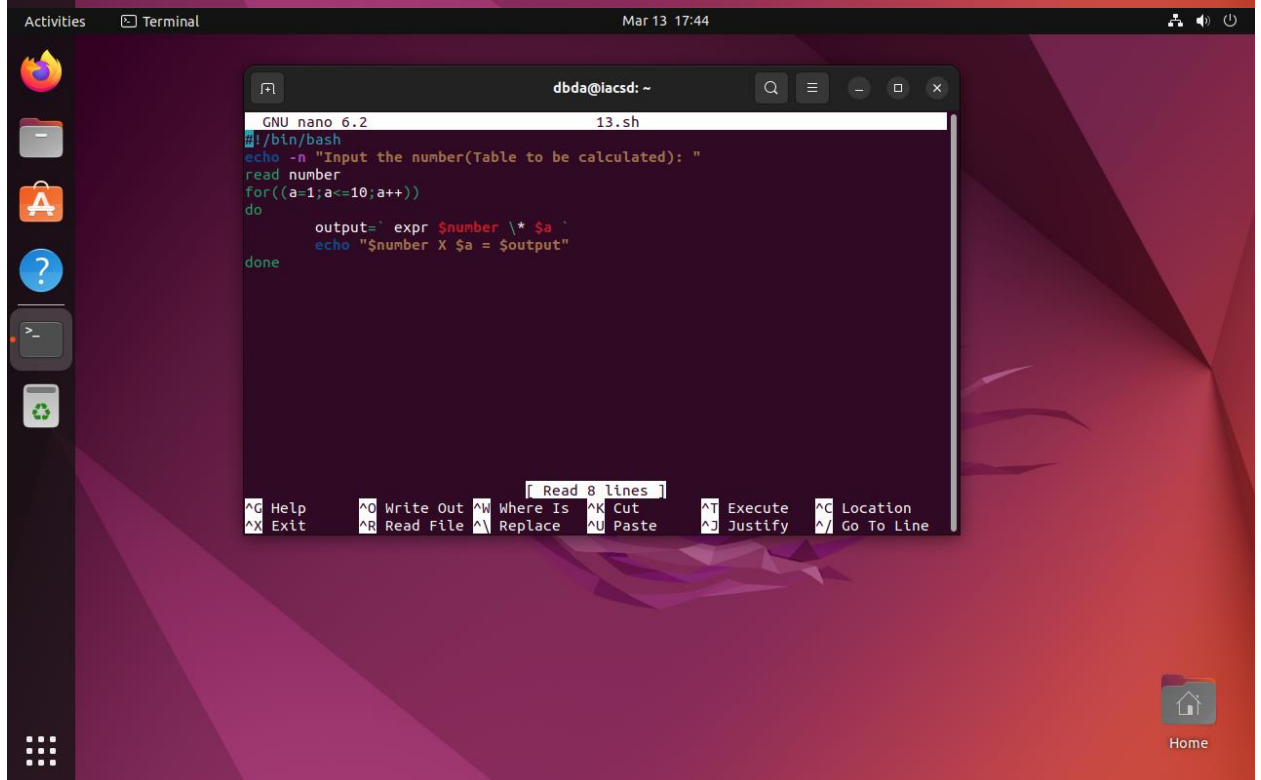
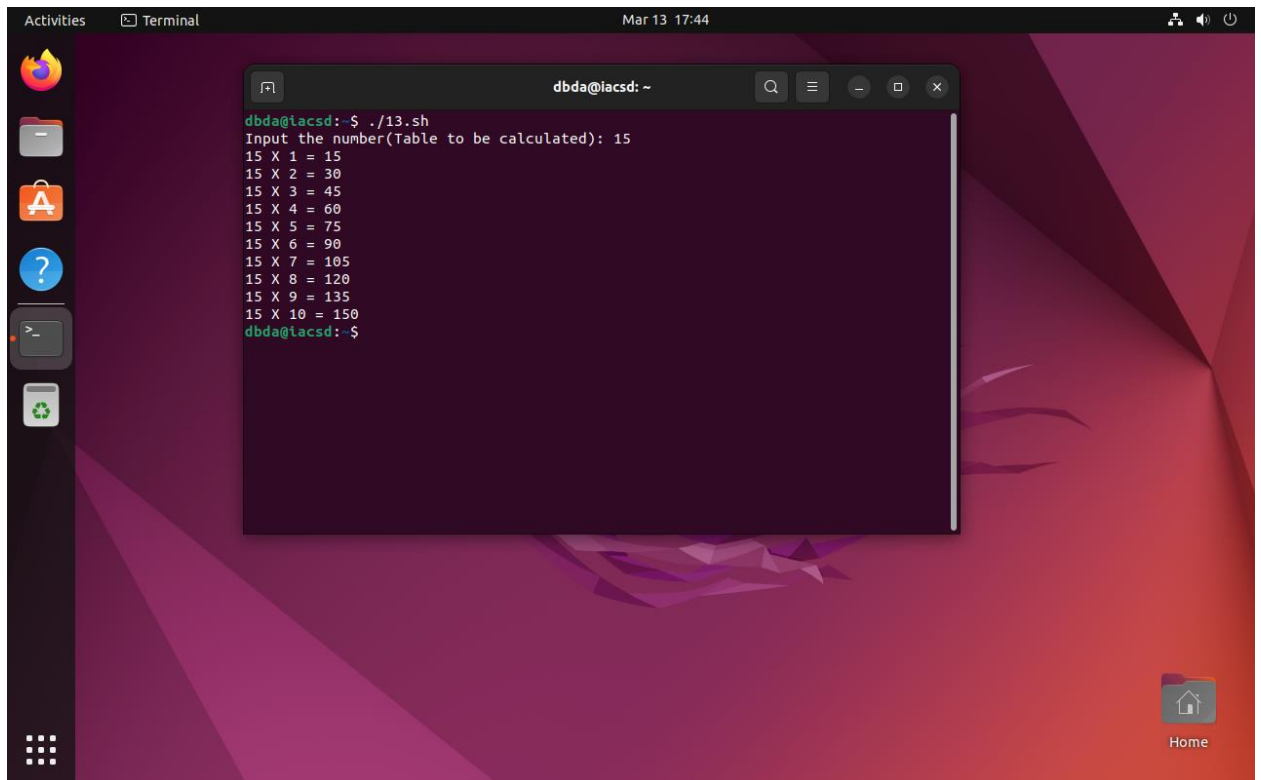
Expected Output :

15 X 1 = 15

...

...

15 X 10 = 150



14. Shell Script to display the multiplier table vertically from 1 to n.

Test Data :

Input upto the table number starting from 1 : 8


Expected Output :

Multiplication table from 1 to 8

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

...

1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80



The screenshot shows a terminal window with the nano 6.2 editor open. The script is named 14.sh and is designed to generate a multiplication table. It prompts the user for a starting number and an ending number. The script uses nested loops: an outer loop for the multiplier (a) and an inner loop for the multiplicand (b). The output is formatted as 'b X a = result'. The terminal window title is 'dbda@lacsds: ~' and the date/time is 'Mar 13 18:18'. The bottom of the window shows a menu bar with various keyboard shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^U Undo, ^A Set Mark, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^D Justify, ^_ Go To Line, ^E Redo, and ^C Copy.

```
GNU nano 6.2 14.sh
#!/bin/bash
echo -n "Input upto the table no starting from : "
read start
echo -n "Input upto the table no starting from : "
read end
for((a=$start;a<=10;a++))
do
    for((b=$start;b<=$end;b++))
    do
        mul=`expr $b \* $a `
        echo -n "$b X $a = $mul, "
    done
    echo ""
done
```

```
Activities Terminal Mar 13 18:19 dbda@lacsds: ~
dbda@lacsds:~$ nano 14.sh
dbda@lacsds:~$ ./14.sh
Input upto the table no starting from : 1
Input upto the table no starting from : 8
1 X 1 = 1, 2 X 1 = 2, 3 X 1 = 3, 4 X 1 = 4, 5 X 1 = 5, 6 X 1 = 6, 7 X 1 = 7, 8 X 1 = 8,
1 X 2 = 2, 2 X 2 = 4, 3 X 2 = 6, 4 X 2 = 8, 5 X 2 = 10, 6 X 2 = 12, 7 X 2 = 14, 8 X 2 = 16,
1 X 3 = 3, 2 X 3 = 6, 3 X 3 = 9, 4 X 3 = 12, 5 X 3 = 15, 6 X 3 = 18, 7 X 3 = 21, 8 X 3 = 24,
1 X 4 = 4, 2 X 4 = 8, 3 X 4 = 12, 4 X 4 = 16, 5 X 4 = 20, 6 X 4 = 24, 7 X 4 = 28, 8 X 4 = 32,
1 X 5 = 5, 2 X 5 = 10, 3 X 5 = 15, 4 X 5 = 20, 5 X 5 = 25, 6 X 5 = 30, 7 X 5 = 35, 8 X 5 = 40,
1 X 6 = 6, 2 X 6 = 12, 3 X 6 = 18, 4 X 6 = 24, 5 X 6 = 30, 6 X 6 = 36, 7 X 6 = 42, 8 X 6 = 48,
1 X 7 = 7, 2 X 7 = 14, 3 X 7 = 21, 4 X 7 = 28, 5 X 7 = 35, 6 X 7 = 42, 7 X 7 = 49, 8 X 7 = 56,
1 X 8 = 8, 2 X 8 = 16, 3 X 8 = 24, 4 X 8 = 32, 5 X 8 = 40, 6 X 8 = 48, 7 X 8 = 56, 8 X 8 = 64,
1 X 9 = 9, 2 X 9 = 18, 3 X 9 = 27, 4 X 9 = 36, 5 X 9 = 45, 6 X 9 = 54, 7 X 9 = 63, 8 X 9 = 72,
1 X 10 = 10, 2 X 10 = 20, 3 X 10 = 30, 4 X 10 = 40, 5 X 10 = 50, 6 X 10 = 60, 7 X 10 = 70, 8 X 10 = 80,
dbda@lacsds:~$
```

15. Shell Script to display the n terms of odd natural numbers and their sum.

Test Data

Input number of terms : 10

Expected Output :

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

