Selected files

1 printable files

LAB 3 .md

LAB 3 .md

Objective: Enhance and customize a script.

print_numbers.sh

Purpose: The original script prints numbers in a sequence (e.g., 1 to 10). 1️⃣ Original Script (print_numbers.sh) #!/bin/bash

Original script: prints numbers from 1 to 10

```
for i in {1..10}
do
    echo $i
done
```

Behavior: Automatically prints numbers from 1 to 10. User cannot control start, end, or step.

▶️ Example Run:

$ ./print_numbers.sh 1 2 3 4 5 6 7 8 9 10

2️⃣ Modified Script (enhanced_numbers.sh)

```
#!/bin/bash
#Enhanced script: prints numbers from user-provided start to end with a step

#Take inputs
read -p "Enter start value: " start
read -p "Enter end value: " end
read -p "Enter step value: " step

#Validate inputs
if [ "$step" -le 0 ]; then
    echo "Error: Step must be a positive number."
    exit 1
fi
```

```
if [ "$start" -gt "$end" ]; then
    echo "Error: Start must be less than or equal to end."
    exit 1
fi

#Print numbers
for ((i=$start; i<=$end; i+=$step))
do
    echo $i
done
```

⚙️ New Behavior: User chooses start, end, and step values. Script checks if: step > 0 start <= end Then prints sequence accordingly. 3️⃣ Example Runs ✅ Case 1: Normal input $ ./enhanced_numbers.sh Enter start value: 1 Enter end value: 10 Enter step value: 2 1 3 5 7 9 ✅ Case 2: Different range

```
$ ./enhanced_numbers.sh
Enter start value: 5
Enter end value: 20
Enter step value: 5
5
10
15
20
```

❌ Case 3: Invalid step

```
$ ./enhanced_numbers.sh
Enter start value: 1
Enter end value: 10
Enter step value: -2
Error: Step must be a positive number.
```

❌ Case 4: Start > End

```
$ ./enhanced_numbers.sh
Enter start value: 20
Enter end value: 10
Enter step value: 2
Error: Start must be less than or equal to end.
✅ This shows how the script evolved from fixed numbers → user-controlled, validate
```

Extra Questions:

Difference between $1, $@, and $# in bash?

- $1 → The first argument
- $@ → All arguments, as separate words
- $# → The number of arguments passed

What does exit 1 mean in a script?

- exit in bash ends the script immediately.
- The number after it (0, 1, 2, etc.) is the exit status code.