

Assignment-2: Shell Scripting

Name: K M Chandrashekhar

Batch: DevOps Batch – 1

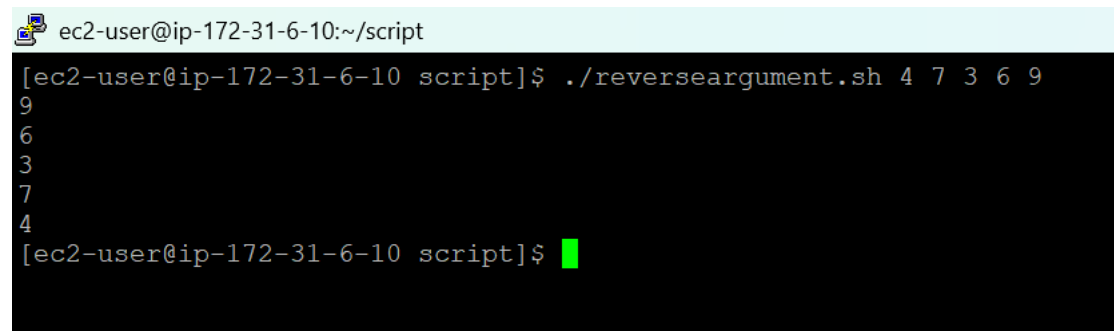
Date: May 2025

1. Write a shell script to print arguments passed to the script in reverse order.

Ans : vi reverseargument.sh

```
#!/bin/bash
# Loop through arguments in reverse
for (( i=$#; i>0; i-- ))
do
    echo "${!i}"
done
```

OUTPUT



```
ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ ./reverseargument.sh 4 7 3 6 9
9
6
3
7
4
[ec2-user@ip-172-31-6-10 script]$
```

2. Write a script that monitors system performance and sends a mail notification with warning messages about resource usage like ram, disk usage

Ans: vi ram_disk_usage_monitor.sh

ec2-user@ip-172-31-6-10:~/script

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

```
TO="kmchandrashekhar.iitkgp@gmail.com"
```

```
RAM_THRESHOLD=1
```

```
DISK_THRESHOLD=1
```

```
ram_usage=$(free | awk '/Mem:/ {printf("%.0f", $3/$2 * 100)}')
```

```
disk_usage=$(df / | awk 'NR==2 {print $5}' | sed 's/%//')
```

```
message=""
```

```
if [ "$ram_usage" -ge "$RAM_THRESHOLD" ]; then
    message+="WARNING: RAM usage is at ${ram_usage}%\n"
fi
```

```
if [ "$disk_usage" -ge "$DISK_THRESHOLD" ]; then
    message+="WARNING: Disk usage is at ${disk_usage}%\n"
fi
```

```
echo "RAM usage: $ram_usage%"
```

```
echo "DISK usage: $disk_usage%"
```

```
echo -e "Message:\n$message"
```

```
# Send mail
```

```
if [ -n "$message" ]; then
    echo -e "$message" | mail -s "System Resource Warning on $(hostname)" "$TO"
    echo "Mail sent to $TO"
else
```

```
    echo "No warning. No mail sent."
fi
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"ram_disk_usage_monitor.sh" 32L, 724B
```

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ ./ram_disk_usage_monitor.sh
```

```
RAM usage: 35%
```

```
DISK usage: 17%
```

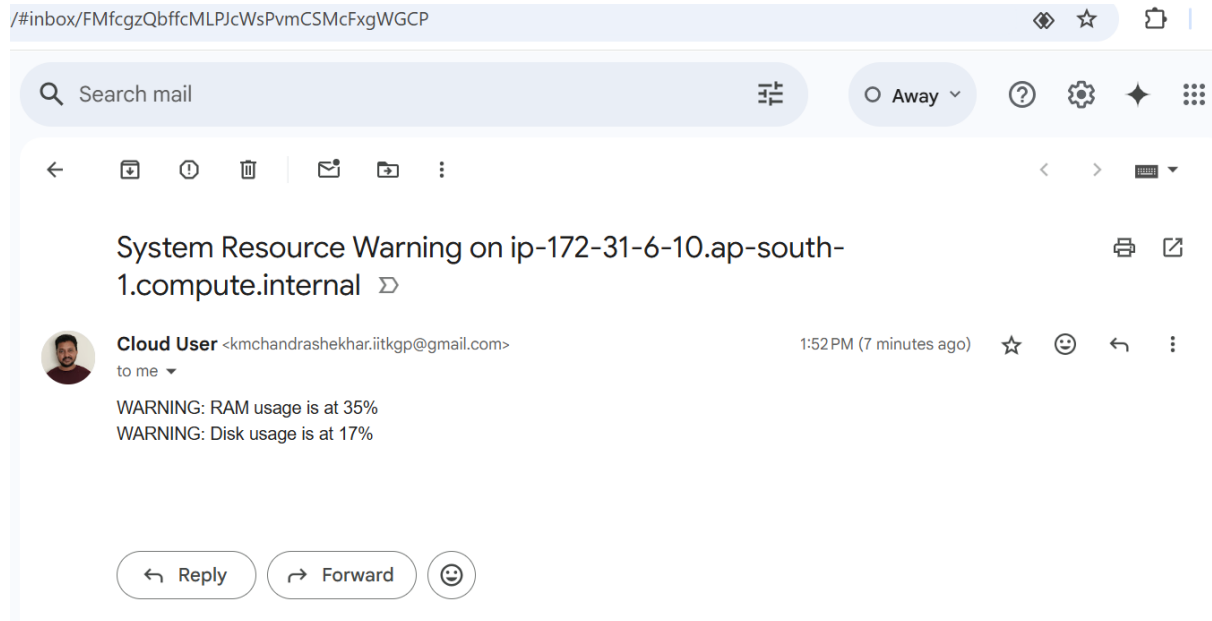
```
Message:
```

```
WARNING: RAM usage is at 35%
```

```
WARNING: Disk usage is at 17%
```

```
Mail sent to kmchandrashekhar.iitkgp@gmail.com
```

```
[ec2-user@ip-172-31-6-10 script]$
```



```
ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ ./ram_disk_usage_monitor.sh
RAM usage: 35%
DISK usage: 17%
Message:
WARNING: RAM usage is at 35%
WARNING: Disk usage is at 17%

Mail sent to kmchandrashekhar.iitkgp@gmail.com
[ec2-user@ip-172-31-6-10 script]$ vi ram_disk_usage_monitor.sh
[ec2-user@ip-172-31-6-10 script]$ ./ram_disk_usage_monitor.sh
RAM usage: 35%
DISK usage: 17%
Message:

No warning. No mail sent.
[ec2-user@ip-172-31-6-10 script]$
```

3. Write a script to find the files in a directory that were last modified more than 4 weeks ago. delete them, starting with the ones that consume the most space.

Ans: touch -d "30 days ago" file{1..5}.txt

```
ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 28
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
-rw-r--r--. 1 ec2-user ec2-user 215 May 10 08:49 file1.txt
-rw-r--r--. 1 ec2-user ec2-user 308 May 10 08:49 file2.txt
-rw-r--r--. 1 ec2-user ec2-user  14 May 10 08:49 file3.txt
-rw-r--r--. 1 ec2-user ec2-user  22 May 10 08:49 file4.txt
-rw-r--r--. 1 ec2-user ec2-user  11 May 10 08:49 file5.txt
[ec2-user@ip-172-31-6-10 script]$
```

vi delete_4week_old_files.sh

ec2-user@ip-172-31-6-10:~/script

#!/bin/bash

```
# Set the directory (current directory)
TARGET_DIR="."
```

```
# Find files older than 28 days, sort by size (biggest first), then delete them
find "$TARGET_DIR" -maxdepth 1 -type f -mtime +28 -printf "%s %p\n" 2>/dev/null | \
sort -nr | \
awk '{print $2}' | \
while read -r file; do
    if [[ -f "$file" ]]; then
        echo "Deleting: $file"
        rm -f "$file"
    fi
done
```

ec2-user@ip-172-31-6-10:~/script

[ec2-user@ip-172-31-6-10 script]\$ vi delete_4week_old_files.sh

[ec2-user@ip-172-31-6-10 script]\$ ls -lt

```
total 32
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
-rw-r--r--. 1 ec2-user ec2-user 215 May 10 08:49 file1.txt
-rw-r--r--. 1 ec2-user ec2-user 308 May 10 08:49 file2.txt
-rw-r--r--. 1 ec2-user ec2-user  14 May 10 08:49 file3.txt
-rw-r--r--. 1 ec2-user ec2-user  22 May 10 08:49 file4.txt
-rw-r--r--. 1 ec2-user ec2-user  11 May 10 08:49 file5.txt
```

[ec2-user@ip-172-31-6-10 script]\$./delete_4week_old_files.sh

Deleting: ./file2.txt

Deleting: ./file1.txt

Deleting: ./file4.txt

Deleting: ./file3.txt

Deleting: ./file5.txt

[ec2-user@ip-172-31-6-10 script]\$ ls -lt

```
total 12
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
```

[ec2-user@ip-172-31-6-10 script]\$

4. Explain the difference between running a script with './script' and running it with 'nohup ./script &'.

Ans: ./Script

- Runs the script normally in the foreground.
- If you close the terminal, the script stops.

Eg: ./backup.sh

nohup ./script &

- nohup = "No hang up"
- Runs the script in the background, even after you close the terminal.
- Output goes to a file called nohup.out by default.

Eg: nohup ./backup.sh &

Vi demo.sh

ec2-user@ip-172-31-6-10:~/script

```
#!/bin/bash
for i in {1..5}
do
    echo "Running step $i at $(date)"
    sleep 5
done
```

~
~

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ vi demo.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x demo.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 16
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./demo.sh
Running step 1 at Mon Jun  9 09:19:07 UTC 2025
Running step 2 at Mon Jun  9 09:19:12 UTC 2025
Running step 3 at Mon Jun  9 09:19:17 UTC 2025
Running step 4 at Mon Jun  9 09:19:22 UTC 2025
Running step 5 at Mon Jun  9 09:19:27 UTC 2025
[ec2-user@ip-172-31-6-10 script]$
```

- You see the output in your terminal.
- If you close the terminal, the script stops.

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ vi demo.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x demo.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 16
-rw-----. 1 ec2-user ec2-user   0 Jun  9 09:23 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./demo.sh
Running step 1 at Mon Jun  9 09:24:56 UTC 2025
Running step 2 at Mon Jun  9 09:25:01 UTC 2025
Running step 3 at Mon Jun  9 09:25:06 UTC 2025
Running step 4 at Mon Jun  9 09:25:11 UTC 2025
Running step 5 at Mon Jun  9 09:25:16 UTC 2025
[ec2-user@ip-172-31-6-10 script]$ nohup ./demo.sh &
nohup: [1] 2158
[ec2-user@ip-172-31-6-10 script]$ ignoring input and appending output to 'nohup.out'
```

```

ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ vi demo.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x demo.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 16
-rw-----. 1 ec2-user ec2-user  0 Jun  9 09:23 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./demo.sh
Running step 1 at Mon Jun  9 09:24:56 UTC 2025
Running step 2 at Mon Jun  9 09:25:01 UTC 2025
Running step 3 at Mon Jun  9 09:25:06 UTC 2025
Running step 4 at Mon Jun  9 09:25:11 UTC 2025
Running step 5 at Mon Jun  9 09:25:16 UTC 2025
[ec2-user@ip-172-31-6-10 script]$ nohup ./demo.sh &
nohup: [1] 2158
[ec2-user@ip-172-31-6-10 script]$ ignoring input and appending output to 'nohup.out'
^C
[1]+  Done                  nohup ./demo.sh
[ec2-user@ip-172-31-6-10 script]$ cat nohup.out
Running step 1 at Mon Jun  9 09:25:42 UTC 2025
Running step 2 at Mon Jun  9 09:25:47 UTC 2025
Running step 3 at Mon Jun  9 09:25:52 UTC 2025
Running step 4 at Mon Jun  9 09:25:57 UTC 2025
Running step 5 at Mon Jun  9 09:26:02 UTC 2025
[ec2-user@ip-172-31-6-10 script]$ █

```

- Runs in the **background**.
- Keeps **running even if terminal is closed**.

5. Read an integer 'n' and generate the following pattern:

```

1
1 2
1 2 3
1 2 3 4
up to 'n' rows.

```

Ans : vi generate_pattern.sh

```

ec2-user@ip-172-31-6-10:~/script
#!/bin/bash

read -p "Enter a number: " n

for ((i=1; i<=n; i++))
do
    for ((j=1; j<=i; j++))
    do
        echo -n "$j "
    done
    echo
done █

```

```

ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ vi generate_pattern.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x generate_pattern.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 24
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun  9 10:12 generate_pattern.sh
-rw----- 1 ec2-user ec2-user 235 Jun  9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./generate_pattern.sh
Enter a number: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
[ec2-user@ip-172-31-6-10 script]$ █

```

6 . Write a script to display the contents of the file in reverse order without using tac command

Ans : vi reverse.txt

```

ec2-user@ip-172-31-6-10:~/script
Hello

This is Reverse file script

content should display in reverse order

done █
~
~
~

```

vi reversefile.sh

```

ec2-user@ip-172-31-6-10:~/script
#!/bin/bash

read -p "Enter filename: " file

if [[ -f "$file" ]]; then
    mapfile -t lines < "$file"

    for ((i=${#lines[@]}-1; i>=0; i--)); do
        echo "${lines[i]}"
    done
else
    echo "File not found!"
fi █
~

```

```

ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ vi reverse_file.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x reverse_file.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 32
-rw-r--r--. 1 ec2-user ec2-user 82 Jun 9 10:23 reverse.txt
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun 9 10:21 reverse_file.sh
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun 9 10:12 generate_pattern.sh
-rw-----. 1 ec2-user ec2-user 235 Jun 9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user 87 Jun 9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun 9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun 9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user 97 Jun 9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./reverse_file.sh
Enter filename: reverse.txt
done

content should display in reverse order

This is Reverse file script

Hello
[ec2-user@ip-172-31-6-10 script]$ █

```

7. Write a script to reverse a string using while loop Input: hello Output: olleh

Ans: vi reverse_string.sh

```

ec2-user@ip-172-31-6-10:~/script
#!/bin/bash

read -p "Enter a string: " str
rev=""

len=${#str}
i=$((len - 1))

while [ $i -ge 0 ]; do
    rev="$rev${str:$i:1}"
    ((i--))
done

echo "Reversed: $rev" █
~
~

```

```

ec2-user@ip-172-31-6-10:~/script
[ec2-user@ip-172-31-6-10 script]$ chmod u+x reverse_string.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 36
-rwxr--r--. 1 ec2-user ec2-user 166 Jun 9 10:35 reverse_string.sh
-rw-r--r--. 1 ec2-user ec2-user 82 Jun 9 10:23 reverse.txt
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun 9 10:21 reverse_file.sh
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun 9 10:12 generate_pattern.sh
-rw-----. 1 ec2-user ec2-user 235 Jun 9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user 87 Jun 9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun 9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun 9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user 97 Jun 9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./reverse_string.sh
Enter a string: HELLO
Reversed: OLLEH
[ec2-user@ip-172-31-6-10 script]$ █

```

8. Read an integer 'n' and generate the following pattern:

1

2 3

4 5 6

7 8 9 10

up to 'n' rows.

Ans : vi pattern_traingle.sh

ec2-user@ip-172-31-6-10:~/script

```
#!/bin/bash

read -p "Enter number of rows: " n
num=1

for ((i=1; i<=n; i++)); do
    for ((j=1; j<=i; j++)); do
        echo -n "$num "
        ((num++))
    done
    echo
done
```

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ vi pattern_triangle.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x pattern_triangle.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 40
-rwxr-xr-x. 1 ec2-user ec2-user 165 Jun  9 10:42 pattern_triangle.sh
-rwxr--r--. 1 ec2-user ec2-user 166 Jun  9 10:35 reverse_string.sh
-rw-r--r--. 1 ec2-user ec2-user  82 Jun  9 10:23 reverse.txt
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun  9 10:21 reverse_file.sh
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun  9 10:12 generate_pattern.sh
-rw-----. 1 ec2-user ec2-user 235 Jun  9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./pattern_triangle.sh
Enter number of rows: 6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
[ec2-user@ip-172-31-6-10 script]$
```

9. Write a script that renames a file or directory by replacing its letters with lowercase from uppercase letters. Example: TEst.txt → test.txt

Ans: touch Test.txt

Vi lowercase.sh

ec2-user@ip-172-31-6-10:~/script

```
#!/bin/bash

read -p "Enter file or directory name: " name

if [[ -e "$name" ]]; then
    lower_name=$(echo "$name" | tr 'A-Z' 'a-z')
    mv "$name" "$lower_name"
    echo "Renamed to: $lower_name"
else
    echo "File or directory not found!"
fi

~
~
~
```

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ vi lowercase.sh
[ec2-user@ip-172-31-6-10 script]$ chmod +x lowercase.sh
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 48
-rwxr-xr-x. 1 ec2-user ec2-user 238 Jun  9 11:06 lowercase.sh
-rw-r--r--. 1 ec2-user ec2-user   0 Jun  9 11:05 TEst.txt
-rw-r--r--. 1 ec2-user ec2-user 238 Jun  9 10:54 rename_lowercase.sh
-rwxr-xr-x. 1 ec2-user ec2-user 165 Jun  9 10:49 pattern_triangle.sh
-rwxr--r--. 1 ec2-user ec2-user 166 Jun  9 10:35 reverse_string.sh
-rw-r--r--. 1 ec2-user ec2-user  82 Jun  9 10:23 reverse.txt
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun  9 10:21 reverse_file.sh
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun  9 10:12 generate_pattern.sh
-rw-----. 1 ec2-user ec2-user 235 Jun  9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$ ./lowercase.sh
Enter file or directory name: TEst.txt
Renamed to: test.txt
[ec2-user@ip-172-31-6-10 script]$ ls -lt
total 48
-rwxr-xr-x. 1 ec2-user ec2-user 238 Jun  9 11:06 lowercase.sh
-rw-r--r--. 1 ec2-user ec2-user   0 Jun  9 11:05 test.txt
-rw-r--r--. 1 ec2-user ec2-user 238 Jun  9 10:54 rename_lowercase.sh
-rwxr-xr-x. 1 ec2-user ec2-user 165 Jun  9 10:49 pattern_triangle.sh
-rwxr--r--. 1 ec2-user ec2-user 166 Jun  9 10:35 reverse_string.sh
-rw-r--r--. 1 ec2-user ec2-user  82 Jun  9 10:23 reverse.txt
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun  9 10:21 reverse_file.sh
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun  9 10:12 generate_pattern.sh
-rw-----. 1 ec2-user ec2-user 235 Jun  9 09:26 nohup.out
-rwxr-xr-x. 1 ec2-user ec2-user  87 Jun  9 09:17 demo.sh
-rwxr--r--. 1 ec2-user ec2-user 386 Jun  9 08:58 delete_4week_old_files.sh
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun  9 08:40 ram_disk_usage_monitor.sh
-rwxr--r--. 1 ec2-user ec2-user  97 Jun  9 07:38 reverseargument.sh
[ec2-user@ip-172-31-6-10 script]$
```

10. Write a script that takes multiple inputs from the user as arguments and only prints the arguments if they are integers

Ans: vi print_integer.sh

ec2-user@ip-172-31-6-10:~/script

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

```
for arg in "$@"; do
    if [[ "$arg" =~ ^-?[0-9]+$ ]]; then
        echo "$arg"
    fi
done
```

```
~
```

```
~
```

ec2-user@ip-172-31-6-10:~/script

```
[ec2-user@ip-172-31-6-10 script]$ vi print_integer.sh
```

```
[ec2-user@ip-172-31-6-10 script]$ chmod +x print_integer.sh
```

```
[ec2-user@ip-172-31-6-10 script]$ ls -lt
```

```
total 52
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 98 Jun 9 11:16 print_integer.sh
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 238 Jun 9 11:06 lowercase.sh
```

```
-rw-r--r--. 1 ec2-user ec2-user 0 Jun 9 11:05 test.txt
```

```
-rw-r--r--. 1 ec2-user ec2-user 238 Jun 9 10:54 rename_lowercase.sh
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 165 Jun 9 10:49 pattern_triangle.sh
```

```
-rwxr--r--. 1 ec2-user ec2-user 166 Jun 9 10:35 reverse_string.sh
```

```
-rw-r--r--. 1 ec2-user ec2-user 82 Jun 9 10:23 reverse.txt
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 208 Jun 9 10:21 reverse_file.sh
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 137 Jun 9 10:12 generate_pattern.sh
```

```
-rw-----. 1 ec2-user ec2-user 235 Jun 9 09:26 nohup.out
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 87 Jun 9 09:17 demo.sh
```

```
-rwxr--r--. 1 ec2-user ec2-user 386 Jun 9 08:58 delete_4week_old_files.sh
```

```
-rwxr-xr-x. 1 ec2-user ec2-user 726 Jun 9 08:40 ram_disk_usage_monitor.sh
```

```
-rwxr--r--. 1 ec2-user ec2-user 97 Jun 9 07:38 reverseargument.sh
```

```
[ec2-user@ip-172-31-6-10 script]$ ./print_integer.sh asas 12 ff 54 1 2 3 4 dda eef nny mikm
```

```
12
```

```
54
```

```
1
```

```
2
```

```
3
```

```
4
```

```
[ec2-user@ip-172-31-6-10 script]$
```