📘 LAB2 – Shell Scripting Basics

This lab demonstrates running and analyzing two simple shell scripts from the `Scripts/` folder:

- `print_numbers.sh`
- `array_loop.sh`

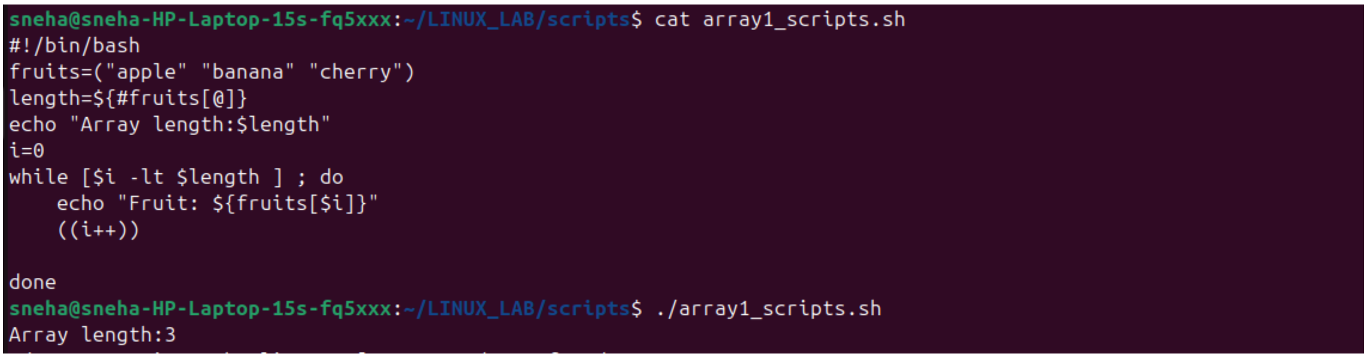
We will explore their purpose, line-by-line explanations, and example outputs.

🔹 Script 1: `print_numbers.sh`

🎯 Purpose

To print numbers from **1 to 5** using a simple loop.

📝 Code Snippet (with output)



```
sneha@sneha-HP-Laptop-15s-fq5xxx:~/LINUX_LAB/scripts$ cat array1_scripts.sh
#!/bin/bash
fruits=("apple" "banana" "cherry")
length=${#fruits[@]}
echo "Array length:$length"
i=0
while [$i -lt $length ] ; do
    echo "Fruit: ${fruits[$i]}"
    ((i++))

done
sneha@sneha-HP-Laptop-15s-fq5xxx:~/LINUX_LAB/scripts$ ./array1_scripts.sh
Array length:3
```

📖 Line-by-Line Explanation

Line	Code	Explanation
1	`#!/bin/bash`	Shebang – tells the system to use the Bash shell.
2	`fruits=("apple" "banana" "cherry")`	Creates an array named `fruits` with 3 elements: `apple`, `banana`, and `cherry`.
3	`length=${#fruits[@]}`	Finds the **length of the array** using `${#array[@]}`. Here, `length = 3`.
4	`echo "Array length:$length"`	Prints the length of the array.
5	`i=0`	Initializes the loop counter `i` to 0 (first index).
6	`while [ $i -lt $length ] ; do`	Starts a `while` loop that runs **while `i` is less than the array length**.
7	`echo "Fruit: ${fruits[$i]}"`	Prints the element of the array at index `i`.

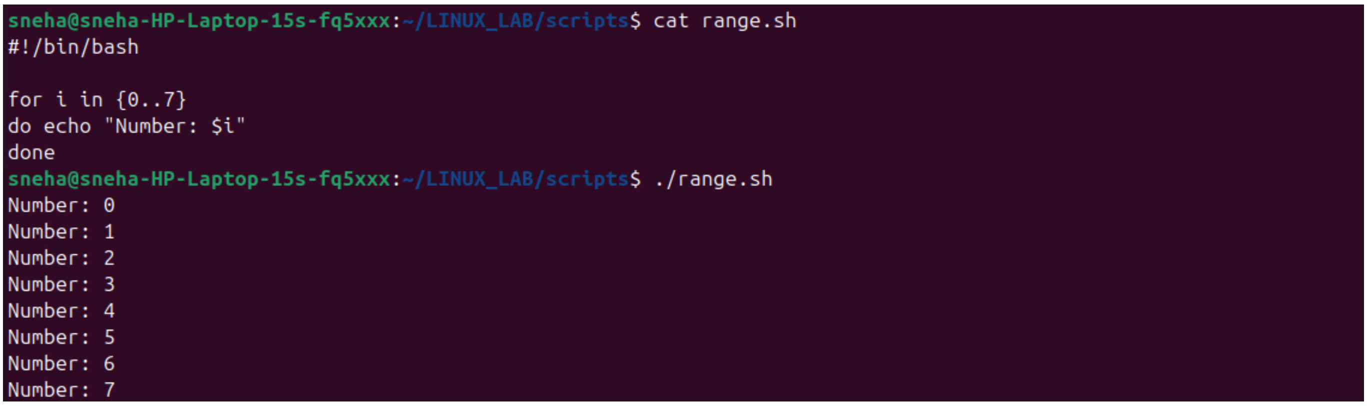
Line	Code	Explanation
8	`((i++))`	Increments `i` by 1 after each iteration.
9	`done`	Marks the end of the `while` loop.

🔹 Script 2: array_loop.sh

🎯 Purpose

To demonstrate a for loop with a numeric range in Bash.

📝 Code Snippet (with output)



```
sneha@sneha-HP-Laptop-15s-fq5xxx:~/LINUX_LAB/scripts$ cat range.sh
#!/bin/bash

for i in {0..7}
do echo "Number: $i"
done
sneha@sneha-HP-Laptop-15s-fq5xxx:~/LINUX_LAB/scripts$ ./range.sh
Number: 0
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
Number: 6
Number: 7
```

📖 Line-by-Line Explanation

Line	Code	Explanation
1	`#!/bin/bash`	Shebang → ensures the script runs with the Bash shell.
2	*(blank)*	Just an empty line for readability.
3	`for i in {0..7}`	Starts a `for` loop. The loop variable `i` will take values from `0` to `7` (inclusive).
4	`do`	Marks the beginning of the loop body.
5	`echo "Number: $i"`	Prints the current value of `i` prefixed with `"Number:"`.
6	`done`	Ends the `for` loop.

❓ Extra Questions

Q1. What is the purpose of #!/bin/bash?

- It is called the shebang. It tells the system to run the script using the Bash shell, ensuring correct execution.

Q2. How do you make a script executable?

- chmod +x scriptname.sh
- Example:

```
chmod +x print_numbers.sh
./print_numbers.sh
```

✨ Summary

- `print_numbers.sh` → Demonstrates a basic for loop.
- `array_loop.sh` → Demonstrates using arrays in Bash.
- `#!/bin/bash` → Ensures the script runs in the Bash shell.
- `chmod +x` → Makes a script executable.