

# LAB 3 – Enhanced Number Script

## 📋 Original Behavior

```
GNU nano 1.4                           ENHANCED_NUMBER_S.SH  
#!/bin/bash  
  
# Get inputs from user  
read -p "Enter start number: " start  
read -p "Enter end number: " end  
read -p "Enter step (positive number): " step  
  
# Validate step  
if ! [[ "$step" =~ ^[0-9]+$ ]] || [ "$step" -le 0 ]; then  
    echo "Error: Step must be a positive integer."  
    exit 1  
fi  
  
# Loop from start to end with step  
for (( i=start; i<=end; i+=step ))  
do  
    echo "Number: $i"  
done
```

Explanation:

⌚ User input collection These read commands take input from the user:

start: the beginning of the number range

end: the maximum value to count to

step: how much to increase each time

☑ Step validation

"\$step" =~ ^[0-9]+\$ → Ensures the input contains only digits (i.e., it's a number).

"\$step" -le 0 → Checks if the number is less than or equal to zero.

If either check fails, the script shows an error and exits.

⌚ C-style for loop

Starts from i = start

Loops until i <= end

Increments i by step in each iteration

Output inside loop

Prints the current number in the loop.

Loop continues based on step.

OUTPUT:

```
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ cd scripts
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ touch print_numbers.sh
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ nano print_numbers.sh
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ nano print_numbers.sh
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ chmod enhanced_numbers.sh
chmod: missing operand after 'enhanced_numbers.sh'
Try 'chmod --help' for more information.
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ chmod 777 enhanced_numbers.sh
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ ./enhanced_numbers.sh
Enter start number: 2
Enter end number: 10
Enter step (positive number): 2
Number: 2
Number: 4
Number: 6
Number: 8
Number: 10
bhaktib0108@bhaktib0108-VirtualBox:~/scripts$ █
```

Script: `print_numbers.sh`

- Printed numbers from 1 to 5 using a hardcoded `for` loop:

```
for i in 1 2 3 4 5
do
    echo "Number: $i"
done

Q1 Difference between $1, $@, and $# in bash?
A1 The difference between $1, $@ and $# is:
• $1 - Takes the first argument.
• $@ - Takes all the arguments separately.
• $# - Number of arguments

Q2 What does exit 1 mean in a script?
A2
• To stop execution when an error is detected.
• To inform other programs or scripts that this script failed or encountered a problem.
```

