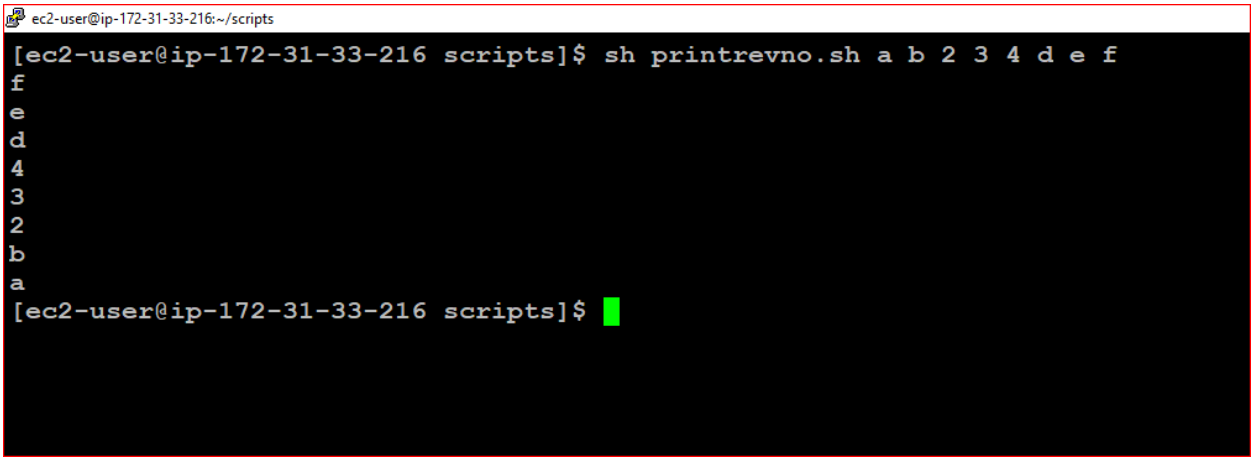
Assignment

Q.1. Write a shell script to print arguments passed to the script in reverse order. Input: sh script.sh 2 5 3 Output: 3 5 2

```
#!/bin/bash

for i in $*

do

temp=$#

done

for((i=$temp; i>0; i--))

do

  revstr=${!i}

  echo $revstr

done
```



Q.2. Write a shell script that checks if the openssh server is installed. If it is not installed, install it and change the SSH port to 2202. If it is already installed, change the SSH port to 2202 so clients can connect to that port.

```
#!/bin/bash rpm -q openssh if [ $? -ne 0 ];  then
 echo "Openssh is not installed..Installing now"
 sudo yum install openssh-server  -y
 else
 echo "Openssh-server is already installed"
fi
#changing port 22 to 2202
sudo sed -i 's/#Port 22/Port 2202/ ‘ /etc/ssh/sshd_config
 echo "Restarting  ssh..."
sudo systemctl restart ssh
```

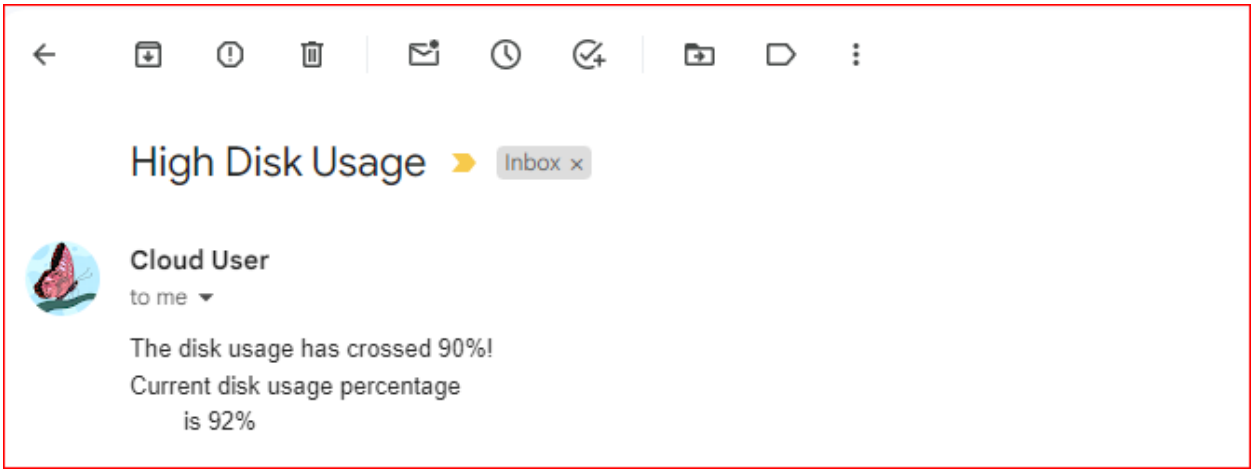
Q.3. Write a script that monitors system performance and sends a mail notification with warning messages about resource usage like ram, disk usage every 24 hours

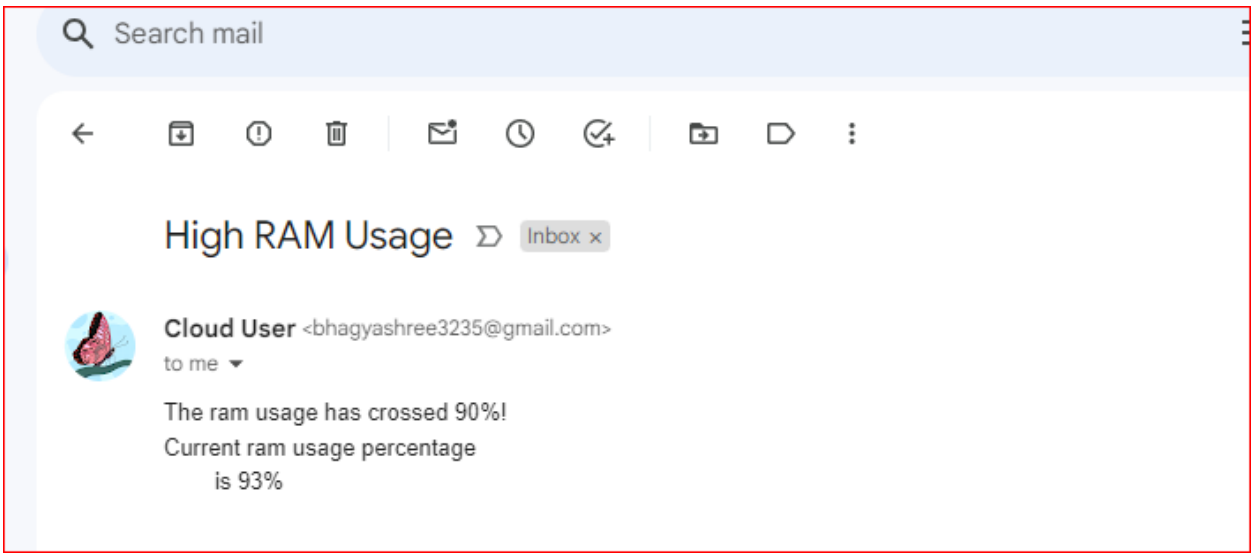
```
chk_ram_usage() {
ram_usage=$(free -m | awk '/Mem/{print int($3/$2*100)}')
 if [ "$ram_usage" -gt 30 ]; then
    echo -e "The ram usage has crossed 90%!\nCurrent ram usage percentage
    is $ram_usage" | mail -s "High RAM Usage" bhagyashree3235@gmail.com
 fi
}
```

```
chk_disk_usage() {
  disk_usage=$(df -h | awk '{print $5}' | sed 1d | sort -rh | awk '$1 > 30')
#if [ "$disk_usage" -gt 30 ]; then
  if [[ "$disk_usage" =~ ^[0-9]+$ && "$disk_usage" -gt 90 ]]; then
echo -e "The disk usage has crossed 90%!\nCurrent disk usage percentage
      is $disk_usage" | mail -s "High Disk Usage" bhagyashree3235@gmail.com
   fi
}
main(){
chk_disk_usage
chk_ram_usage
}
Main
```

Crontab -e:

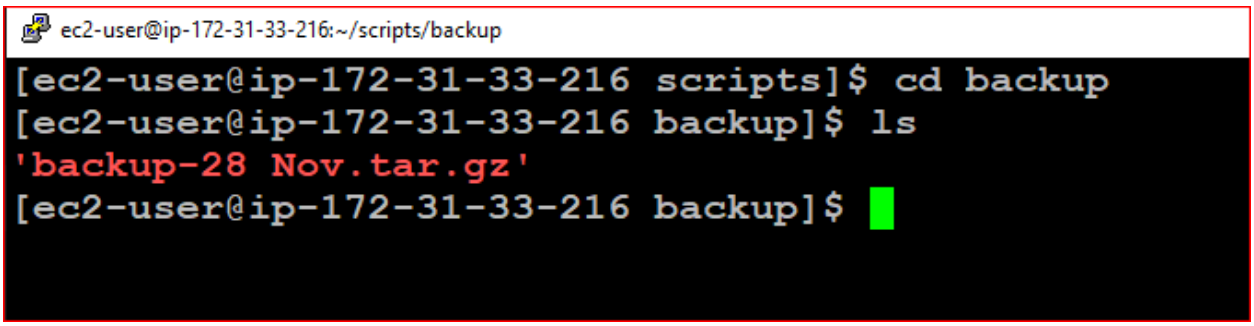
```
0 0 * * * /home/ec2-user/scripts/chusuagecp.sh
```





Q.4Write a script to take backups of a directory as tar zip file every weekend and store them as versions [Example backup-1.tar.gz, backup-2.tar.gz]

```
#!/bin/bash

file_to_backup="/home/ec2-user/scripts/dir1"

backup_directory="/home/ec2-user/scripts/backup"


version=$(date "+%d %b")

backup_filename="backup-$version.tar.gz"

tar -czf "$backup_directory/$backup_filename" "$file_to_backup"

echo "Backup created: $backup_filename"
```



Q.5. Write a script to take backups of a directory as tar zip file every weekend and store them as versions [Example backup-1.tar.gz, backup-2.tar.gz]

```
#!/bin/bash

echo "Enter the path of the dir to search"

 read  dirpath

find $dirPath -type f -mtime  +28 | xargs du -h | sort -r | awk '{print $2}' | xargs rm
```

Q.6. Explain the difference between running a script with ' ./script' and running it with ' nohup ./script &'

- **./script :**

  - ./script, it is the use to run script in the foreground, and it's display output to terminal.

  - If you exit the script or close the terminal it will receive 'SIGHUP' signal by default this is terminate the script

  - After executed the script output is display in the terminal

- **nohup ./script & :**

  - nohup ./script &, here, nohup is a stands for 'No Hang Up'.

- nohup ./script &, it is the use to run script in the background, so output did not display in the terminal.

- nohup prevents the script from receiving the hangup signal (SIGHUP). This means the script continues running even if you close the terminal or log out.

- The script's output is redirected to a file named 'nohup.out' in the same directory where the 'nohup' command was executed.

- The script is assigned a job ID by the shell, and you typically see a message like [1] 12345 indicating the job ID and process ID.

Q.7. Read an integer ' n ' and generate the following pattern:

1

1 2

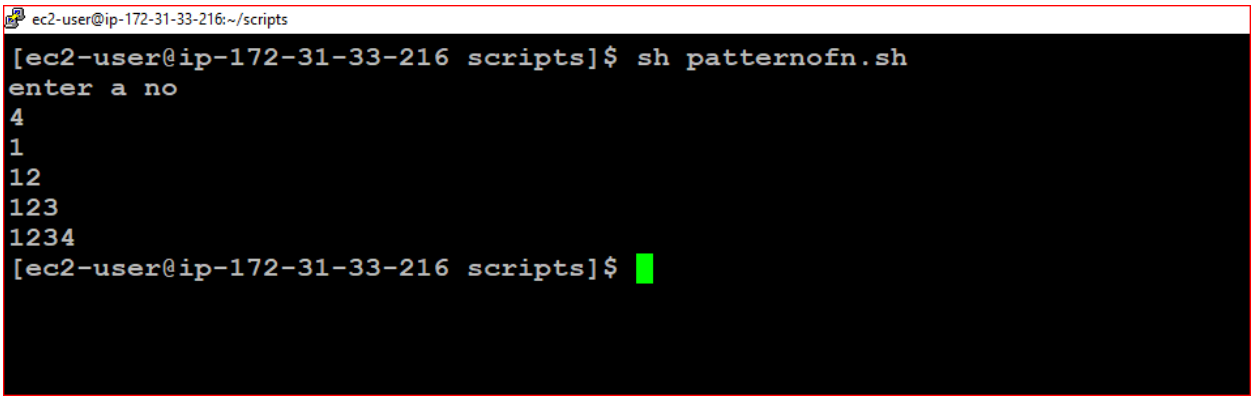
1 2 3

1 2 3 4

up to ' n ' rows

```
#!/bin/bash
echo "enter a no"
read n
for((i=1; i<=n; i++)); do
for((j=1; j<=i; j++)); do
  echo -n $j
done
echo
done
```

~

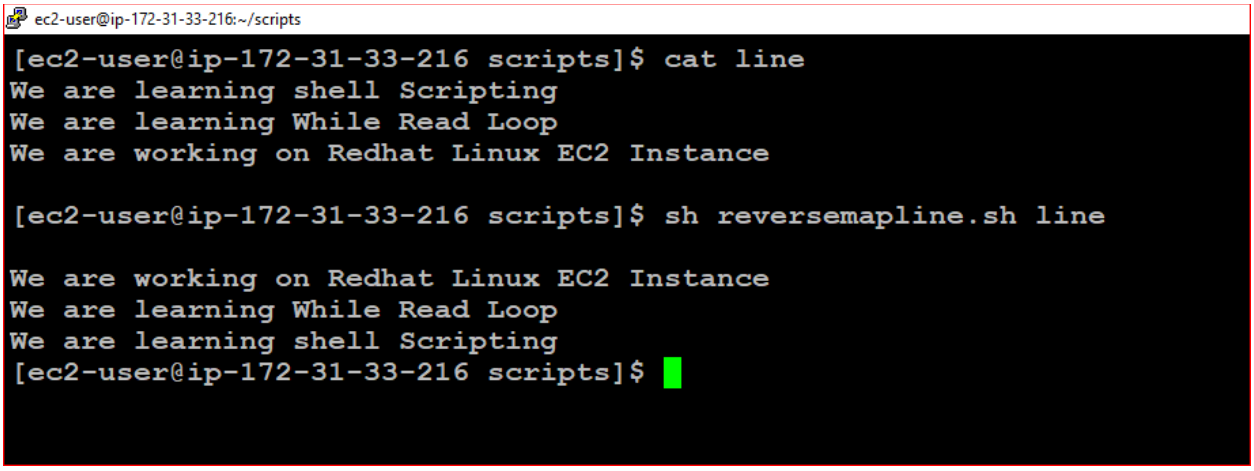


Q.8. Write a script to display the contents of the file in reverse order without using tac command

#!/bin/bash

filename="$1"

awk '{ lines[NR] = $0 } END { for (i = NR; i >= 1; i--) print lines[i] }' "$filename"

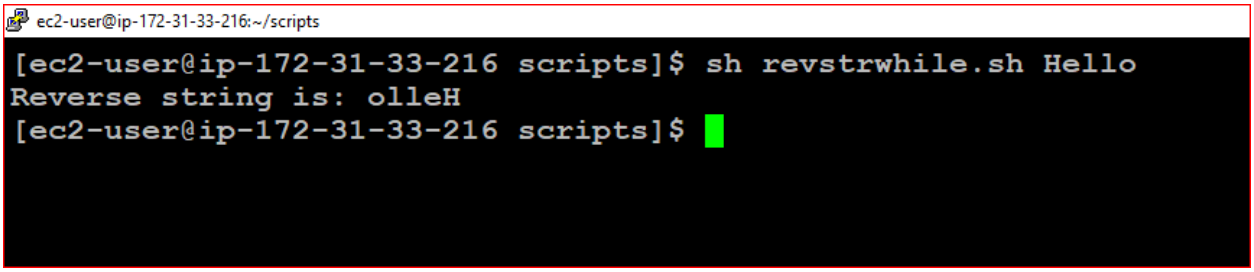


Q.9. Write a script to reverse a string using while loop

Input: hello

Output: olleh

```
#!/bin/bash
string="$1"
length=${#string}
while [ "$length" -gt 0 ]; do
   length=$((length-1))
   reverse=$length
echo “$reverse”
done
```



Q.10. Read an integer ' n ' and generate the following pattern:

1

2 3

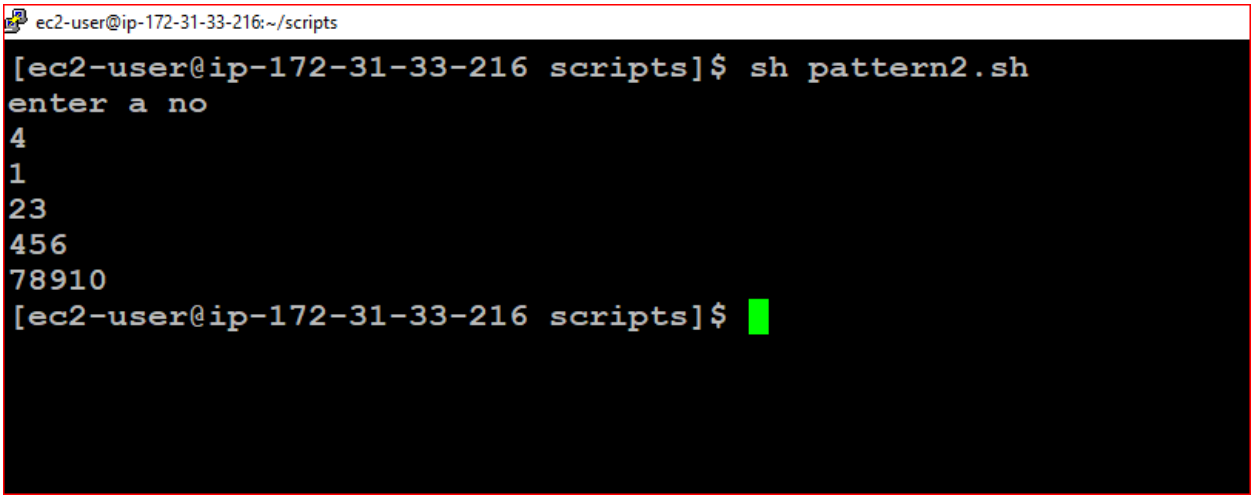
4 5 6

7 8 9 10

up to ' n ' rows.

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

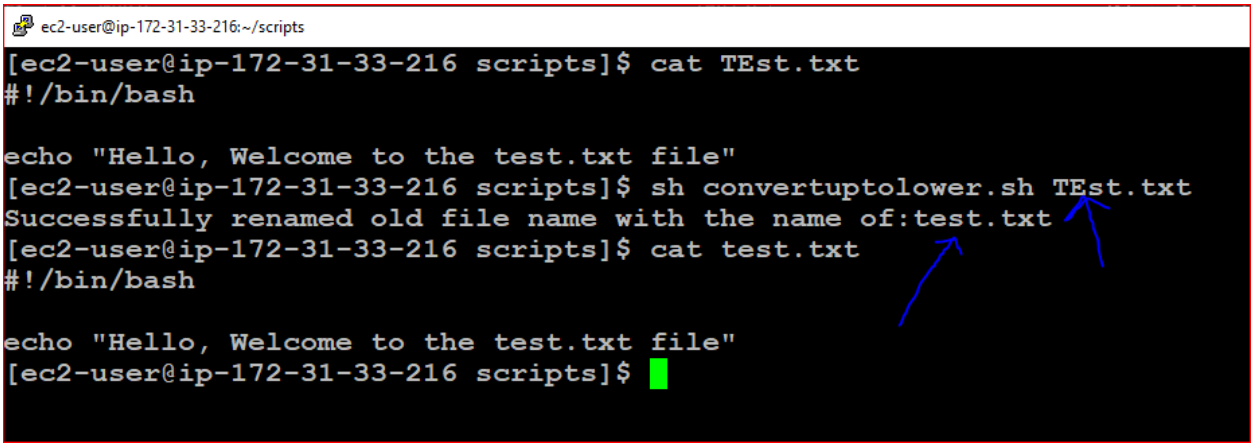
```
echo "enter a no"
read n
temp=1
for((i=1;i<=n;i++)); do
for((j=1;j<=i;j++)); do
   echo -n "$temp"
   temp=$((temp+1))
done
  echo
done
```



Q.11. Write a script that rename a file or directory by replacing its letters with lowercase from uppercase letters. Example: TEst.txt → test.txt

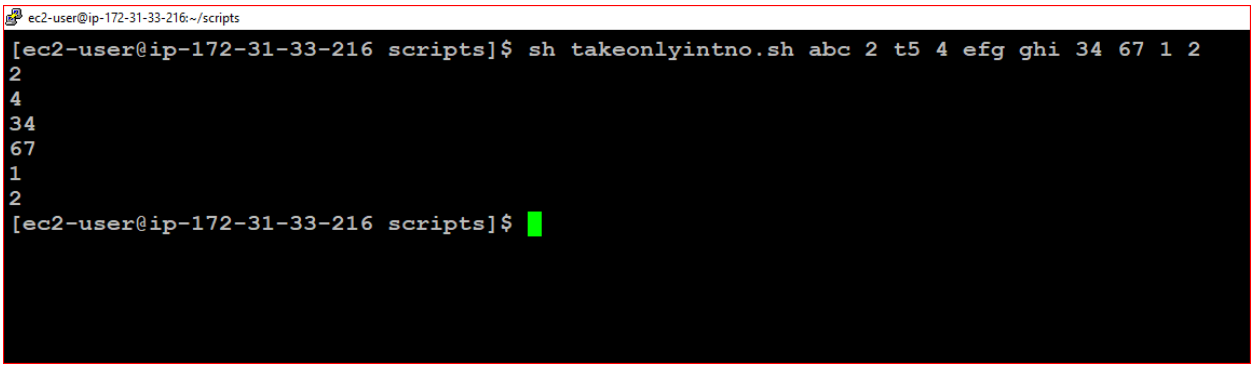
```
#!/bin/bash
filename="$1"
if [ ! -e "$filename" ]; then
```

```
    echo "File or directory not found: $filename"
    exit 1
fi
uppercase=$(echo $filename | tr 'A-Z' 'a-z')
mv "$filename" "$uppercase"
echo "Successfully renamed old file name with the name of:$uppercase"
```



Q.12. Write a script that takes multiple inputs from the user as arguments and only prints the arguments if they are integers

```
#!/bin/bash
for i in $@
do
if [[ $i =~ ^[0-9] ]]; then
 echo $i
fi
done
```


ec2-user@ip-172-31-33-216:~/scripts
[ec2-user@ip-172-31-33-216 scripts]$ sh takeonlyintno.sh abc 2 t5 4 efg ghi 34 67 1 2
2
4
34
67
1
2
[ec2-user@ip-172-31-33-216 scripts]$