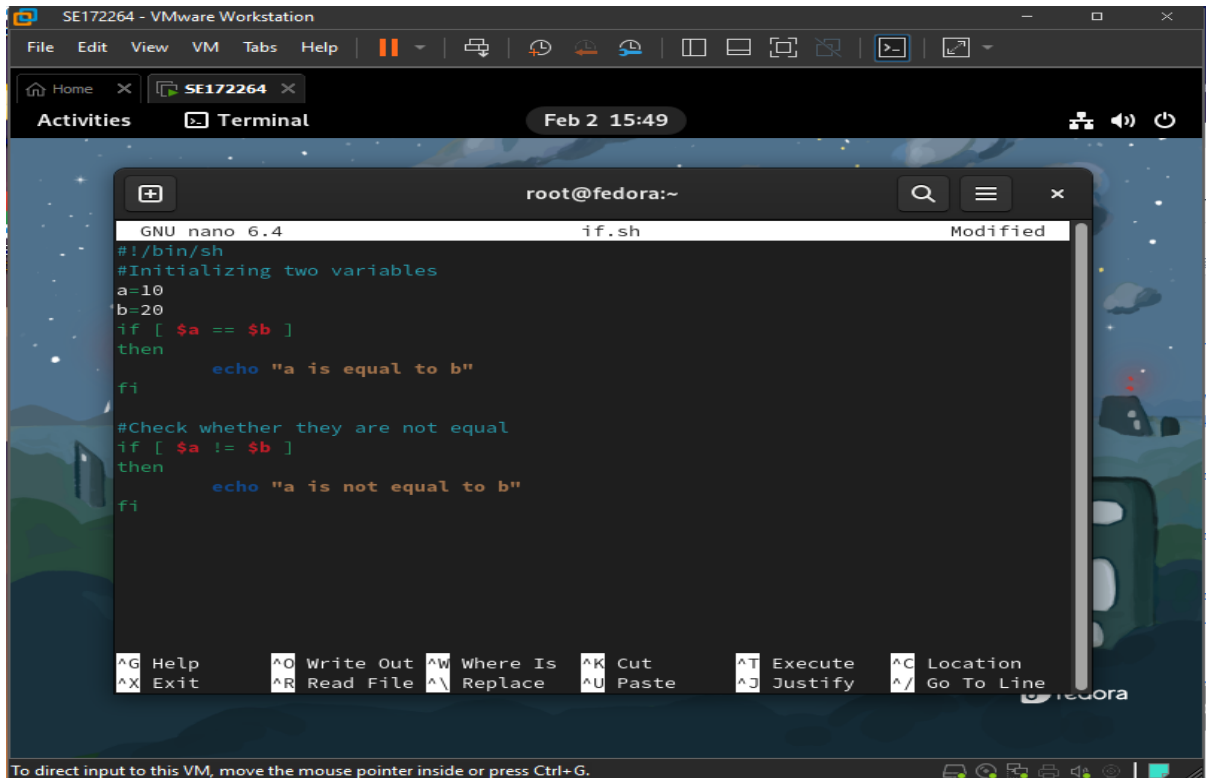


Lab 4 for Operating Systems

Note: Command to take permission: `chmod 777 ./<namefile>.sh`

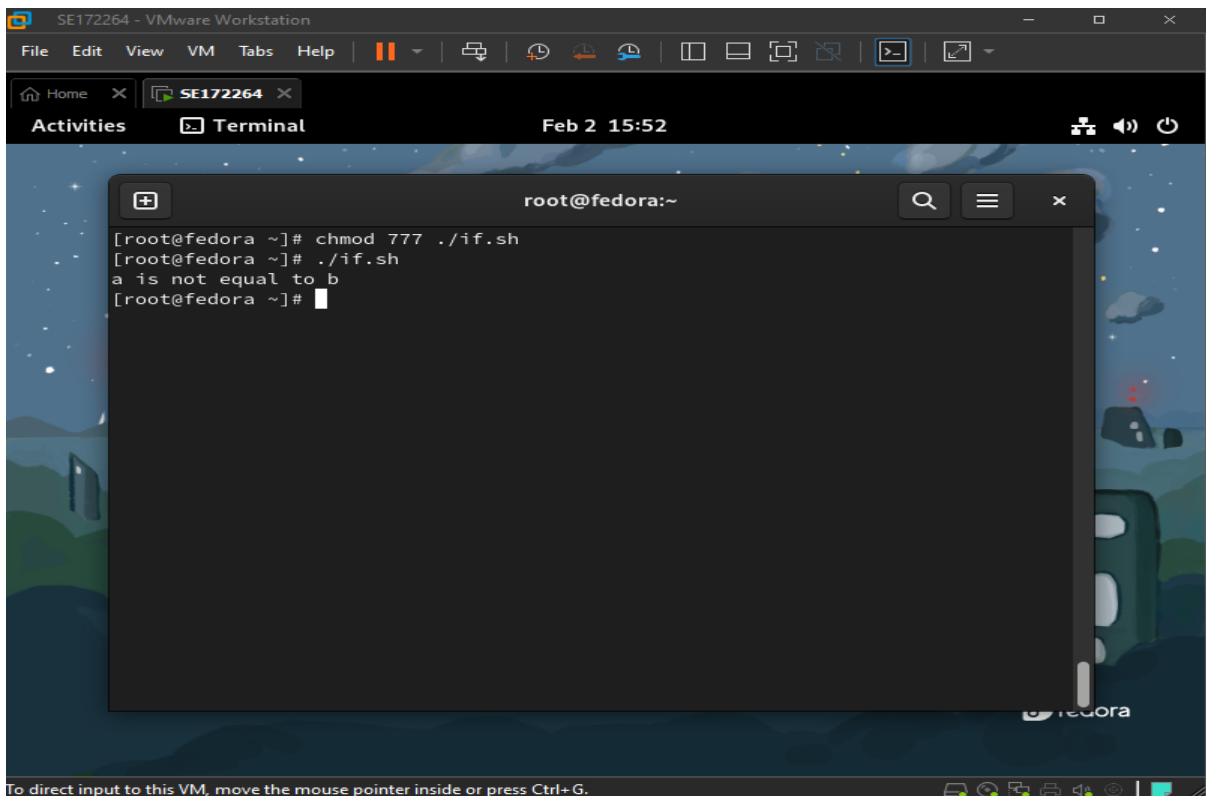
If constructs: `if.sh`



The screenshot shows a terminal window titled 'root@fedora:~' within a VMware Workstation environment. The terminal is running the GNU nano 6.4 text editor to create a file named 'if.sh'. The script content is as follows:

```
#!/bin/sh
#Initializing two variables
a=10
b=20
if [ $a == $b ]
then
    echo "a is equal to b"
fi
#Check whether they are not equal
if [ $a != $b ]
then
    echo "a is not equal to b"
fi
```

The terminal window includes a menu bar with options like Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line. The status bar at the bottom indicates the date and time as 'Feb 2 15:49'.

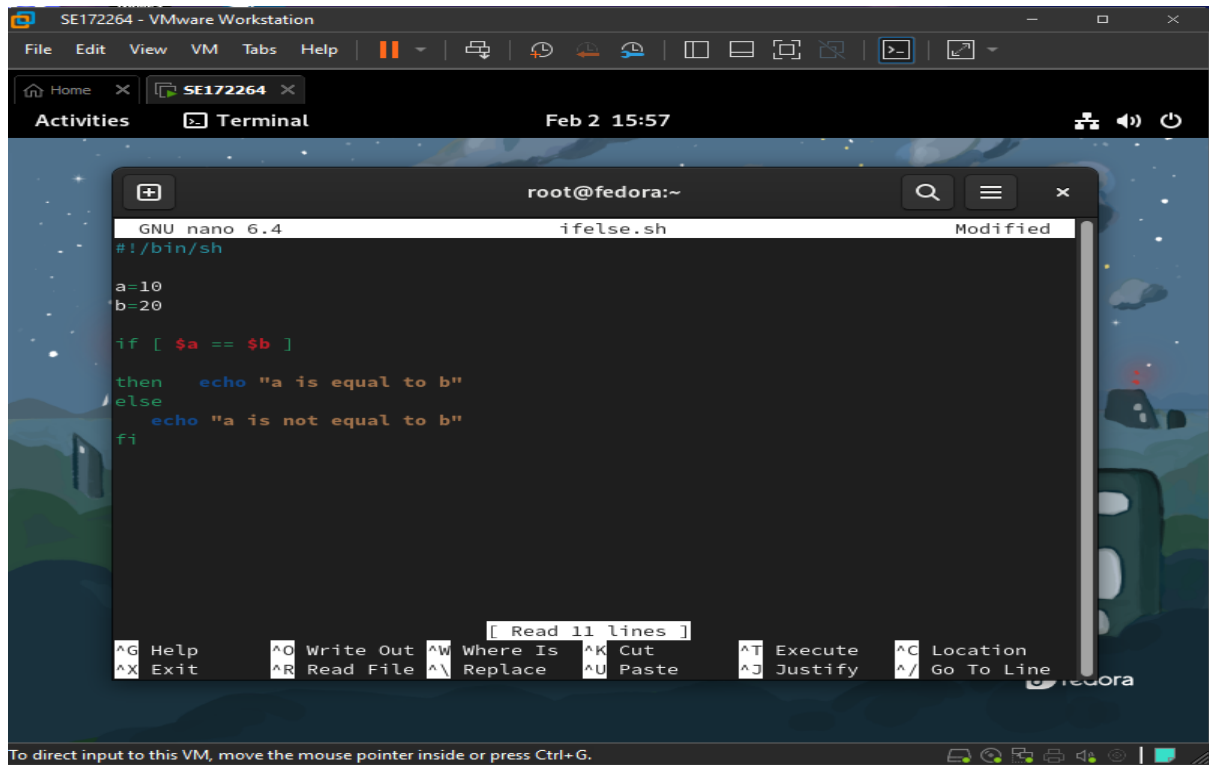


The screenshot shows the same terminal window after the script has been executed. The commands entered and the output are:

```
[root@fedora ~]# chmod 777 ./if.sh
[root@fedora ~]# ./if.sh
a is not equal to b
[root@fedora ~]#
```

The terminal window shows the same menu bar and status bar as the previous screenshot, with the date and time now 'Feb 2 15:52'.

If ... else constructs: ifelse.sh



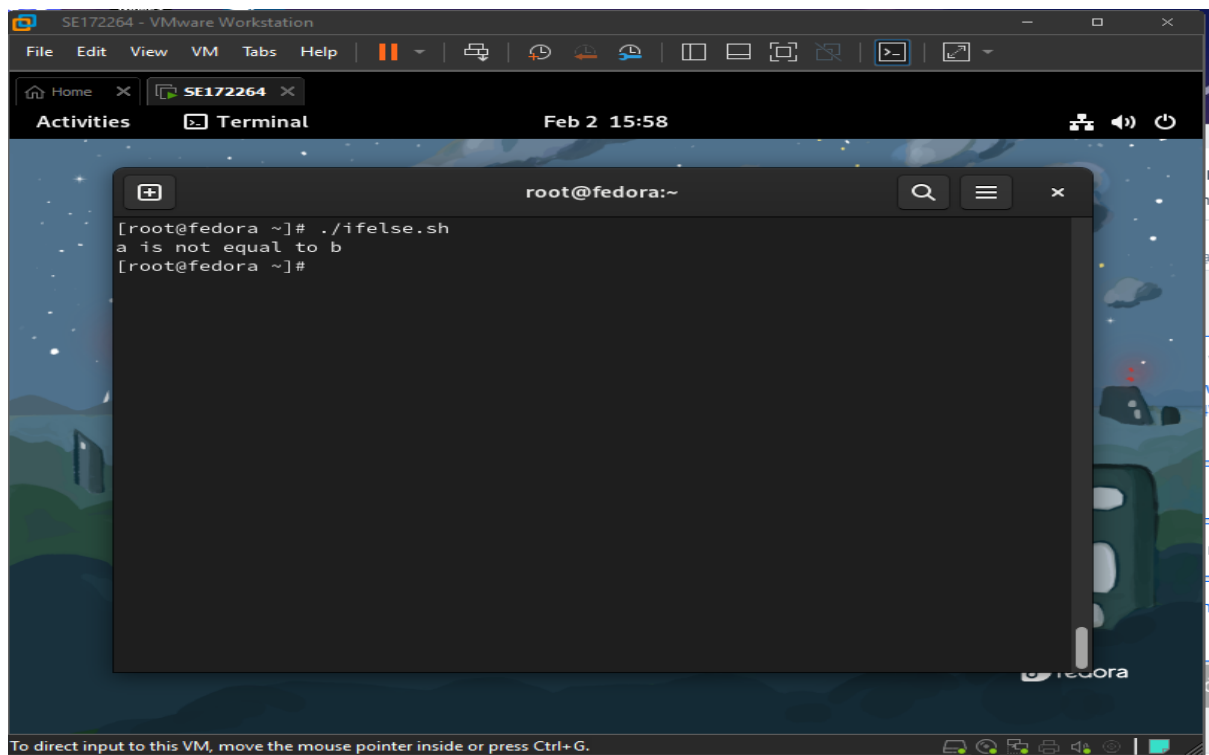
The screenshot shows a terminal window titled "root@fedora:~" with a nano editor open to create a file named "ifelse.sh". The script content is as follows:

```
GNU nano 6.4 ifelse.sh Modified
#!/bin/sh

a=10
b=20

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

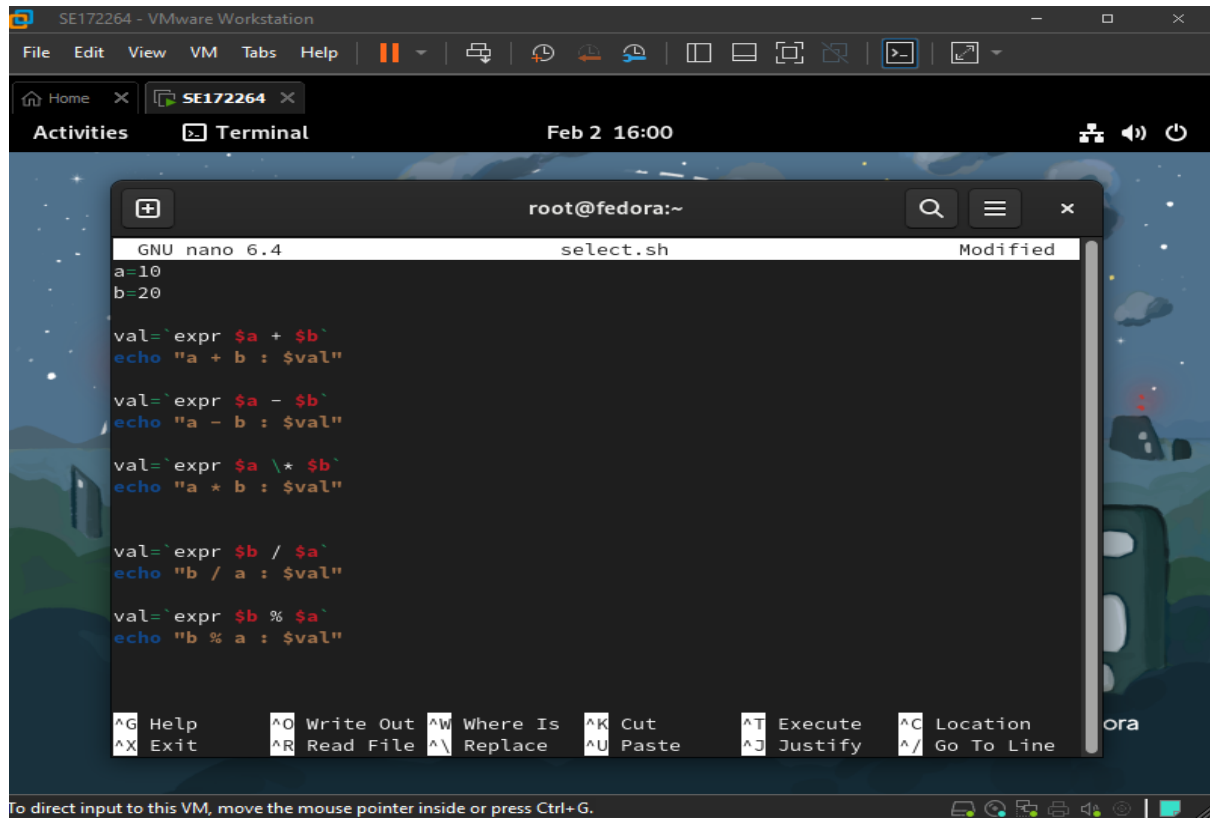
At the bottom of the terminal, a status bar displays various keyboard shortcuts: `^G Help`, `^O Write Out`, `^W Where Is`, `^K Cut`, `^T Execute`, `^C Location`, `^X Exit`, `^R Read File`, `^_ Replace`, `^U Paste`, `^J Justify`, and `^_ Go To Line`.



The screenshot shows the same terminal window after the script has been executed. The prompt is now `[root@fedora ~]#`. The command `./ifelse.sh` has been entered, and the output is `a is not equal to b`.

```
[root@fedora ~]# ./ifelse.sh
a is not equal to b
[root@fedora ~]#
```

Select construct: `select.sh`



The screenshot shows a terminal window titled "root@fedora:~" with a search icon, a menu icon, and a close button. The terminal is running GNU nano 6.4. The script content is as follows:

```
GNU nano 6.4 select.sh Modified
a=10
b=20

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

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

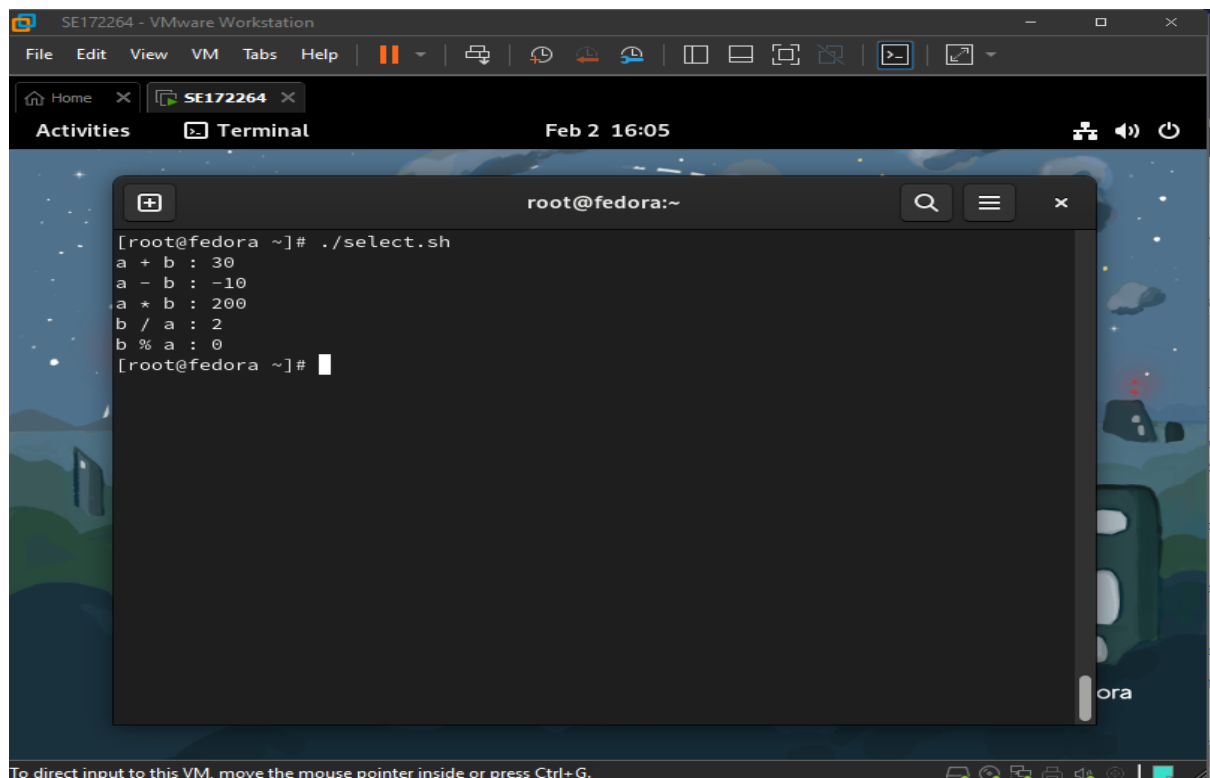
val=`expr $a \* $b`
echo "a * b : $val"

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

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

At the bottom of the terminal window, there is a table of keyboard shortcuts:

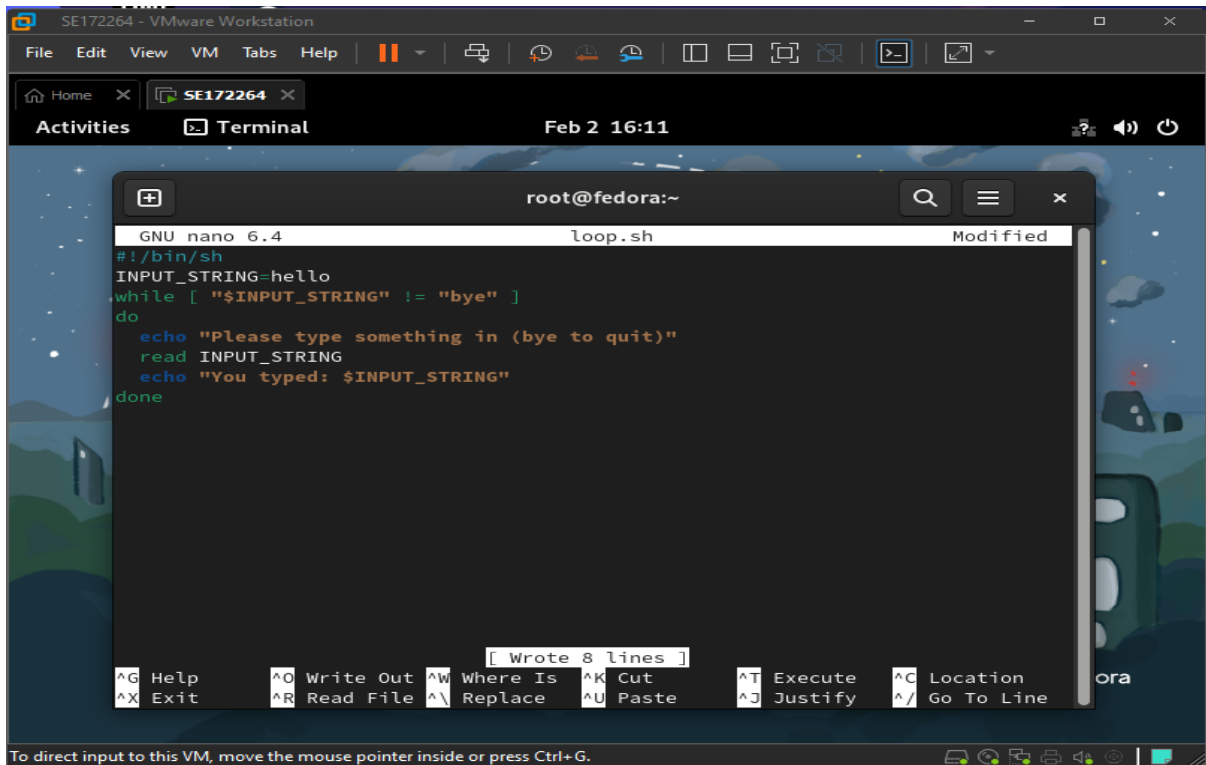
<code>^G</code> Help	<code>^O</code> Write Out	<code>^W</code> Where Is	<code>^K</code> Cut	<code>^T</code> Execute	<code>^C</code> Location
<code>^X</code> Exit	<code>^R</code> Read File	<code>^_\</code> Replace	<code>^U</code> Paste	<code>^J</code> Justify	<code>^_</code> Go To Line



The screenshot shows the same terminal window after running the script. The output is as follows:

```
[root@fedora ~]# ./select.sh
a + b : 30
a - b : -10
a * b : 200
b / a : 2
b % a : 0
[root@fedora ~]#
```

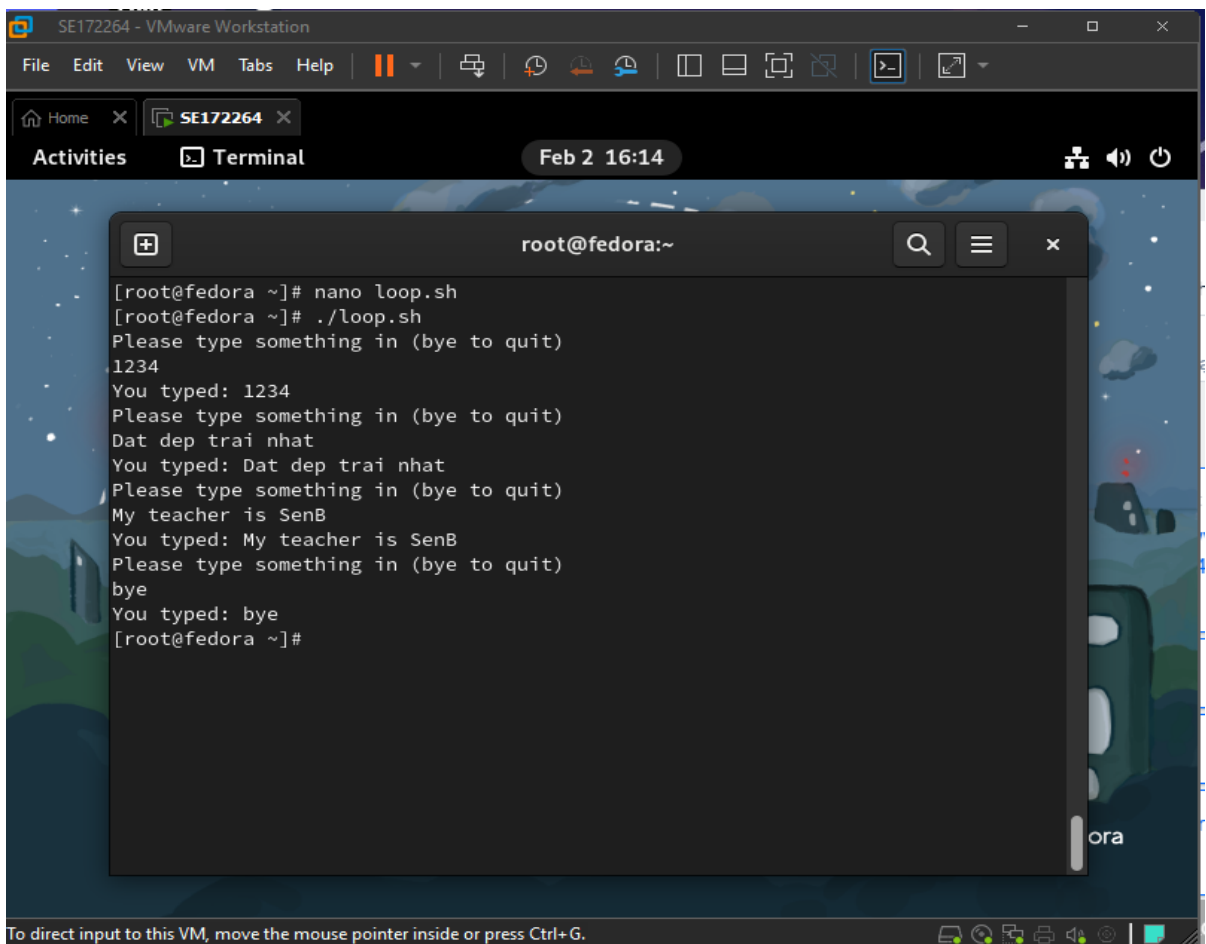
whileloop: loop.sh



The screenshot shows a terminal window titled 'root@fedora:~' with a nano editor open to create a file named 'loop.sh'. The script content is as follows:

```
GNU nano 6.4 loop.sh Modified
#!/bin/sh
INPUT_STRING=hello
while [ "$INPUT_STRING" != "bye" ]
do
    echo "Please type something in (bye to quit)"
    read INPUT_STRING
    echo "You typed: $INPUT_STRING"
done
```

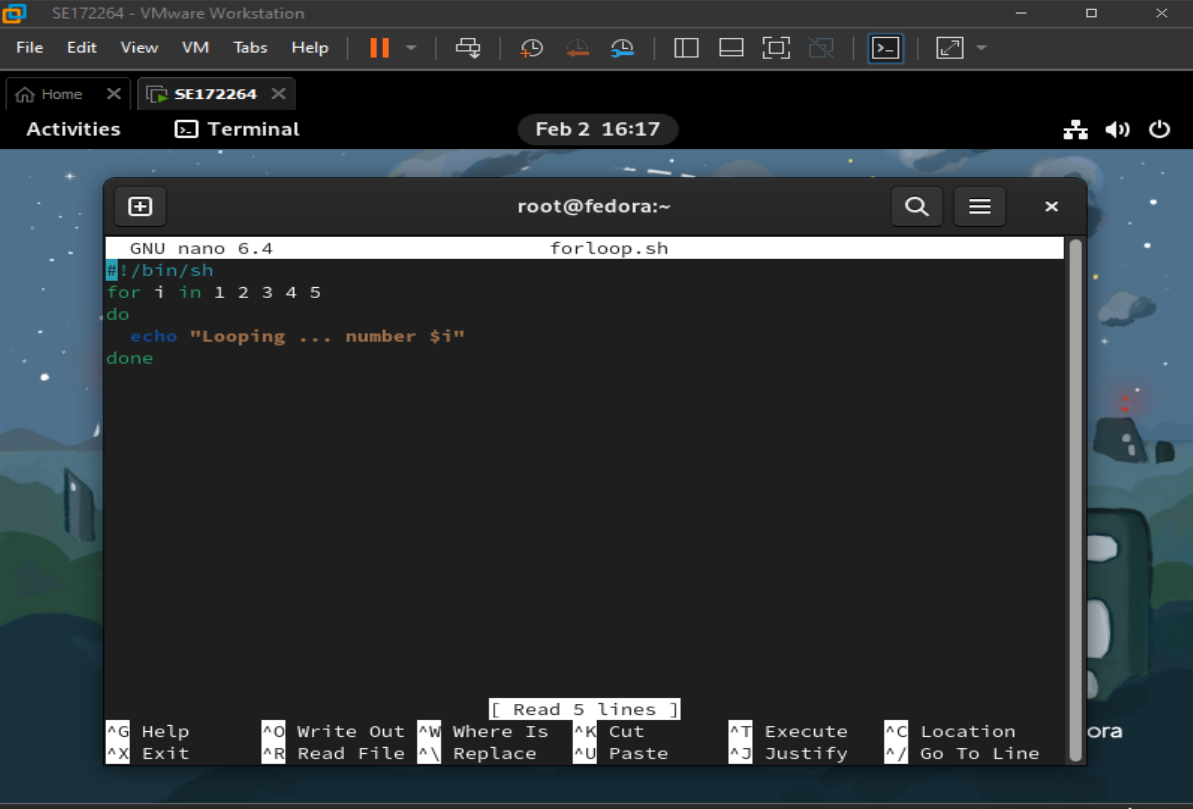
At the bottom of the terminal, a status bar indicates 'Wrote 8 lines' and provides keyboard shortcuts for various actions like Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line.



The screenshot shows the same terminal window after running the script. The output is as follows:

```
[root@fedora ~]# nano loop.sh
[root@fedora ~]# ./loop.sh
Please type something in (bye to quit)
1234
You typed: 1234
Please type something in (bye to quit)
Dat dep trai nhát
You typed: Dat dep trai nhát
Please type something in (bye to quit)
My teacher is SenB
You typed: My teacher is SenB
Please type something in (bye to quit)
bye
You typed: bye
[root@fedora ~]#
```

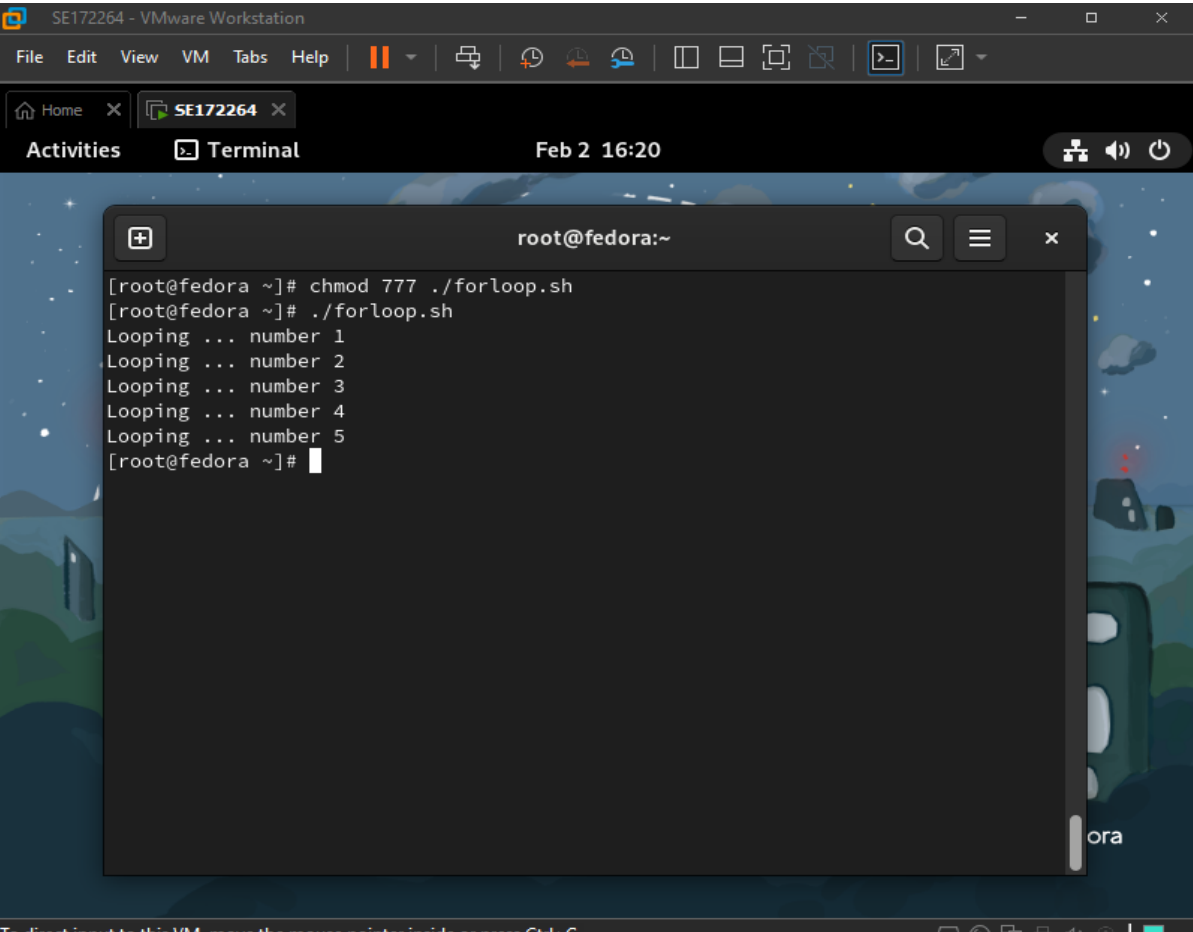
For loop: forloop.sh



The screenshot shows a terminal window titled "root@fedora:~" within a VMware Workstation environment. The terminal is running the nano 6.4 text editor to create a file named "forloop.sh". The script content is as follows:

```
#!/bin/sh
for i in 1 2 3 4 5
do
    echo "Looping ... number $i"
done
```

The terminal window includes a status bar at the bottom with various keyboard shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_/ Go To Line. A message at the bottom of the VM window states: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

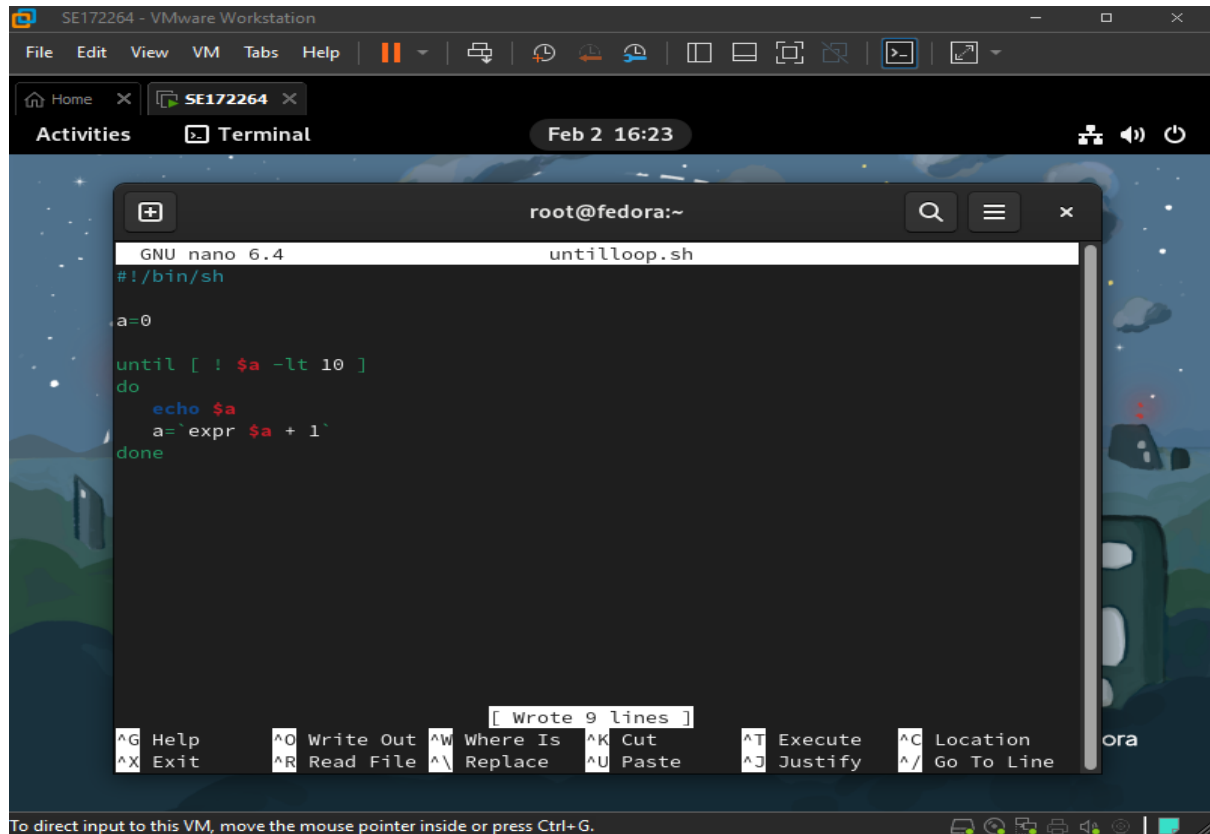


The screenshot shows the same terminal window after the script has been executed. The user has run the command "chmod 777 ./forloop.sh" and then " ./forloop.sh". The output of the script is displayed as five lines of text:

```
[root@fedora ~]# chmod 777 ./forloop.sh
[root@fedora ~]# ./forloop.sh
Looping ... number 1
Looping ... number 2
Looping ... number 3
Looping ... number 4
Looping ... number 5
[root@fedora ~]#
```

The terminal window and the VMware Workstation interface are identical to the previous screenshot, with the same status bar and input instructions at the bottom.

Until loop: untilloop.txt



SE172264 - VMware Workstation

File Edit View VM Tabs Help

Activities Terminal Feb 2 16:23

root@fedora:~

```
GNU nano 6.4 untilloop.sh
#!/bin/sh

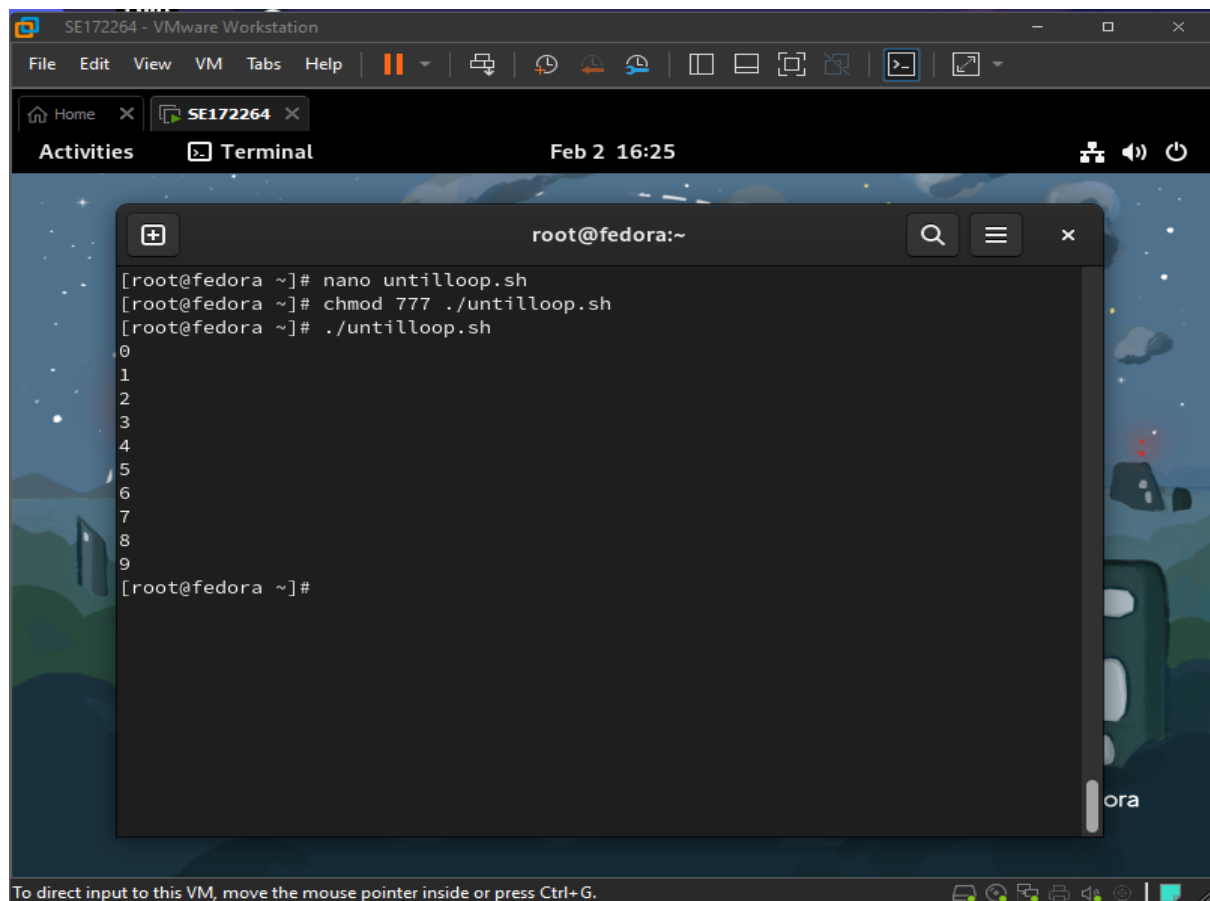
a=0

until [ ! $a -lt 10 ]
do
    echo $a
    a=`expr $a + 1`
done
```

[Wrote 9 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

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



SE172264 - VMware Workstation

File Edit View VM Tabs Help

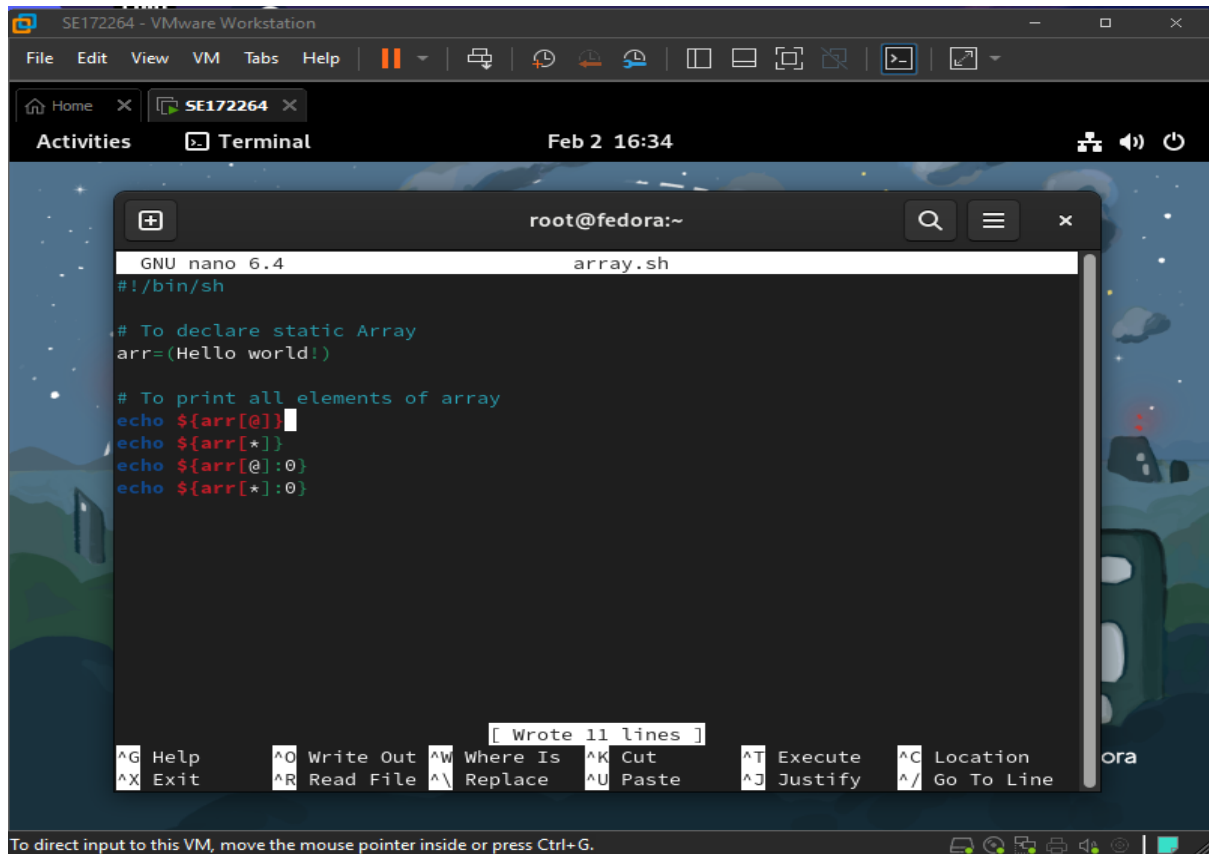
Activities Terminal Feb 2 16:25

root@fedora:~

```
[root@fedora ~]# nano untilloop.sh
[root@fedora ~]# chmod 777 ./untilloop.sh
[root@fedora ~]# ./untilloop.sh
0
1
2
3
4
5
6
7
8
9
[root@fedora ~]#
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Array: array.sh



SE172264 - VMware Workstation

File Edit View VM Tabs Help

SE172264

Activities Terminal Feb 2 16:34

root@fedora:~

GNU nano 6.4 array.sh

```
#!/bin/sh

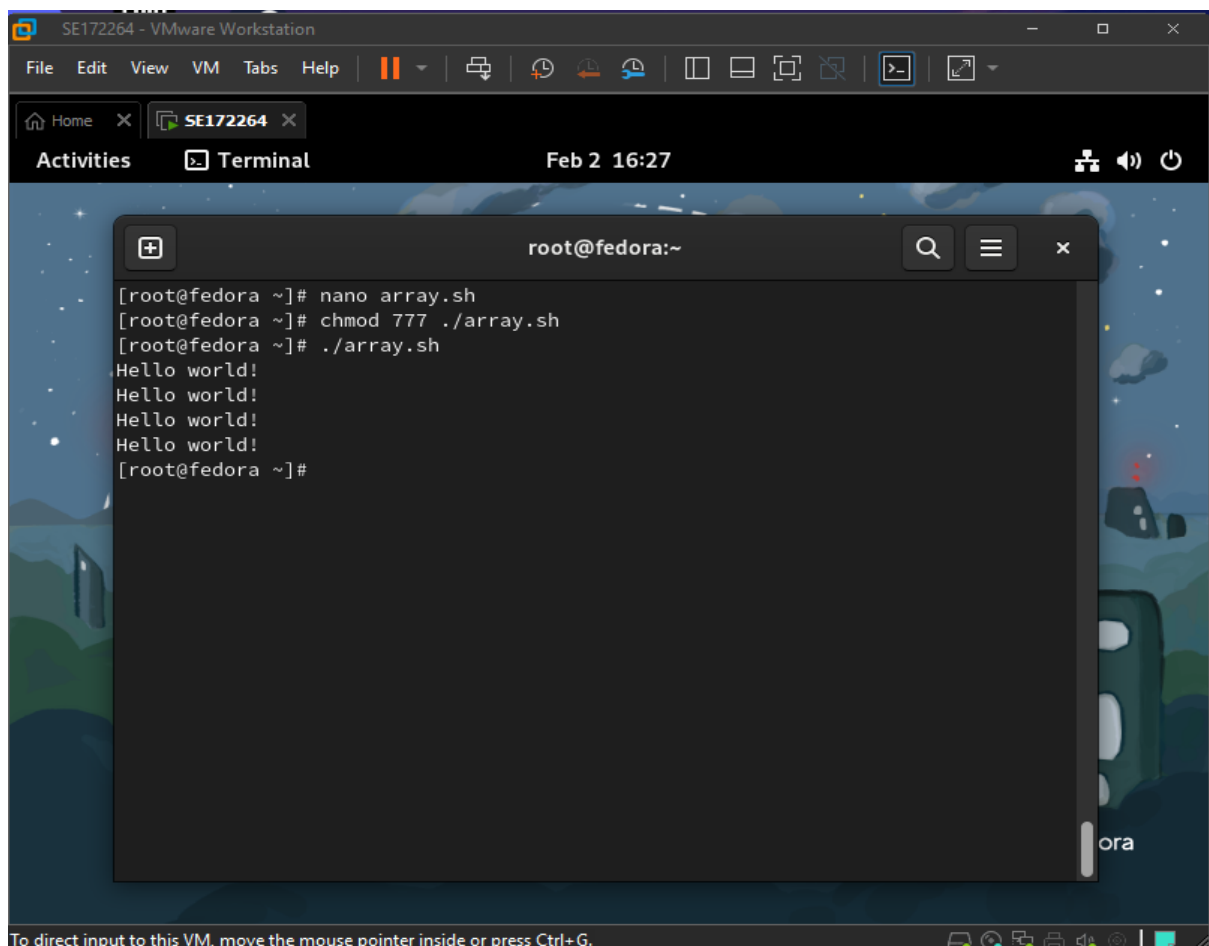
# To declare static Array
arr=(Hello world!)

# To print all elements of array
echo ${arr[@]}
echo ${arr[*]}
echo ${arr[@]:0}
echo ${arr[*]:0}
```

[Wrote 11 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

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



SE172264 - VMware Workstation

File Edit View VM Tabs Help

SE172264

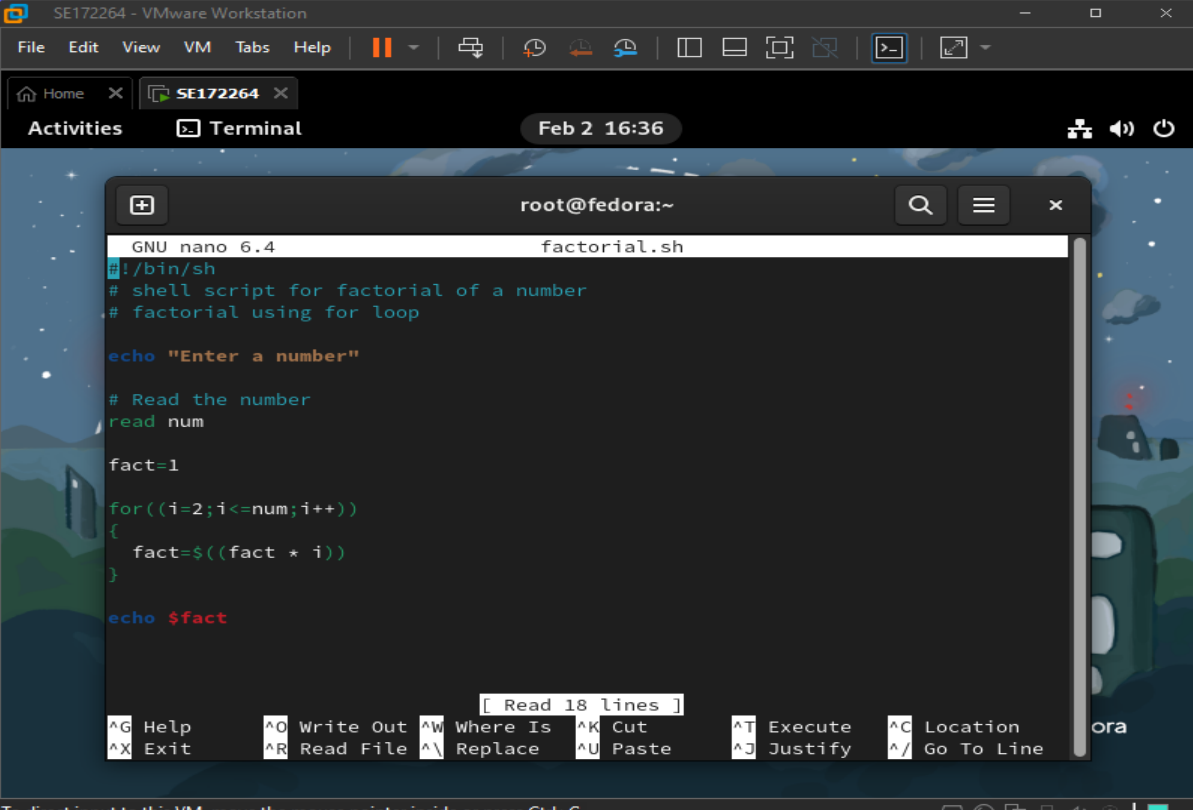
Activities Terminal Feb 2 16:27

root@fedora:~

```
[root@fedora ~]# nano array.sh
[root@fedora ~]# chmod 777 ./array.sh
[root@fedora ~]# ./array.sh
Hello world!
Hello world!
Hello world!
Hello world!
[root@fedora ~]#
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Factorial: factorial.sh



The screenshot shows a terminal window titled "root@fedora:~" with a nano 6.4 editor open to a file named "factorial.sh". The script content is as follows:

```
#!/bin/sh
# shell script for factorial of a number
# factorial using for loop

echo "Enter a number"

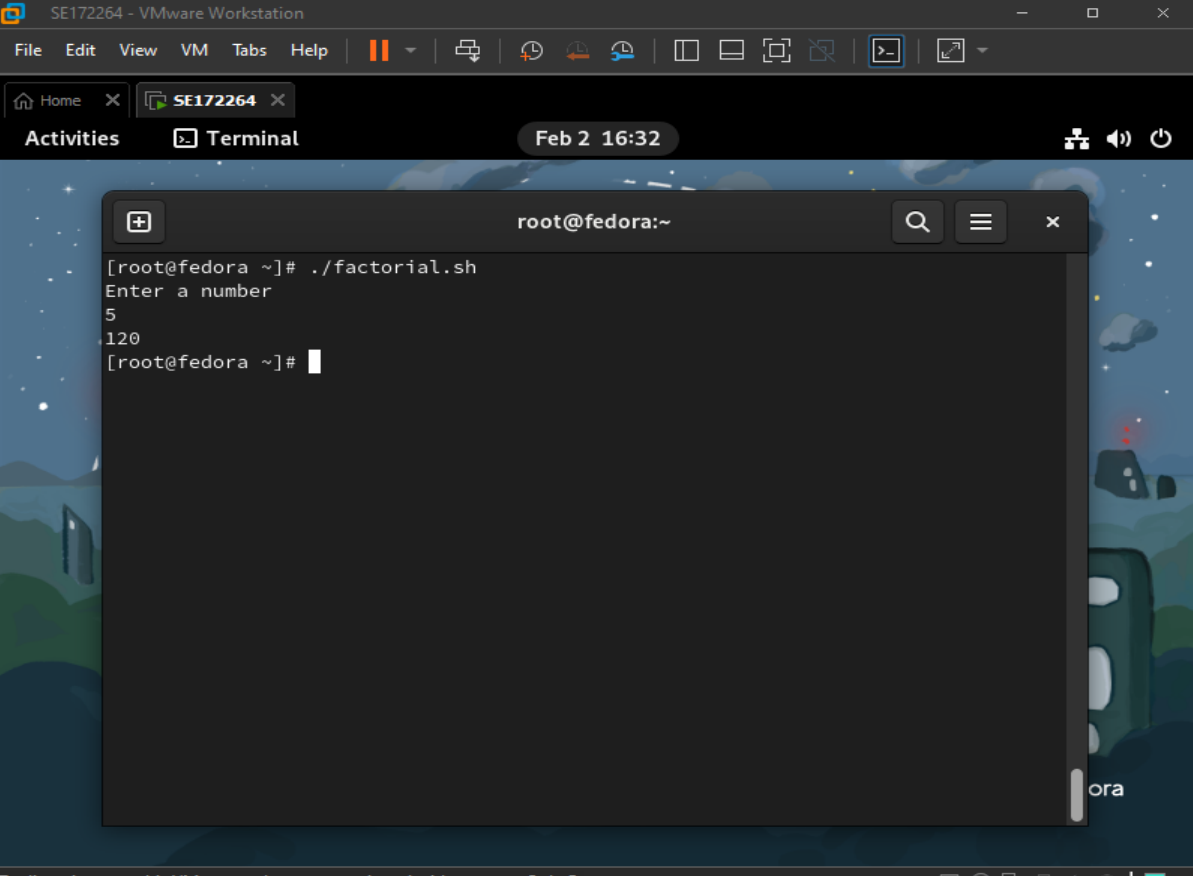
# Read the number
read num

fact=1

for((i=2;i<=num;i++))
{
    fact=$((fact * i))
}

echo $fact
```

The terminal window includes a status bar at the bottom with various keyboard shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_ Go To Line. A message at the bottom of the window states: "To direct input to this VM, move the mouse pointer inside or press Ctrl+G."

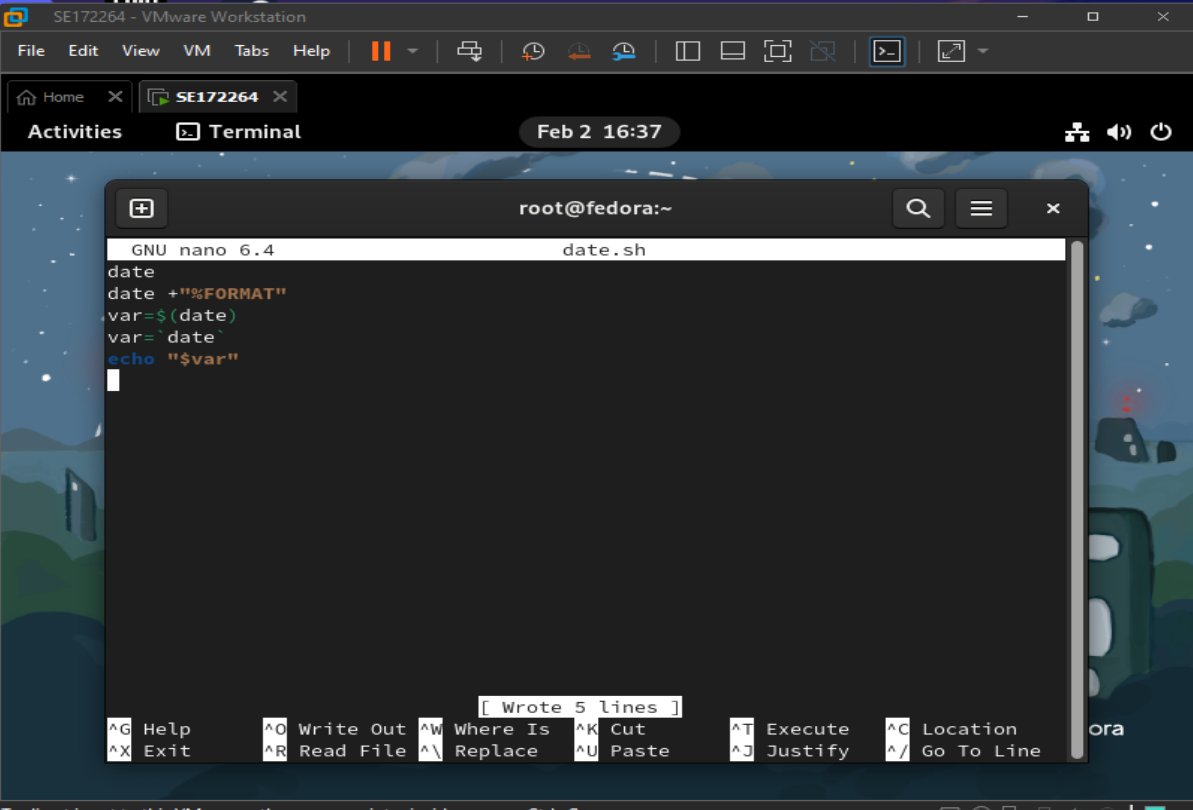


The screenshot shows the same terminal window after the script has been executed. The prompt is now "[root@fedora ~]# ./factorial.sh". The user has entered "5" and the script has output "120". The prompt is now "[root@fedora ~]# " with a cursor.

```
[root@fedora ~]# ./factorial.sh
Enter a number
5
120
[root@fedora ~]#
```

The terminal window still shows the same status bar and the "To direct input to this VM, move the mouse pointer inside or press Ctrl+G." message at the bottom.

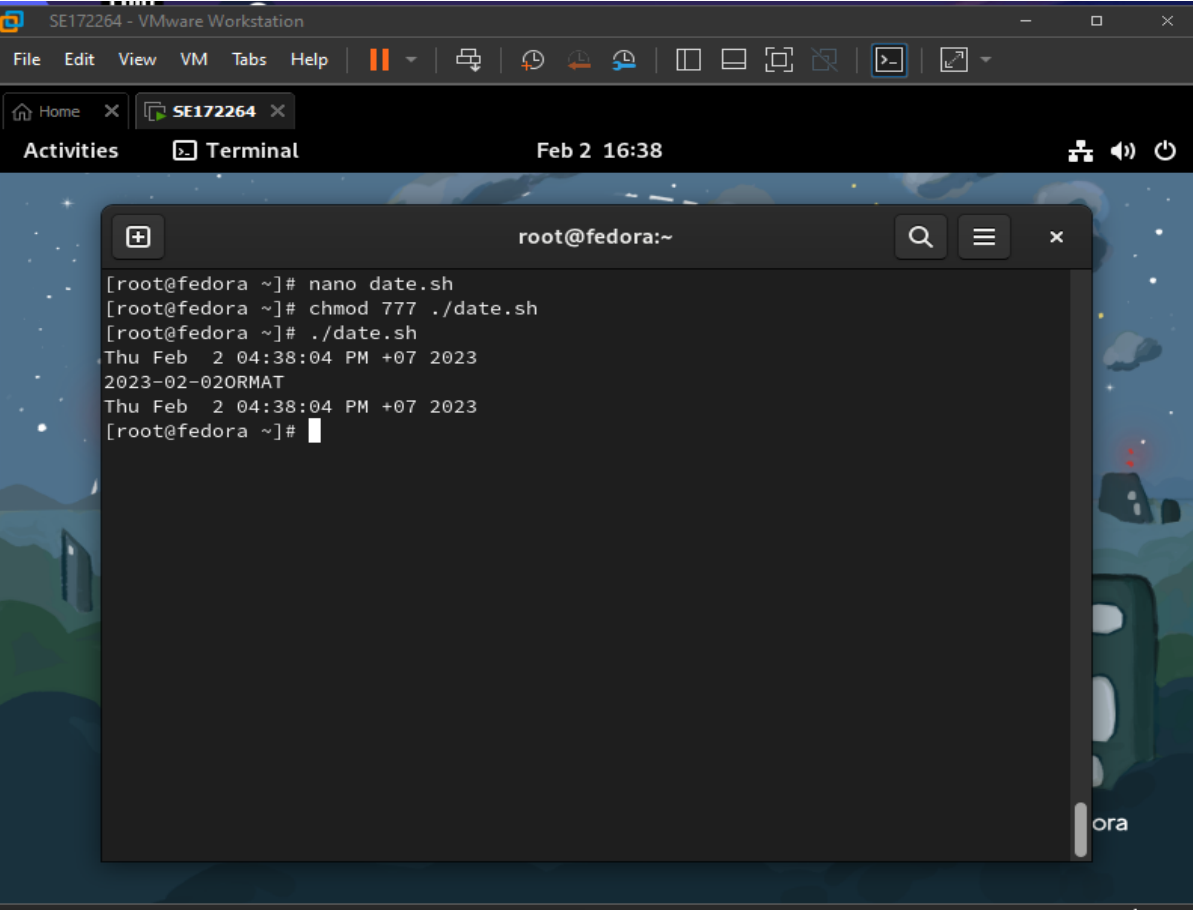
Checking date: date.sh



The screenshot shows a terminal window titled "root@fedora:~" inside a VMware Workstation. The terminal is running the GNU nano 6.4 editor, editing a file named "date.sh". The code in the file is as follows:

```
date
date +"%FORMAT"
var=$(date)
var=`date`
echo "$var"
```

The terminal window has a status bar at the bottom that says "To direct input to this VM, move the mouse pointer inside or press Ctrl+G." The VMware Workstation window title is "SE172264 - VMware Workstation".

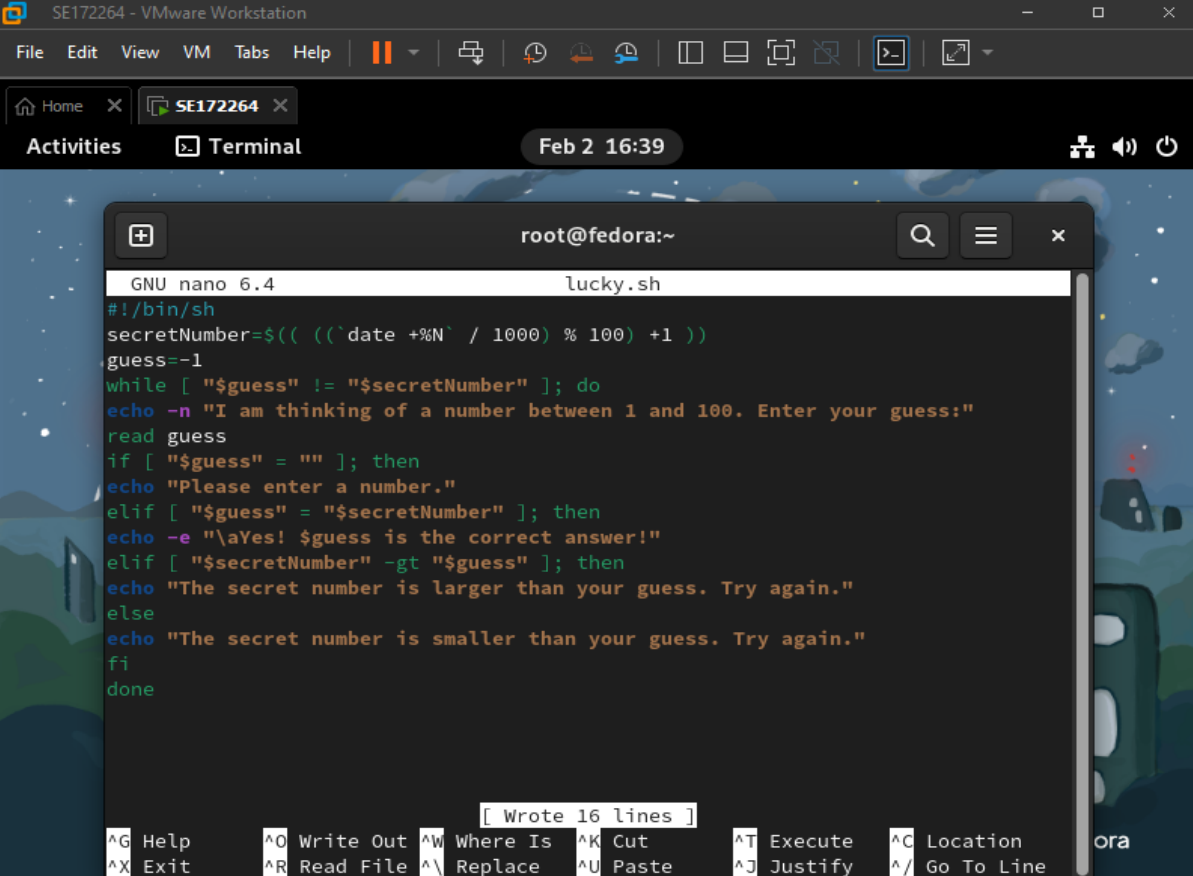


The screenshot shows the same terminal window after the script has been executed. The terminal output is as follows:

```
[root@fedora ~]# nano date.sh
[root@fedora ~]# chmod 777 ./date.sh
[root@fedora ~]# ./date.sh
Thu Feb  2 04:38:04 PM +07 2023
2023-02-020RMAT
Thu Feb  2 04:38:04 PM +07 2023
[root@fedora ~]#
```

The terminal window still has the same status bar and VMware Workstation window title.

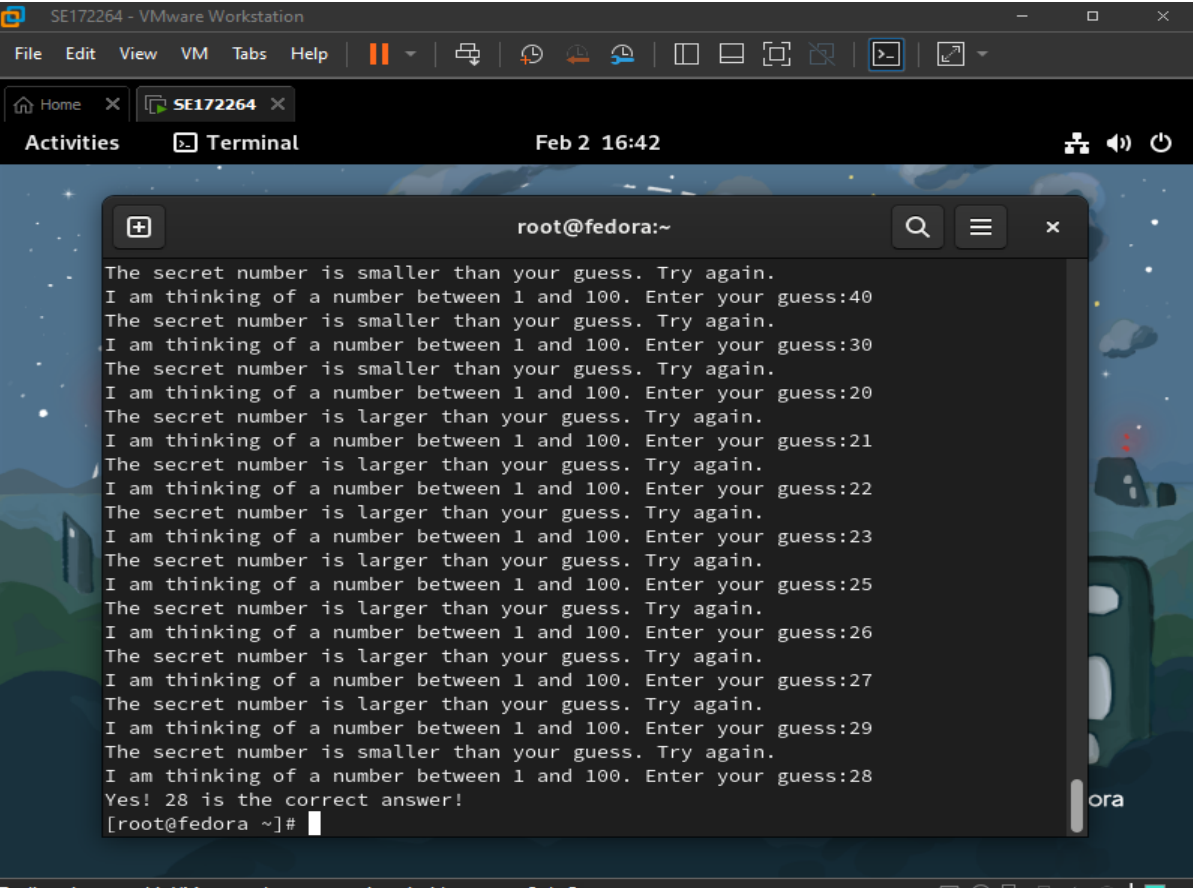
Lucky Number: lucky.sh



The screenshot shows a terminal window titled "root@fedora:~" with a nano 6.4 editor open to a file named "lucky.sh". The script content is as follows:

```
#!/bin/sh
secretNumber=$(( (`date +%N` / 1000) % 100) + 1 ))
guess=-1
while [ "$guess" != "$secretNumber" ]; do
echo -n "I am thinking of a number between 1 and 100. Enter your guess:"
read guess
if [ "$guess" = "" ]; then
echo "Please enter a number."
elif [ "$guess" = "$secretNumber" ]; then
echo -e "\aYes! $guess is the correct answer!"
elif [ "$secretNumber" -gt "$guess" ]; then
echo "The secret number is larger than your guess. Try again."
else
echo "The secret number is smaller than your guess. Try again."
fi
done
```

The bottom of the terminal shows the nano editor's status bar with various shortcuts like ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_/ Go To Line. A message "[Wrote 16 lines]" is also visible.



The screenshot shows the same terminal window after running the script. The output is as follows:

```
The secret number is smaller than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:40
The secret number is smaller than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:30
The secret number is smaller than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:20
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:21
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:22
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:23
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:25
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:26
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:27
The secret number is larger than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:29
The secret number is smaller than your guess. Try again.
I am thinking of a number between 1 and 100. Enter your guess:28
Yes! 28 is the correct answer!
[root@fedora ~]#
```

The terminal window title is "root@fedora:~" and the date/time is "Feb 2 16:42".

