

1. vi

```
Thunderbird Mail user@user-VirtualBox:~/Desktop$ vi raman.sh
user@user-VirtualBox:~/Desktop$ vi biod.sh
user@user-VirtualBox:~/Desktop$ cat vi biod.sh
cat: vi: No such file or directory

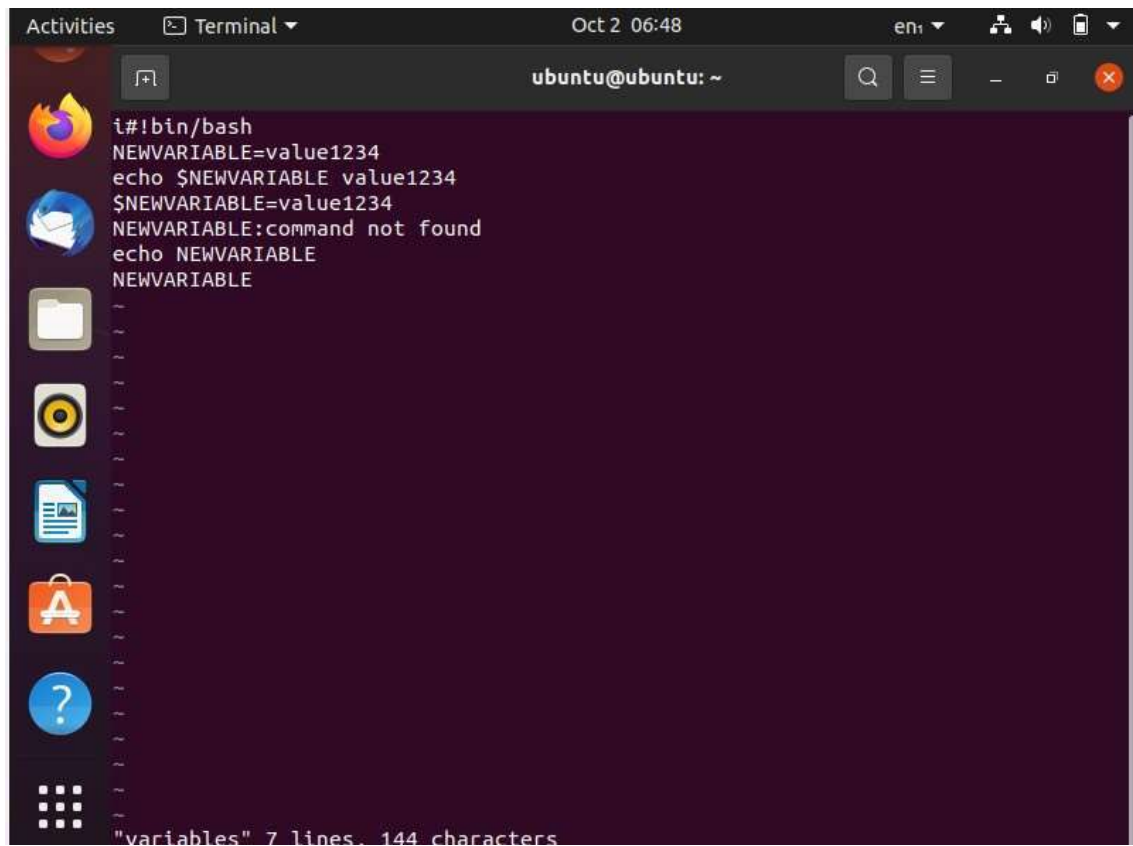
echo enter your name
read sujithj
echo enter college name
read amal jyothi college
echo Name:$name
echo college:$amal jyothi college
```

```
user@user-VirtualBox:~/Desktop$ ls -l biod.sh
-rw-rw-r-- 1 user user 135 Oct  2 22:44 biod.sh
user@user-VirtualBox:~/Desktop$ chmod +x biod.sh
user@user-VirtualBox:~/Desktop$ ls -l biod.sh
-rwxrwxr-x 1 user user 135 Oct  2 22:44 biod.sh
user@user-VirtualBox:~/Desktop$ ./biod.sh
enter your name
s Terminal
enter college name
amal jyothi college
Name:
college:amal jyothi college
```

```
ubuntu@ubuntu:~$ ./profile.sh
./profile.sh: line 1: i#!/bin/bash: No such file or directory
Enter your name
sruthy
Enter college name
Amal Jyothi College
Name:
college:Amal Jyothi College
ubuntu@ubuntu:~$ vi profile.sh
ubuntu@ubuntu:~$ ./profile.sh
./profile.sh: line 1: i#!/bin/bash: No such file or directory
Enter your name
Sruthy
Enter college name
Amal Jyothi College
Name:
college:Amal Jyothi College
ubuntu@ubuntu:~$ vi profile.sh
ubuntu@ubuntu:~$
```

2. Write a shell script to set a value for a variable and display it on command line interface.

```
Activities Terminal Oct 2 06:45 en1
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi variables
ubuntu@ubuntu:~$ vi variables
ubuntu@ubuntu:~$ cat variables.sh
cat: variables.sh: No such file or directory
ubuntu@ubuntu:~$ cat variables
i#!/bin/bash
NEWVARIABLE=value1234
echo $NEWVARIABLE value1234
$NEWVARIABLE=value1234
NEWVARIABLE:command not found
echo NEWVARIABLE
NEWVARIABLE
ubuntu@ubuntu:~$ ls -l variables
-rw-rw-r-- 1 ubuntu ubuntu 144 Oct 2 06:42 variables
ubuntu@ubuntu:~$ chmod +x variables
ubuntu@ubuntu:~$ ls -l variables
-rwxrwxr-x 1 ubuntu ubuntu 144 Oct 2 06:42 variables
ubuntu@ubuntu:~$ ./variables
./variables: line 1: i#!/bin/bash: No such file or directory
value1234 value1234
./variables: line 4: value1234=value1234: command not found
./variables: line 5: NEWVARIABLE:command: command not found
NEWVARIABLE
./variables: line 7: NEWVARIABLE: command not found
ubuntu@ubuntu:~$
```

A screenshot of a Linux terminal window. The window title bar shows 'Activities', 'Terminal', and the date 'Oct 2 06:48'. The terminal prompt is 'ubuntu@ubuntu: ~'. The user has entered the following commands: 'i#!/bin/bash', 'NEWVARIABLE=value1234', 'echo \$NEWVARIABLE value1234', '\$NEWVARIABLE=value1234', 'NEWVARIABLE:command not found', 'echo NEWVARIABLE', and 'NEWVARIABLE'. The output shows the variable's value and an error message. The terminal window has a dark purple background and a sidebar with application icons on the left.

```
i#!/bin/bash
NEWVARIABLE=value1234
echo $NEWVARIABLE value1234
$NEWVARIABLE=value1234
NEWVARIABLE:command not found
echo NEWVARIABLE
NEWVARIABLE
```

3. Write a shell script to perform addition, substraction, multiplication, division with two numbers that is accepted from user.

4. Write a shell script to check the value of a given number and display whether the number is found or not.

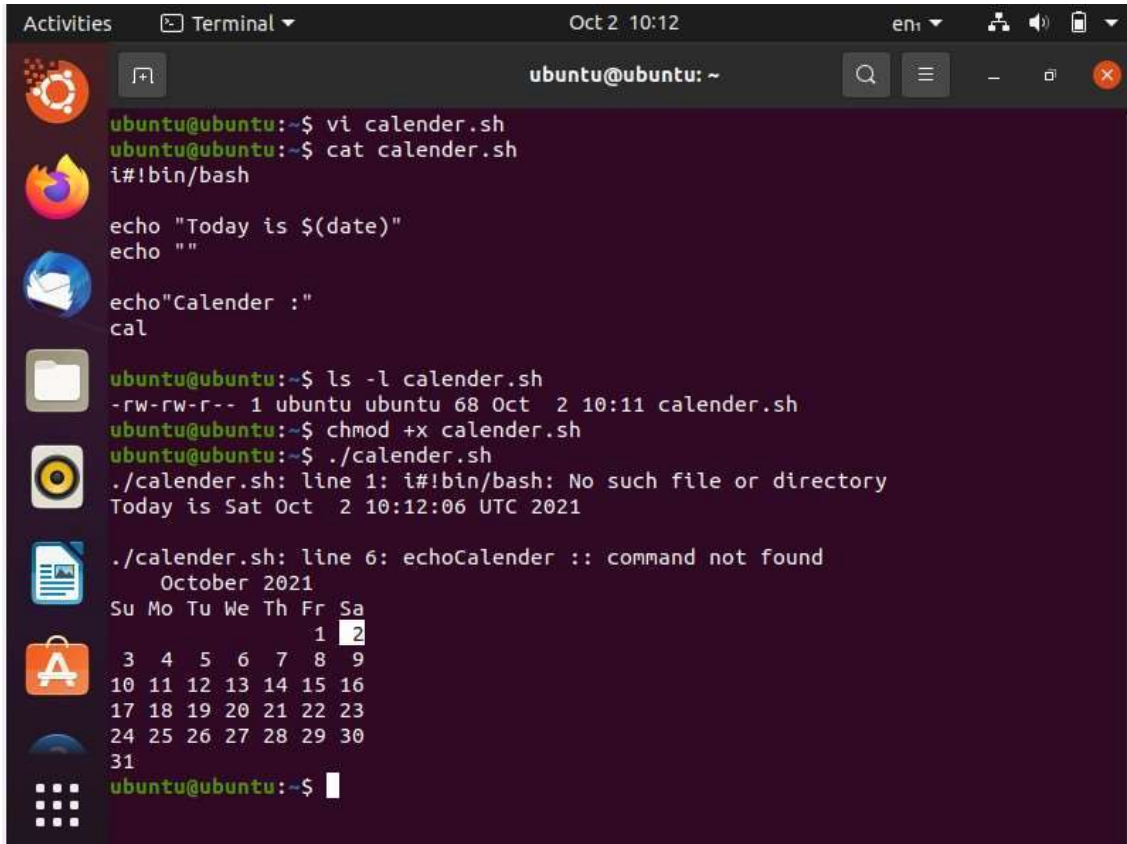
The screenshot shows a terminal window titled "Terminal" with the prompt "ubuntu@ubuntu: ~". The script being executed is as follows:

```
i#!/bin/bash
echo "Enter a number"
read num
i=1
ans=0
while [ $i -le $((num / 2)) ]
do
if [[ $((num%i)) -eq 0 ]]
then
ans=$((ans + i))
fi
done
if [ $num -eq $ans ]
then
echo "$num is perfect"
else
echo "$num is NOT perfect"
fi
```

The output of the script is shown at the bottom: "numbers.sh" 18 lines, 229 characters.


```
Activities Terminal Oct 2 10:05 en: ubuntu@ubuntu: ~
echo "Enter a number"
read no
i=1
ans=0
while [ $i -le $((no / 2))]
do
if [[ $((no%i)) -eq 0 ]]
then ans=$((ans + i))
fi
i=$((i + 1))
done
if [ $no -eq $ans ]
then
echo "$no is perfect"
else
echo "no is not perfect"
fi
ubuntu@ubuntu:~$ ls -l number.sh
-rwxrwxr-x 1 ubuntu ubuntu 233 Oct 2 09:42 number.sh
ubuntu@ubuntu:~$ chmod +x number.sh
ubuntu@ubuntu:~$ ls -l number.sh
-rwxrwxr-x 1 ubuntu ubuntu 233 Oct 2 09:42 number.sh
ubuntu@ubuntu:~$ ./number.sh
./number.sh: line 1: i#!/bin/bash: No such file or directory
Enter a number
7
./number.sh: line 6: [: missing `]'
no is not perfect
ubuntu@ubuntu:~$
```

5. Write a shell script to display current date, calendar.



```
Activities Terminal Oct 2 10:12 en: ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi calender.sh
ubuntu@ubuntu:~$ cat calender.sh
i#!/bin/bash

echo "Today is $(date)"
echo ""

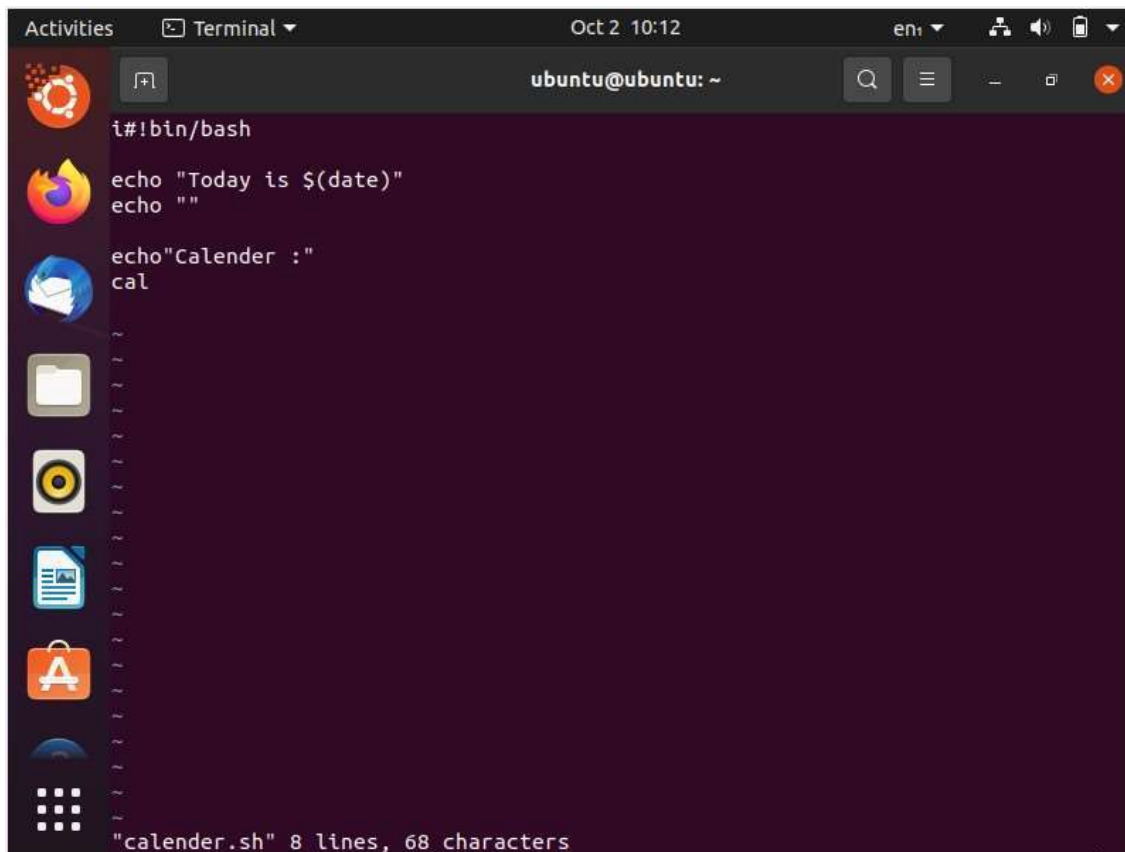
echo"Calender : "
cal

ubuntu@ubuntu:~$ ls -l calender.sh
-rw-rw-r-- 1 ubuntu ubuntu 68 Oct  2 10:11 calender.sh
ubuntu@ubuntu:~$ chmod +x calender.sh
ubuntu@ubuntu:~$ ./calender.sh
./calender.sh: line 1: i#!/bin/bash: No such file or directory
Today is Sat Oct  2 10:12:06 UTC 2021

./calender.sh: line 6: echoCalender :: command not found
October 2021
Su Mo Tu We Th Fr Sa
      1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
ubuntu@ubuntu:~$
```

The image shows a terminal window with a dark purple background. The window title is "Terminal" and the date/time is "Oct 2 10:12". The user is logged in as "ubuntu" on a machine named "ubuntu". The terminal shows the following commands and output:

- `vi calender.sh`: Opens the file `calender.sh` in the `vi` editor.
- `cat calender.sh`: Displays the contents of `calender.sh`. The script starts with `i#!/bin/bash` (note the extra 'i'), followed by `echo "Today is $(date)"`, `echo ""`, `echo"Calender : "` (note the missing space), and `cal`.
- `ls -l calender.sh`: Shows the file permissions and details: `-rw-rw-r-- 1 ubuntu ubuntu 68 Oct 2 10:11 calender.sh`.
- `chmod +x calender.sh`: Makes the script executable.
- `./calender.sh`: Executes the script. The output is:
 - `./calender.sh: line 1: i#!/bin/bash: No such file or directory`
 - `Today is Sat Oct 2 10:12:06 UTC 2021`
 - `./calender.sh: line 6: echoCalender :: command not found`
 - A calendar for October 2021, showing days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and dates (1-31). The date 2 is highlighted.

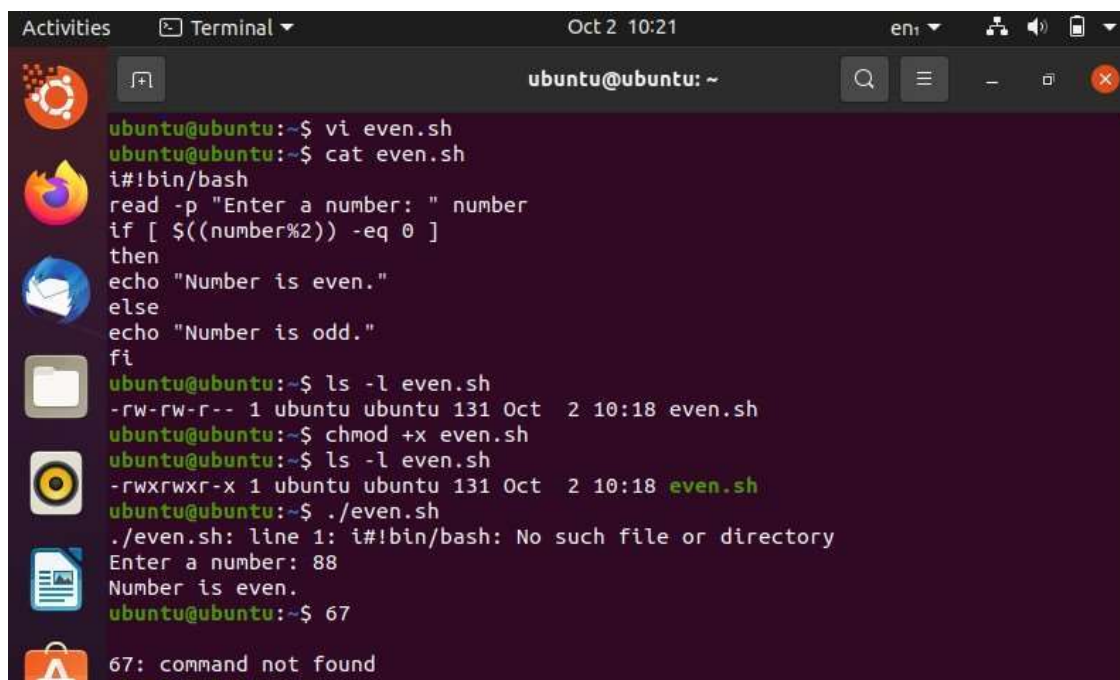


A terminal window titled "Terminal" with the date "Oct 2 10:12" and language "en". The prompt is "ubuntu@ubuntu: ~". The user enters the following commands:

```
i#!bin/bash
echo "Today is $(date)"
echo ""
echo"Calender :"
cal
```

The terminal shows the output of the 'cal' command, which displays a calendar for the current month. At the bottom, it says "calender.sh" 8 lines, 68 characters.

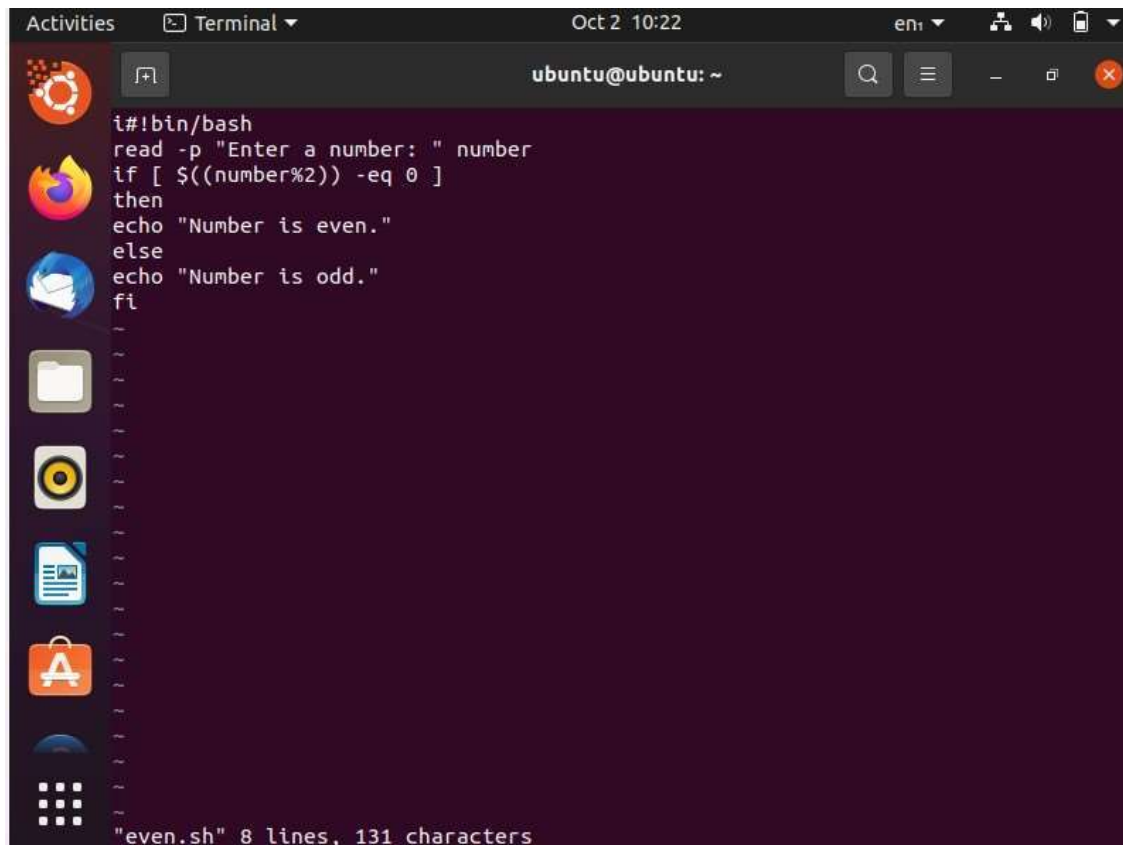
6. Write a shell script to check a number is even or odd.



A terminal window titled "Terminal" with the date "Oct 2 10:21" and language "en". The prompt is "ubuntu@ubuntu: ~". The user enters the following commands:

```
ubuntu@ubuntu:~$ vi even.sh
ubuntu@ubuntu:~$ cat even.sh
i#!bin/bash
read -p "Enter a number: " number
if [ $(number%2) -eq 0 ]
then
echo "Number is even."
else
echo "Number is odd."
fi
ubuntu@ubuntu:~$ ls -l even.sh
-rw-rw-r-- 1 ubuntu ubuntu 131 Oct  2 10:18 even.sh
ubuntu@ubuntu:~$ chmod +x even.sh
ubuntu@ubuntu:~$ ls -l even.sh
-rwxrwxr-x 1 ubuntu ubuntu 131 Oct  2 10:18 even.sh
ubuntu@ubuntu:~$ ./even.sh
./even.sh: line 1: i#!bin/bash: No such file or directory
Enter a number: 88
Number is even.
ubuntu@ubuntu:~$ 67
67: command not found
```

The terminal shows the output of the 'cat' command, which displays the contents of the 'even.sh' script. The user then runs 'ls -l even.sh' and 'chmod +x even.sh' to make the script executable. Finally, the user runs './even.sh' and enters the number 88, which results in the output "Number is even.".

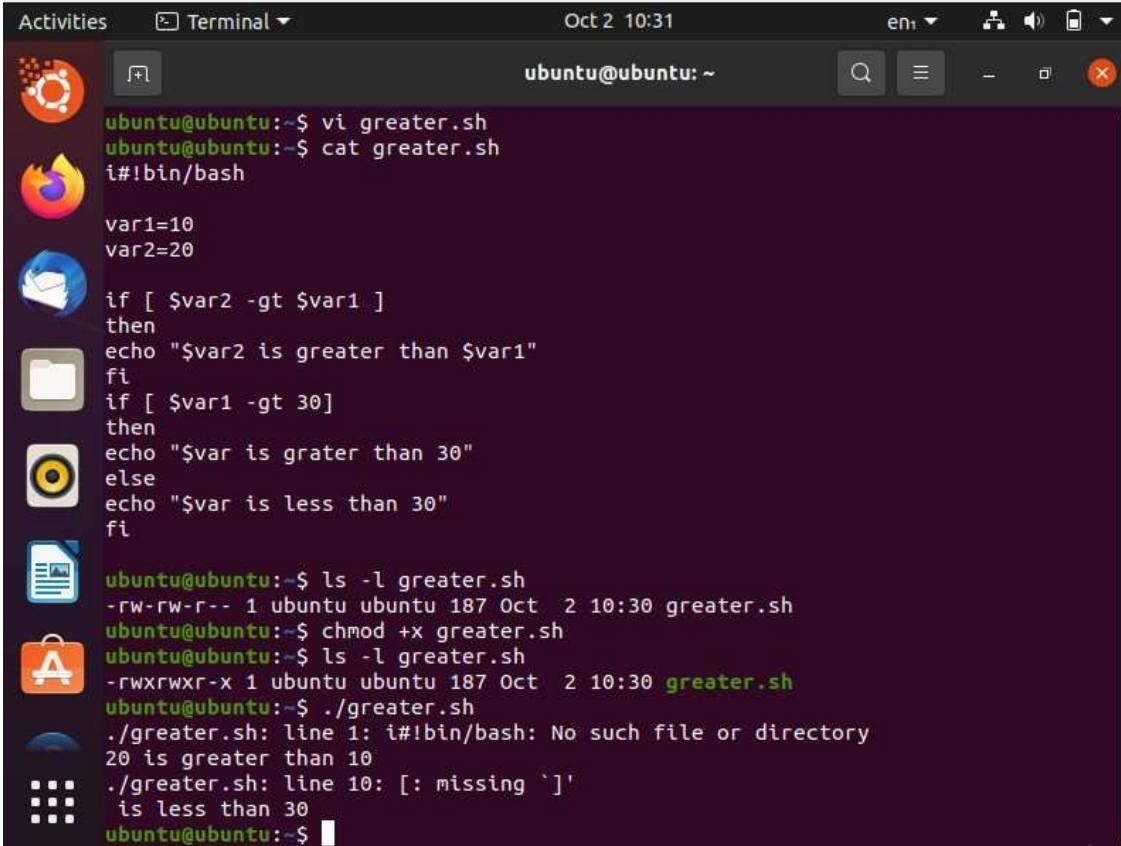


The image shows a terminal window on a Linux desktop. The window title is "Terminal" and it shows the date and time as "Oct 2 10:22". The prompt is "ubuntu@ubuntu: ~". The script content is as follows:

```
i#!/bin/bash
read -p "Enter a number: " number
if [ $((number%2)) -eq 0 ]
then
echo "Number is even."
else
echo "Number is odd."
fi
```

At the bottom of the terminal, a status message reads: "even.sh" 8 lines, 131 characters. The desktop background is dark purple, and the left sidebar contains icons for the Dash, Home, Files, Music, Videos, Documents, Applications, and a grid of other applications.

7. Write a shell script to check a number is greater than, less than or equal to another number.



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ vi greater.sh  
ubuntu@ubuntu:~$ cat greater.sh  
i#!/bin/bash  
  
var1=10  
var2=20  
  
if [ $var2 -gt $var1 ]  
then  
echo "$var2 is greater than $var1"  
fi  
if [ $var1 -gt 30]  
then  
echo "$var is grater than 30"  
else  
echo "$var is less than 30"  
fi  
  
ubuntu@ubuntu:~$ ls -l greater.sh  
-rw-rw-r-- 1 ubuntu ubuntu 187 Oct  2 10:30 greater.sh  
ubuntu@ubuntu:~$ chmod +x greater.sh  
ubuntu@ubuntu:~$ ls -l greater.sh  
-rwxrwxr-x 1 ubuntu ubuntu 187 Oct  2 10:30 greater.sh  
ubuntu@ubuntu:~$ ./greater.sh  
./greater.sh: line 1: i#!/bin/bash: No such file or directory  
20 is greater than 10  
./greater.sh: line 10: [: missing `']`  
is less than 30  
ubuntu@ubuntu:~$
```



```
Activities Terminal Oct 2 10:42 en1 ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi first.sh
ubuntu@ubuntu:~$ cat first.sh
i#!/bin/bash
echo "Enter Size(N)"
read N
i=1
sum=0
echo "Enter Numbers"
while [ $i -le $N ]
do
read num
sum=$((sum + num))
i=$((i + 1))
done
echo $sum

ubuntu@ubuntu:~$ ls -l first.sh
-rw-rw-r-- 1 ubuntu ubuntu 152 Oct  2 10:40 first.sh
ubuntu@ubuntu:~$ chmod +x first.sh
ubuntu@ubuntu:~$ ls -l first.sh
-rwxrwxr-x 1 ubuntu ubuntu 152 Oct  2 10:40 first.sh
ubuntu@ubuntu:~$ ./first.sh
./first.sh: line 1: i#!/bin/bash: No such file or directory
Enter Size(N)
10
Enter Numbers
1
2
3
```

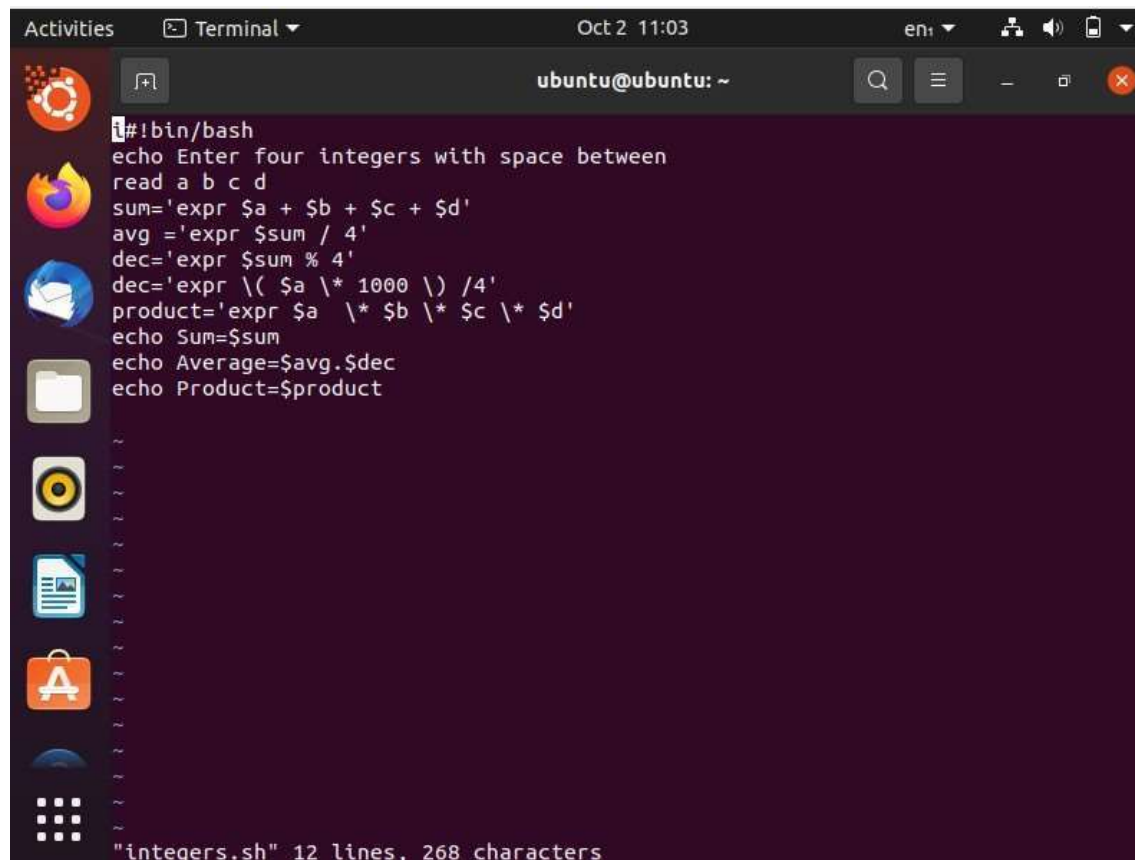
```
Activities Terminal Oct 2 10:42 en1 ubuntu@ubuntu: ~
do
read num
sum=$((sum + num))
i=$((i + 1))
done
echo $sum

ubuntu@ubuntu:~$ ls -l first.sh
-rw-rw-r-- 1 ubuntu ubuntu 152 Oct  2 10:40 first.sh
ubuntu@ubuntu:~$ chmod +x first.sh
ubuntu@ubuntu:~$ ls -l first.sh
-rwxrwxr-x 1 ubuntu ubuntu 152 Oct  2 10:40 first.sh
ubuntu@ubuntu:~$ ./first.sh
./first.sh: line 1: i#!/bin/bash: No such file or directory
Enter Size(N)
10
Enter Numbers
1
2
3
4
5
6
7
8
9
10
55
ubuntu@ubuntu:~$
```



```
Activities  Terminal  Oct 2 11:02  en:  [system icons]
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi integers.sh
ubuntu@ubuntu:~$ cat integers.sh
#!/bin/bash
echo Enter four integers with space between
read a b c d
sum='expr $a + $b + $c + $d'
avg='expr $sum / 4'
dec='expr $sum % 4'
dec='expr \( $a \* 1000 \) /4'
product='expr $a \* $b \* $c \* $d'
echo Sum=$sum
echo Average=$avg.$dec
echo Product=$product

ubuntu@ubuntu:~$ ls -l integers.sh
-rw-rw-r-- 1 ubuntu ubuntu 268 Oct  2 10:59 integers.sh
ubuntu@ubuntu:~$ chmod +x integers.sh
ubuntu@ubuntu:~$ ls -l integers.sh
-rwxrwxr-x 1 ubuntu ubuntu 268 Oct  2 10:59 integers.sh
ubuntu@ubuntu:~$ ./integers.sh
./integers.sh: line 1: i#!/bin/bash: No such file or directory
Enter four integers with space between
3
./integers.sh: line 5: avg: command not found
Sum=expr $a + $b + $c + $d
Average=.expr \( $a \* 1000 \) /4
Product=expr $a \* $b \* $c \* $d
ubuntu@ubuntu:~$
```



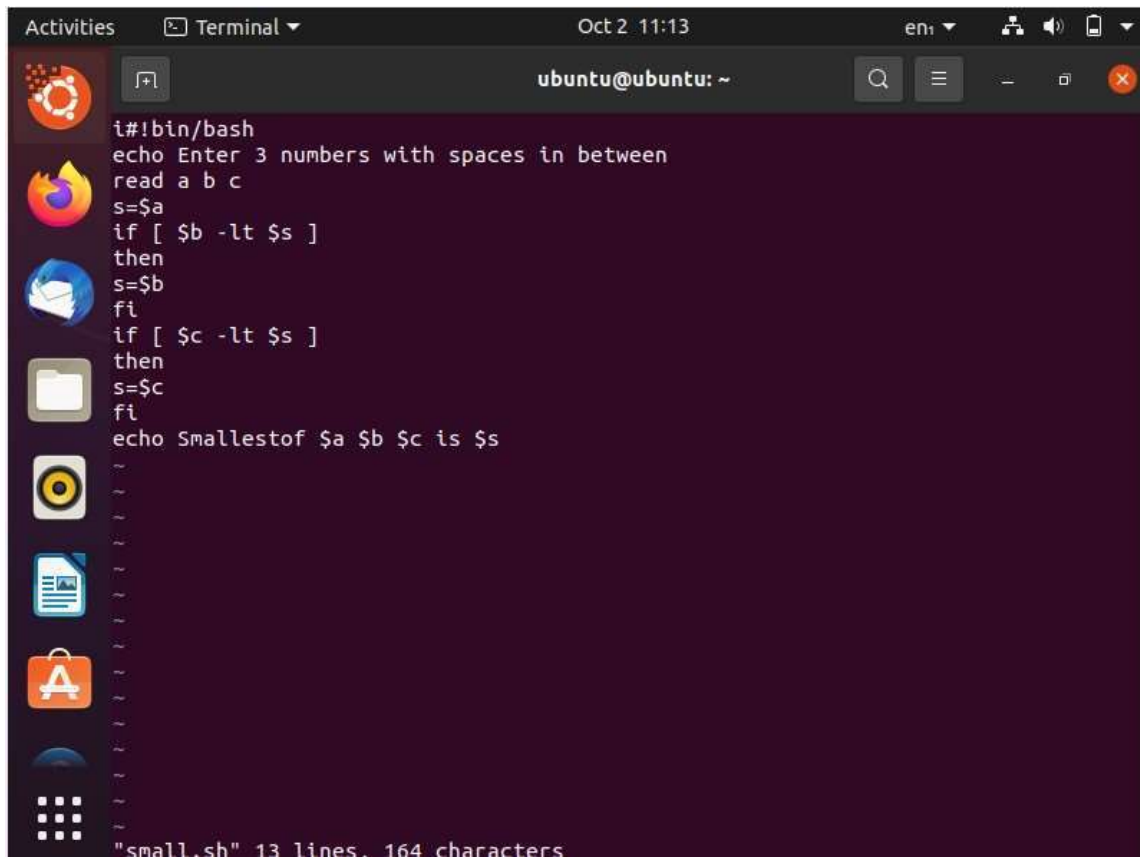
The screenshot shows a terminal window titled "Terminal" with the date and time "Oct 2 11:03". The terminal prompt is "ubuntu@ubuntu: ~". The script content is as follows:

```
#!/bin/bash
echo Enter four integers with space between
read a b c d
sum='expr $a + $b + $c + $d'
avg='expr $sum / 4'
dec='expr $sum % 4'
dec='expr \( $a \* 1000 \) /4'
product='expr $a \* $b \* $c \* $d'
echo Sum=$sum
echo Average=$avg.$dec
echo Product=$product
```

At the bottom of the terminal, it says: "integers.sh" 12 lines, 268 characters

10. Write a shell program to find the smallest of three numbers.

```
Activities Terminal Oct 2 11:12 en1 ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi small.sh
ubuntu@ubuntu:~$ cat small.sh
#!/bin/bash
echo Enter 3 numbers with spaces in between
read a b c
s=$a
if [ $b -lt $s ]
then
s=$b
fi
if [ $c -lt $s ]
then
s=$c
fi
echo Smallestof $a $b $c is $s
ubuntu@ubuntu:~$ ls -l small.sh
-rw-rw-r-- 1 ubuntu ubuntu 164 Oct  2 11:10 small.sh
ubuntu@ubuntu:~$ chmod +x small.sh
ubuntu@ubuntu:~$ ls -l small.sh
-rwxrwxr-x 1 ubuntu ubuntu 164 Oct  2 11:10 small.sh
ubuntu@ubuntu:~$ ./small.sh
./small.sh: line 1: i#!/bin/bash: No such file or directory
Enter 3 numbers with spaces in between
3
./small.sh: line 5: [: -lt: unary operator expected
./small.sh: line 9: [: -lt: unary operator expected
Smallestof 3 is 3
ubuntu@ubuntu:~$ ./small.sh
./small.sh: line 1: i#!/bin/bash: No such file or directory
```



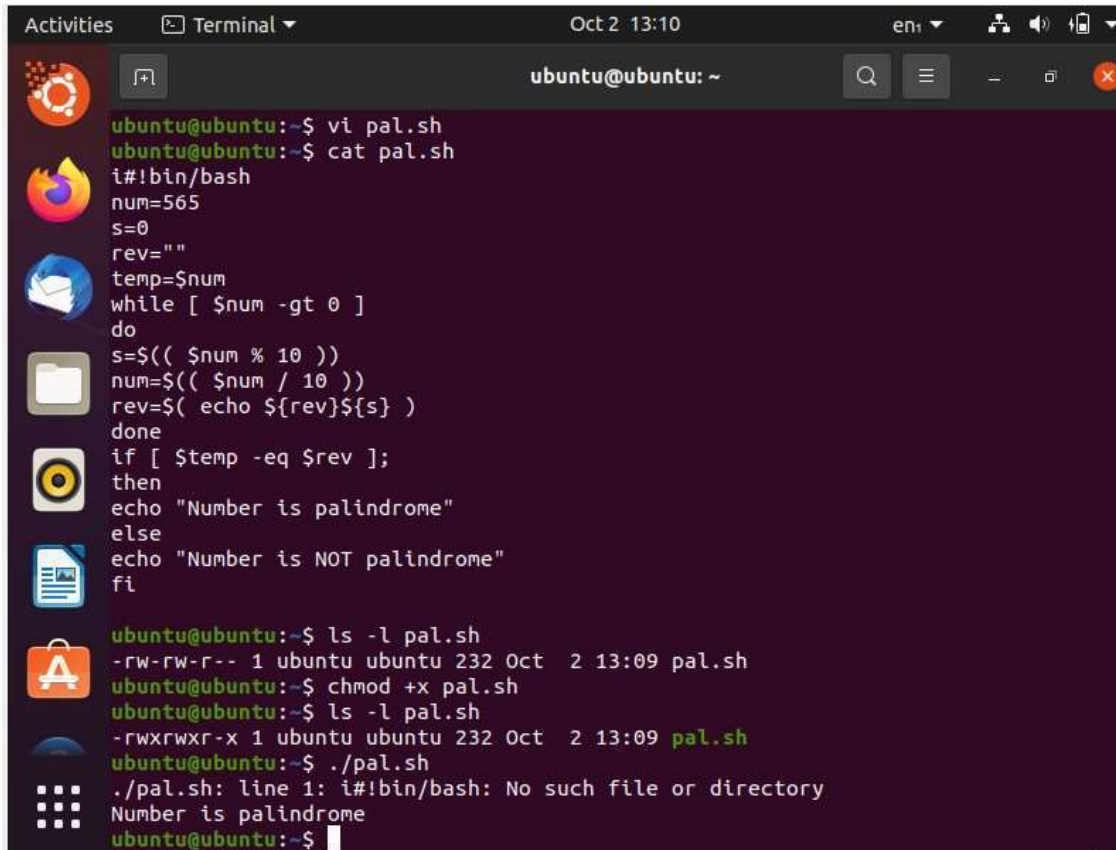
The image shows a terminal window titled "Terminal" with the date and time "Oct 2 11:13". The prompt is "ubuntu@ubuntu: ~". The script content is as follows:

```
i#!/bin/bash
echo Enter 3 numbers with spaces in between
read a b c
s=$a
if [ $b -lt $s ]
then
s=$b
fi
if [ $c -lt $s ]
then
s=$c
fi
echo Smallestof $a $b $c is $s
```

At the bottom of the terminal, it says: "small.sh" 13 lines, 164 characters.

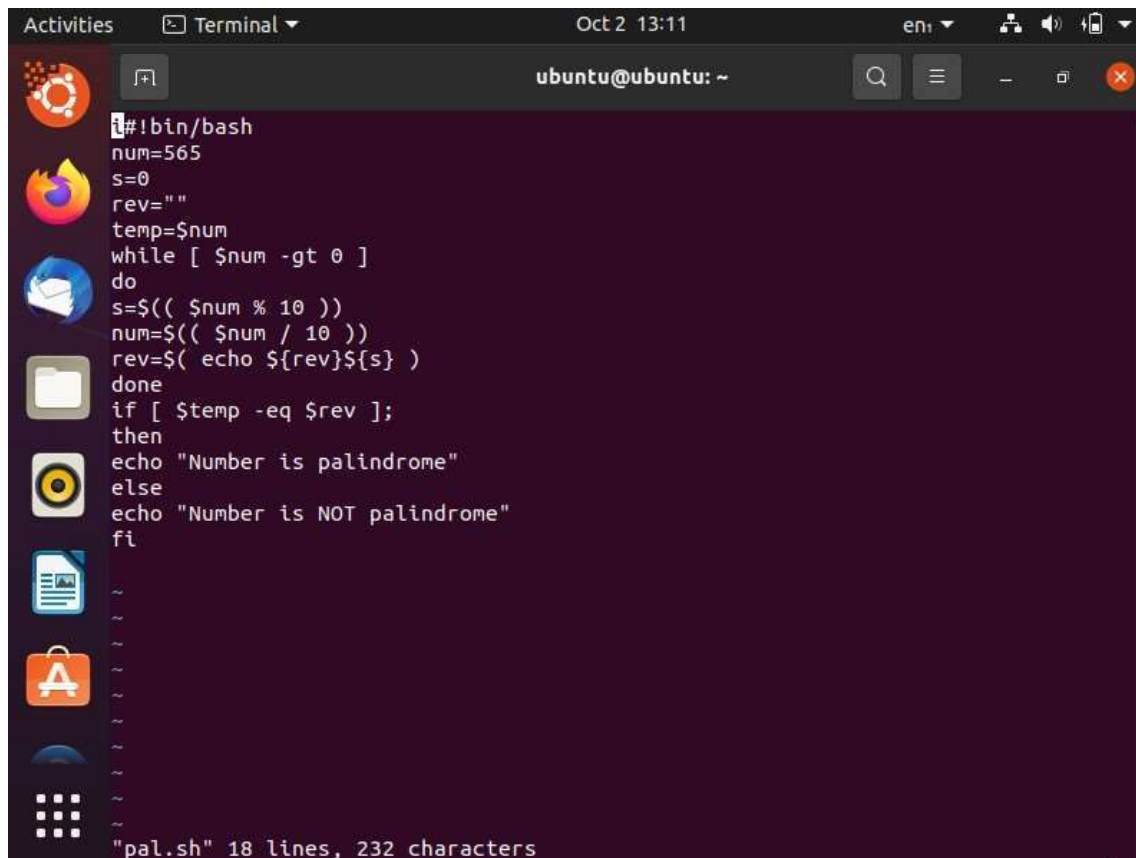
11. Write a shell program to find factorial of given number.

12. Write a shell program to check a number is palindrome or not.



```
Activities Terminal Oct 2 13:10 en: ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi pal.sh
ubuntu@ubuntu:~$ cat pal.sh
#!/bin/bash
num=565
s=0
rev=""
temp=$num
while [ $num -gt 0 ]
do
s=$(( $num % 10 ))
num=$(( $num / 10 ))
rev=$(( echo ${rev}${s} ))
done
if [ $temp -eq $rev ];
then
echo "Number is palindrome"
else
echo "Number is NOT palindrome"
fi

ubuntu@ubuntu:~$ ls -l pal.sh
-rw-rw-r-- 1 ubuntu ubuntu 232 Oct  2 13:09 pal.sh
ubuntu@ubuntu:~$ chmod +x pal.sh
ubuntu@ubuntu:~$ ls -l pal.sh
-rwxrwxr-x 1 ubuntu ubuntu 232 Oct  2 13:09 pal.sh
ubuntu@ubuntu:~$ ./pal.sh
./pal.sh: line 1: i#!/bin/bash: No such file or directory
Number is palindrome
ubuntu@ubuntu:~$
```

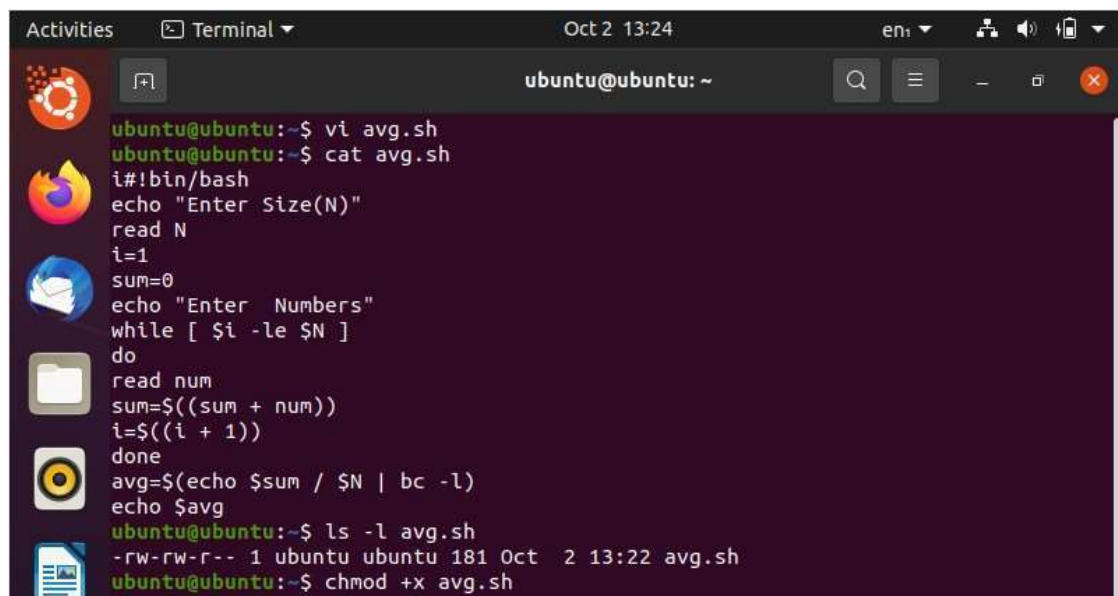


A terminal window titled "Terminal" with a dark background. The window shows a shell script for checking if a number is a palindrome. The script starts with `#!/bin/bash`, sets `num=565`, `s=0`, and `rev=""`. It then enters a `while` loop that calculates the reverse of the number by repeatedly taking the last digit and prepending it to the reversed string. After the loop, it compares the original number with the reversed string. If they are equal, it prints "Number is palindrome"; otherwise, it prints "Number is NOT palindrome". The script ends with `fi`. The status bar at the bottom indicates the script is 18 lines long and 232 characters.

```
#!/bin/bash
num=565
s=0
rev=""
temp=$num
while [ $num -gt 0 ]
do
s=$(( $num % 10 ))
num=$(( $num / 10 ))
rev=$( echo ${rev}${s} )
done
if [ $temp -eq $rev ];
then
echo "Number is palindrome"
else
echo "Number is NOT palindrome"
fi
```

"pal.sh" 18 lines, 232 characters

13. Write a shell script to find the average of the numbers entered in command line.



A terminal window titled "Terminal" with a dark background. The window shows the creation and execution of a shell script named `avg.sh`. The script prompts the user to enter the size of the array (`N`) and then the numbers themselves. It calculates the sum of the numbers and then divides it by `N` to find the average. The script is saved, permissions are set to `chmod +x avg.sh`, and it is executed. The output shows the file details and the calculated average.

```
ubuntu@ubuntu:~$ vi avg.sh
ubuntu@ubuntu:~$ cat avg.sh
#!/bin/bash
echo "Enter Size(N)"
read N
i=1
sum=0
echo "Enter Numbers"
while [ $i -le $N ]
do
read num
sum=$((sum + num))
i=$((i + 1))
done
avg=$(echo $sum / $N | bc -l)
echo $avg
ubuntu@ubuntu:~$ ls -l avg.sh
-rw-rw-r-- 1 ubuntu ubuntu 181 Oct 2 13:22 avg.sh
ubuntu@ubuntu:~$ chmod +x avg.sh
```



```
Activities  Terminal  Oct 2 13:33  en:  [network icon] [sound icon] [battery icon]
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ vi sum.sh
ubuntu@ubuntu:~$ cat sum.sh
i#!/bin/bash
Num=123
g=$Num
s=0
while [ $Num -gt 0 ]
do
k=$(( $Num % 10 ))
Num=$(( $Num / 10 ))
s=$(( $s + $k ))
done
echo "sum of digits of $g is : $s"
ubuntu@ubuntu:~$ ls -l sum.sh
-rw-rw-r-- 1 ubuntu ubuntu 153 Oct  2 13:32 sum.sh
ubuntu@ubuntu:~$ chmod +x sum.sh
ubuntu@ubuntu:~$ ls -l sum.sh
-rwxrwxr-x 1 ubuntu ubuntu 153 Oct  2 13:32 sum.sh
ubuntu@ubuntu:~$ ./sum.sh
./sum.sh: line 1: i#!/bin/bash: No such file or directory
sum of digits of 123 is : 6
ubuntu@ubuntu:~$
```

The screenshot shows a terminal window titled "Terminal" with the prompt "ubuntu@ubuntu: ~". The script being executed is as follows:

```
#!/bin/bash
Num=123
g=$Num
s=0
while [ $Num -gt 0 ]
do
k=$(( $Num % 10 ))
Num=$(( $Num / 10 ))
s=$(( $s + $k ))
done
echo "sum of digits of $g is : $s"
```

The output of the script is "sum of digits of 123 is : 6". At the bottom of the terminal, a status bar indicates: "sum.sh" 11 lines, 153 characters.

15. Write a shell program to check whether given year is leap year or not.

