



**Daffodil**  
*International*  
**University**

## Lab Report

Course Code: CSE324

Course Title: Operating Systems Lab

Experiment No: 02

Experiment Name: Introduction to Linux commands

Submitted To:

Teacher Name: Nushrat Jahan Oyshi

Department: CSE

Daffodil International University.

Submitted By:

Name: Sumaiya Salekin Saba

ID: 0242220005101631

Section: 63\_M

Semester: Spring-25

Department: CSE

Daffodil International University.

Submission Date: 03 February, 2025

## Introduction:

This lab focuses on handling multiple inputs and processing them efficiently in operating systems. Arrays are used to organize and store inputs, allowing operations like sorting and iteration. The lab also emphasizes the use of if statements to implement decision-making based on specific conditions. Combining arrays with conditional logic provides practical experience in input management, enhancing problem-solving skills and understanding data handling in operating systems.

## Read Multiple Inputs:

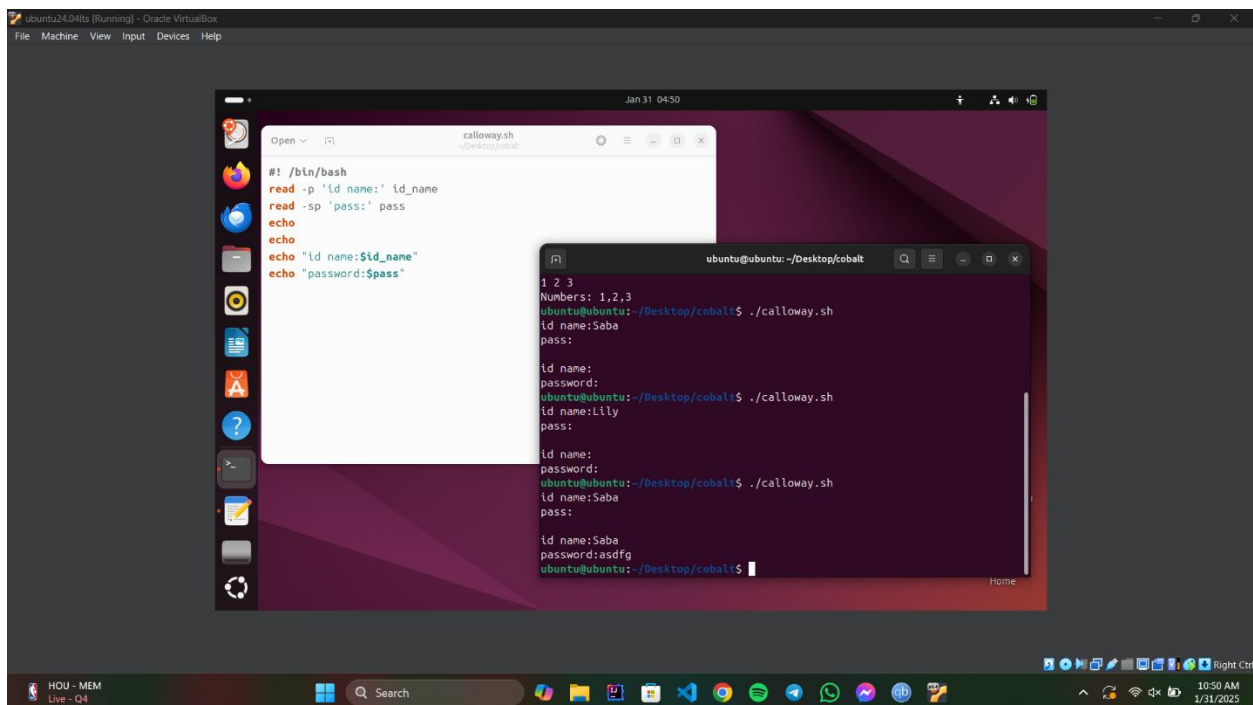
**read id\_name:"id name: "** : Reads input from the user and stores it in the variable id\_name. The text after the variable name is the prompt displayed to the user.

**read -sp pass:"pass: "** : Reads input from the user without displaying it on the screen. This is crucial for password input. The input is stored in the variable pass.

**echo** (blank lines): These lines create empty lines in the output for better readability.

**echo "id name:\$id\_name"** : Prints "id name:" followed by the value stored in the id\_name variable.

**echo "password:\$pass"** : Prints "password:" followed by the value stored in the pass variable.



The screenshot shows a virtual machine window titled 'ubuntu2404lts [running] - Oracle VM VirtualBox'. Inside the VM, there is a desktop environment with a purple background. A terminal window is open, displaying the execution of a script named 'calloway.sh'. The script's content is visible in a separate window above the terminal. The terminal shows the following output:

```
1 2 3
Numbers: 1,2,3
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
id name:Saba
pass:

id name:
password:
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
id name:lilly
pass:

id name:
password:
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
id name:Saba
pass:

id name:Saba
password:asdfg
ubuntu@ubuntu:~/Desktop/cobalt$
```

**Writing the script:**

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

```
read -p 'id name:' id_name
```

```
read -sp 'pass:' pass
```

```
echo
```

```
echo
```

```
echo "id name: $id_name"
```

```
echo "password:$pass"
```

**Output:**

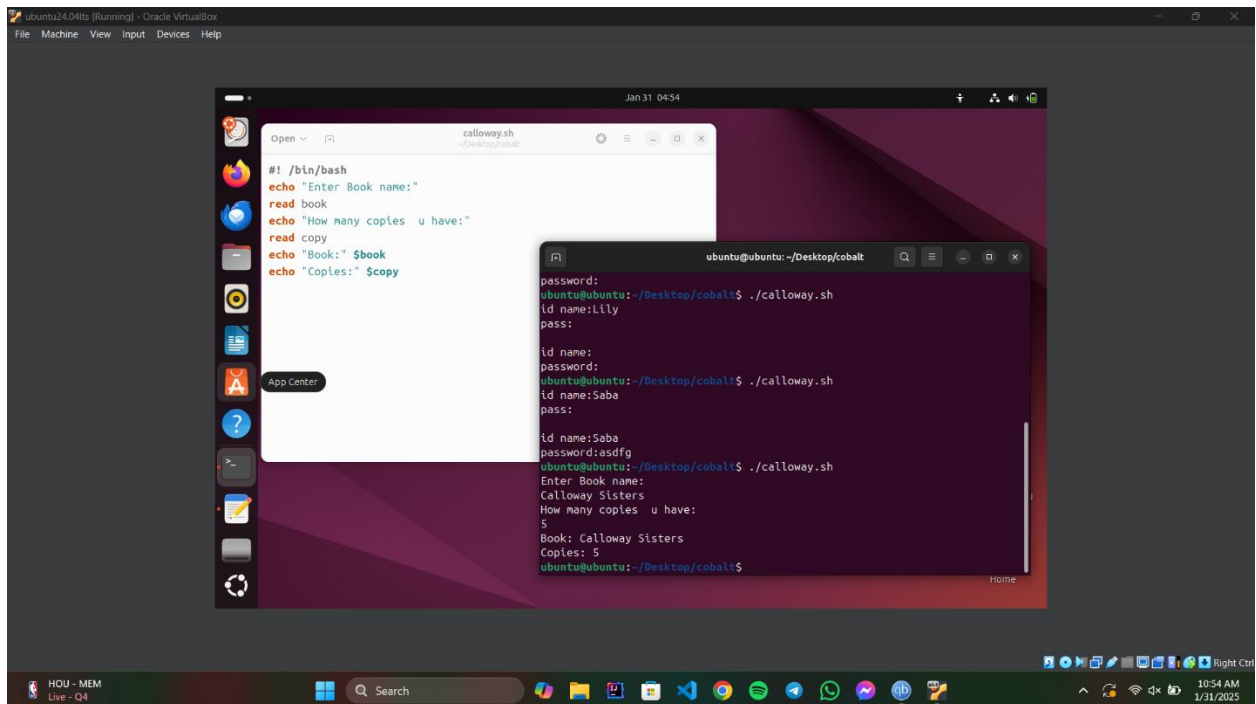
id name: Saba

pass:

id name: Saba

pass:asdfg

**Practice:**



```
#!/bin/bash
echo "Enter numbers:"
read n1 n2 n3
echo "Numbers: $n1,$n2,$n3 "
```

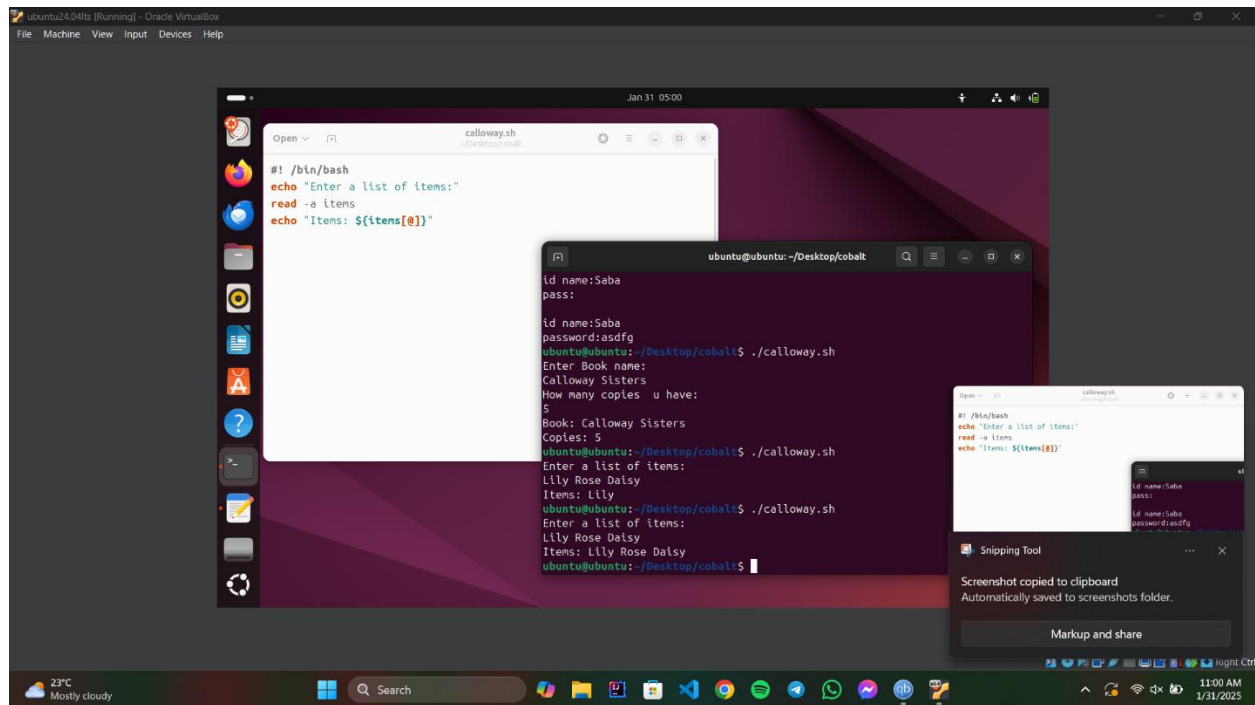
```
ubuntu@ubuntu: ~/Desktop/cobalt
ubuntu@ubuntu: ~/Desktop$ cd Desktop
ubuntu@ubuntu: ~/Desktop$ mkdir cobalt
ubuntu@ubuntu: ~/Desktop$ cd cobalt
ubuntu@ubuntu: ~/Desktop/cobalt$ ls
ubuntu@ubuntu: ~/Desktop/cobalt$ touch calloway.sh
ubuntu@ubuntu: ~/Desktop/cobalt$ chmod +x calloway.sh
ubuntu@ubuntu: ~/Desktop/cobalt$ ls -al
total 0
drwxrwxr-x 2 ubuntu ubuntu 60 Jan 31 04:39 .
drwxr-xr-x 3 ubuntu ubuntu 80 Jan 31 04:38 ..
-rwxrwxr-x 1 ubuntu ubuntu 0 Jan 31 04:39 calloway.sh
ubuntu@ubuntu: ~/Desktop/cobalt$ ./calloway.sh
Enter numbers:
1 2 3
Numbers: 1,2,3
ubuntu@ubuntu: ~/Desktop/cobalt$
```

## Read Multiple Inputs using Array:

**echo "Enter a list of items:"**: This line prints the message "Enter a list of items:" to the console, prompting the user for input.

**Read -a items: -a**: This is a flag for the read command. It tells read to split the input into separate words (separated by spaces or tabs) and store them as elements of an array. This line reads the input from the user and stores it in the variable named items. The user can enter multiple words separated by spaces, as this will all be captured as a single string in the items variable.

**echo "Items: \${items[@]}"**: This line prints "Items:" followed by the contents of the items variable. The `${items[@]}` syntax treats each word in the items variable as a separate element, preserving the spaces in the output. If we just used `$items`, it would only print the first word.



**Writing the script:**

**#!/bin/bash**

**echo "Enter a list of items:"**

**read -a items**

**echo "Items: \${items[@]}"**

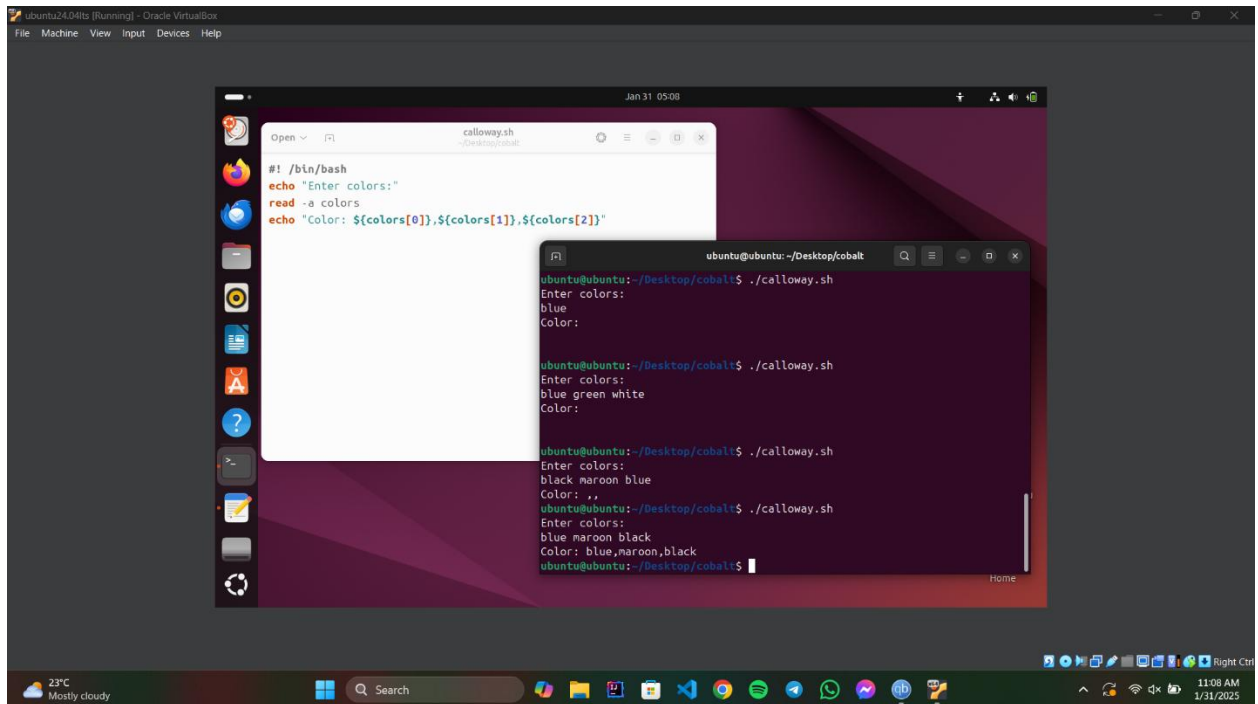
**Output:**

Enter a list of items:

Lily Rose Daisy

Items: Lily Rose Daisy

## Practice:



The screenshot shows a VirtualBox window titled 'ubuntu24.04ts [Running] - Oracle VM VirtualBox'. Inside the window is an Ubuntu desktop environment. A terminal window is open, showing the execution of a script named 'calloway.sh'. The script prompts the user to enter colors and then displays them in a formatted way. The script content is visible in a separate window titled 'calloway.sh'.

```
#! /bin/bash
echo "Enter colors:"
read -a colors
echo "Color: ${colors[0]},${colors[1]},${colors[2]}"
```

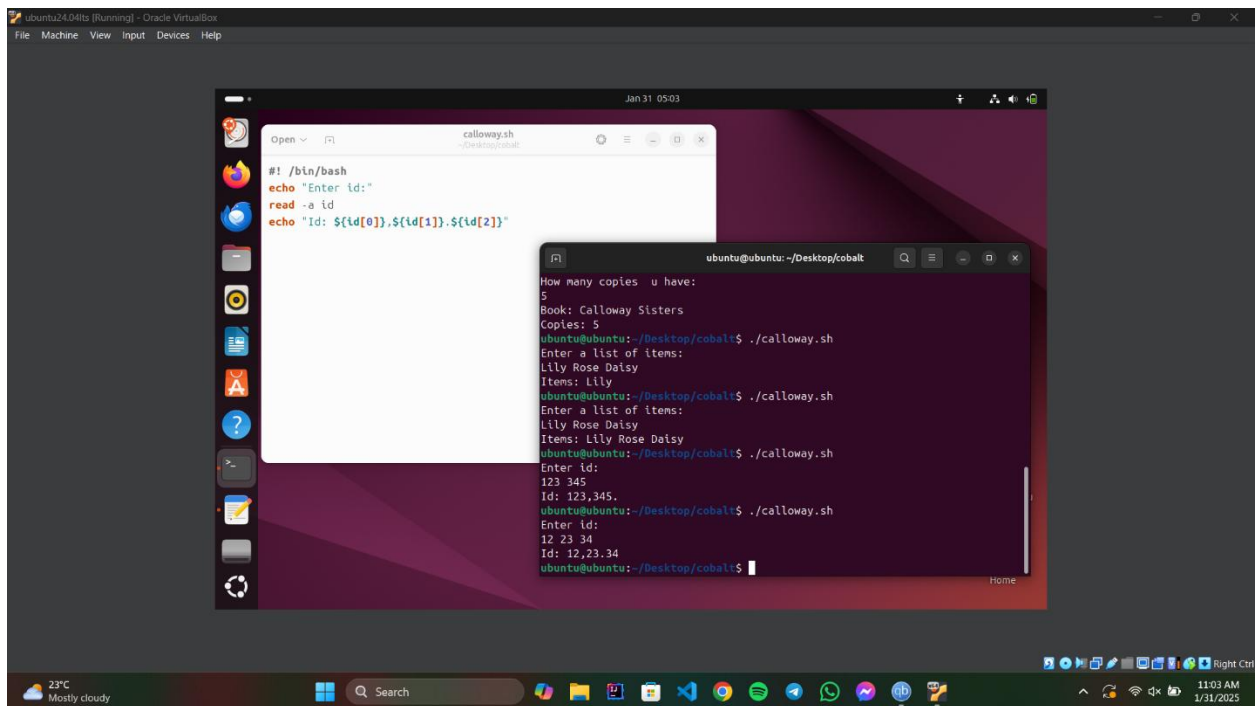
Terminal output:

```
ubuntu@ubuntu: ~/Desktop/cobalt
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter colors:
blue
Color:

ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter colors:
blue green white
Color:

ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter colors:
black maroon blue
Color: ,,

ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter colors:
blue maroon black
Color: blue,maroon,black
ubuntu@ubuntu:~/Desktop/cobalt$
```



The screenshot shows a VirtualBox window titled 'ubuntu24.04ts [Running] - Oracle VM VirtualBox'. Inside the window is an Ubuntu desktop environment. A terminal window is open, showing the execution of a script named 'calloway.sh'. The script prompts the user to enter a list of items and then displays them in a formatted way. The script content is visible in a separate window titled 'calloway.sh'.

```
#!/bin/bash
echo "Enter id:"
read -a id
echo "Id: ${id[0]},${id[1]},${id[2]}"
```

Terminal output:

```
ubuntu@ubuntu: ~/Desktop/cobalt
How many copies u have:
5
Book: Calloway Sisters
Copies: 5
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter a list of items:
Lily Rose Daisy
Items: Lily
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter a list of items:
Lily Rose Daisy
Items: Lily Rose Daisy
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter id:
123 345
Id: 123,345.
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter id:
12 23 34
Id: 12,23,34
ubuntu@ubuntu:~/Desktop/cobalt$
```

## **If Statement:**

**echo "Enter number: ":** This line displays the message "Enter number: " on the console, prompting the user to input a number.

**read num:** This line reads the number entered by the user and stores it in the variable num.

**if (( \$num > 18 )); then:** This line begins an if statement that checks if the value of num is greater than 18. The double parentheses (( ... )) are used for arithmetic comparisons in Bash.

**echo "\$num can vote":** If the condition in the if statement is true (i.e., num is greater than 18), this line is executed, displaying a message indicating that the number entered can vote.

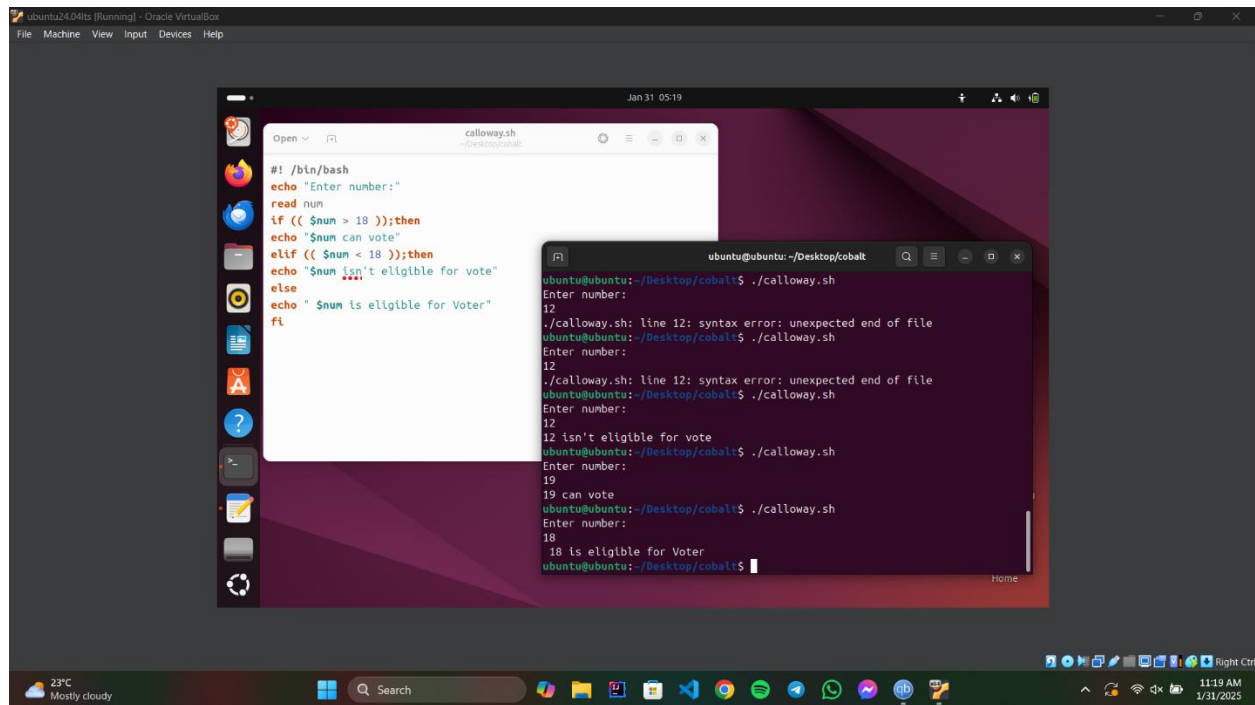
**elif (( \$num < 18 )); then:** This line introduces an elif (else if) condition, which is checked only if the initial if condition is false. It checks if num is less than 18.

**echo "\$num isn't eligible for vote":** If the elif condition is true (i.e., num is less than 18), this line is executed, displaying a message that the number is not eligible to vote.

**else:** This keyword introduces the else block, which is executed if neither the if nor the elif conditions are true.

**echo "\$num is eligible for Voter":** If num is neither greater than nor less than 18, it must be equal to 18. This line is executed, indicating that the number is eligible to vote (note the slight wording difference "eligible for Voter" instead of "eligible for vote").

**fi:** This keyword marks the end of the if statement block



## Writing the script:

**#!/bin/bash**

**echo "Enter number:"**

**read num**

**if (( \$num > 18 )); then**

**echo "\$num can vote"**

**elif (( \$num < 18 )); then**

**echo "\$num isn't eligible for vote"**

**else**

**echo**

**\$num is eligible for Voter"**



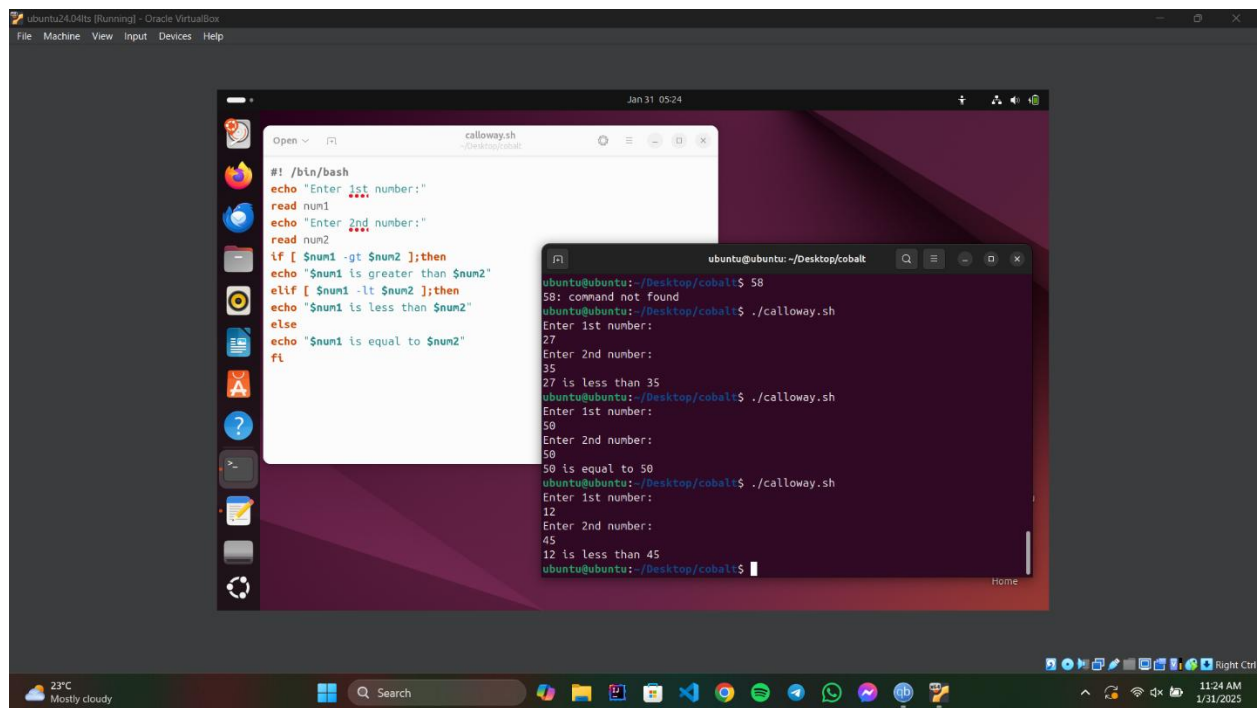
fi

## Output:

Enter number: 12

12 isn't eligible for vote

## Practice:



The screenshot shows a VirtualBox window titled 'ubuntu24.04ltz [Running] - Oracle VM VirtualBox'. Inside the window is an Ubuntu desktop environment. A terminal window is open, showing the execution of a script named 'calloway.sh'. The script prompts the user to enter two numbers and then compares them. The user has entered 12 for the first number and 45 for the second number. The script outputs '12 is less than 45'. A second window, titled 'calloway.sh', shows the script's source code. The script uses 'if', 'elif', and 'else' statements to compare the two numbers and print the result. The desktop background is a dark purple Ubuntu logo. The system tray at the bottom shows the date and time as 11:24 AM on 1/31/2025.

```
#! /bin/bash
echo "Enter 1st number:"
read num1
echo "Enter 2nd number:"
read num2
if [ $num1 -gt $num2 ];then
echo "$num1 is greater than $num2"
elif [ $num1 -lt $num2 ];then
echo "$num1 is less than $num2"
else
echo "$num1 is equal to $num2"
fi
```

```
ubuntu@ubuntu:~/Desktop/cobalt$ 58
58: command not found
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter 1st number:
27
Enter 2nd number:
35
27 is less than 35
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter 1st number:
50
Enter 2nd number:
50
50 is equal to 50
ubuntu@ubuntu:~/Desktop/cobalt$ ./calloway.sh
Enter 1st number:
12
Enter 2nd number:
45
12 is less than 45
ubuntu@ubuntu:~/Desktop/cobalt$
```

The screenshot shows a VirtualBox window titled 'ubuntu24.04ts [Running] - Oracle VM VirtualBox'. Inside the window is an Ubuntu desktop environment. A terminal window is open, displaying the execution of a script named 'calloway.sh'. The script prompts the user to 'enter number:' and then checks if the number is even or odd using an if statement. The terminal shows several test runs with inputs 6 and 5, resulting in 'Number is even' and 'Number is odd' respectively. A script editor window is also open, showing the source code of 'calloway.sh'.

```
#!/bin/bash
echo "enter number:"
read x
if [  $$(x \% 2)$  == 0 ];then
echo "Number is even"
else
echo "Number is odd"
fi
```

```
ubuntu@ubuntu: ~/Desktop
ubuntu@ubuntu:~/Desktop$ ./calloway.sh
enter number:
6
Number is even
ubuntu@ubuntu:~/Desktop$ ./calloway.sh
enter number:
5
Number is odd
ubuntu@ubuntu:~/Desktop$ 6
6: command not found
ubuntu@ubuntu:~/Desktop$ 6
6: command not found
ubuntu@ubuntu:~/Desktop$ ./calloway.sh
enter number:
5
Number is odd
ubuntu@ubuntu:~/Desktop$ ./calloway.sh
enter number:
6
Number is even
ubuntu@ubuntu:~/Desktop$
```

## Conclusion:

This lab demonstrated the effective use of arrays to manage multiple inputs and the application of if statements for decision-making. Arrays allowed for structured organization and processing of data, while conditional logic enabled actions based on specific criteria. Together, these concepts provided valuable insight into input handling and logical programming, essential for operating system tasks. The lab enhanced coding proficiency and emphasized the significance of data management and decision-making in system development.

