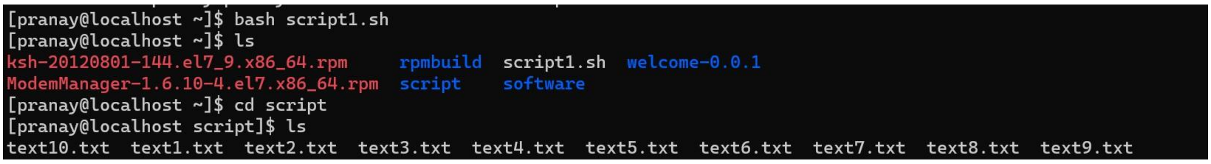
Shell Scripting Assignment - 1 (write the script files for the following)
Note: strictly don't use root user
Instructions
Write 2 script files:

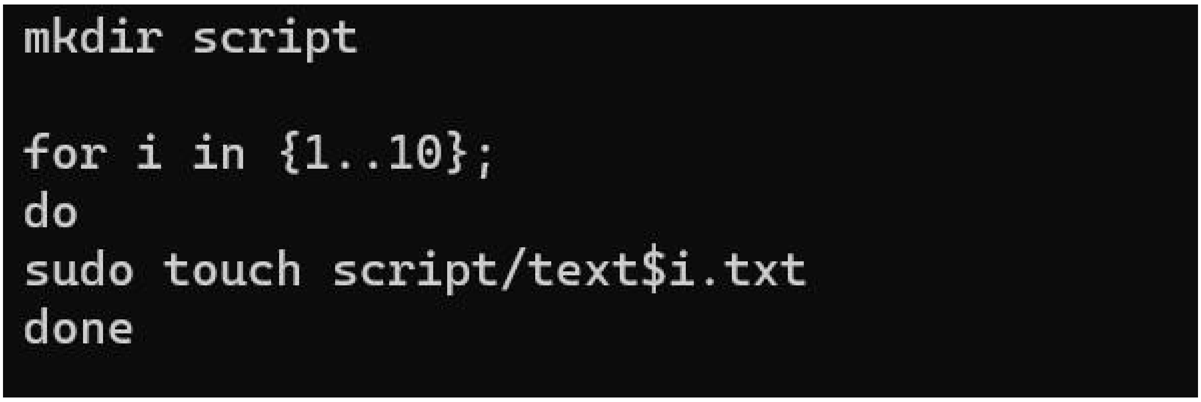
1. script1.sh

- When we execute script1.sh, it should create a folder called 'Script' in your home directory and then 10 files under the folder. The files will be like file1.txt, file2.txt....... File10.txt

A:



```
[pranay@localhost ~]$ bash script1.sh
[pranay@localhost ~]$ ls
ksh-20120801-144.el7_9.x86_64.rpm     rpmbuild  script1.sh  welcome-0.0.1
ModemManager-1.6.10-4.el7.x86_64.rpm  script    software
[pranay@localhost ~]$ cd script
[pranay@localhost script]$ ls
text10.txt  text1.txt  text2.txt  text3.txt  text4.txt  text5.txt  text6.txt  text7.txt  text8.txt  text9.txt
```



```
mkdir script

for i in {1..10};
do
sudo touch script/text$i.txt
done
```

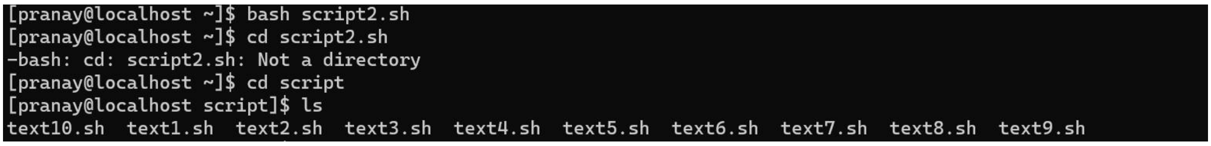
2. script2.sh

-When we execute script2.sh, it should rename all the .txt file into .sh file from the Script folder.

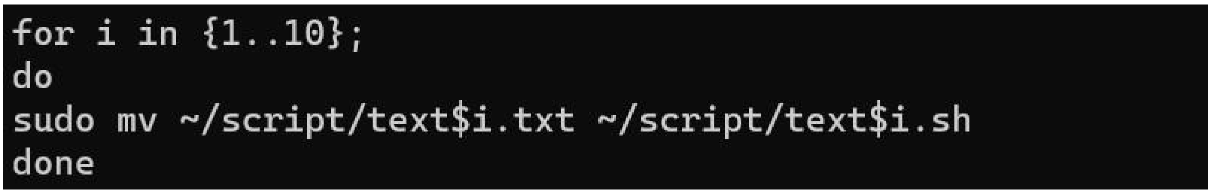
So the files will be like file1.sh, file2.sh,......... file10.sh

Note: You should not rename the files individually like mv file1.txt file1.sh

A:



```
[pranay@localhost ~]$ bash script2.sh
[pranay@localhost ~]$ cd script2.sh
-bash: cd: script2.sh: Not a directory
[pranay@localhost ~]$ cd script
[pranay@localhost script]$ ls
text10.sh  text1.sh  text2.sh  text3.sh  text4.sh  text5.sh  text6.sh  text7.sh  text8.sh  text9.sh
```



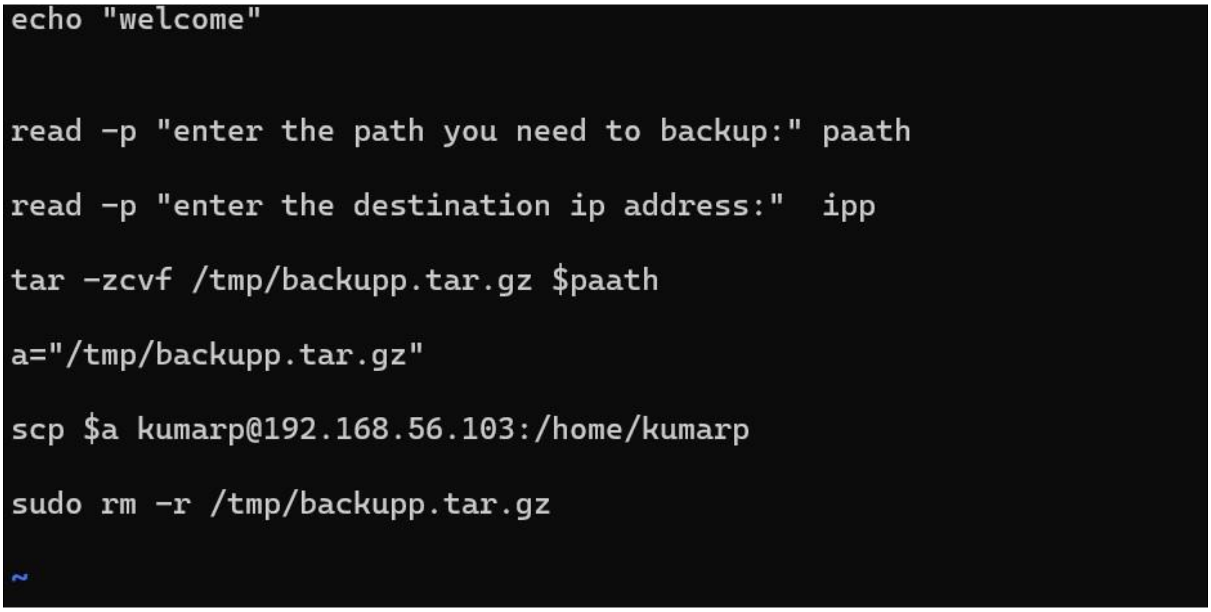
```
for i in {1..10};
do
sudo mv ~/script/text$i.txt ~/script/text$i.sh
done
```

Shell Scripting Assignment - 2 - Automating backups
Instructions

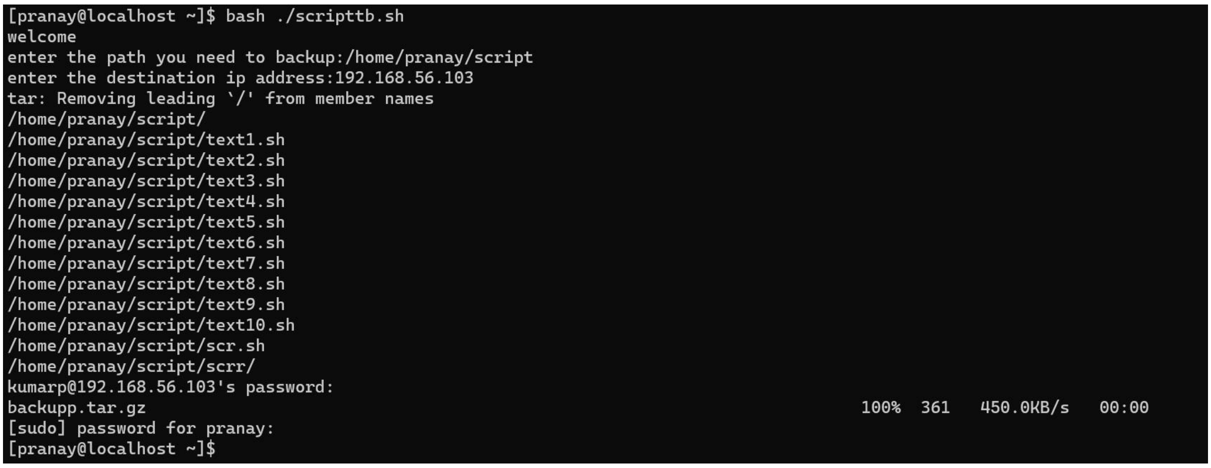
1. Script should prompt the user to provide an absolute path of the directories to take backup.
2. script should prompt the user to provide the destination server ip address to where the backup need to be transferred
3. create gzip compressed archive of all the given directories and store in /tmp
4. use the scp command to send the archive to the destination server
5. remove gzip archives from /tmp

Note: you can probably take a clone of the centos machine and use that as destination server

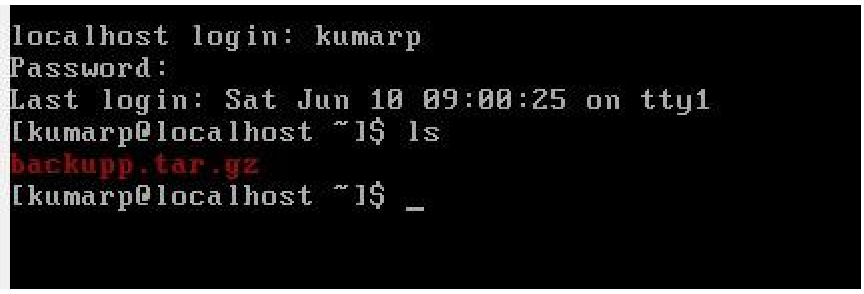
A:



```
echo "welcome"


read -p "enter the path you need to backup:" paath

read -p "enter the destination ip address:"  ipp

tar -zcvf /tmp/backupp.tar.gz $paath

a="/tmp/backupp.tar.gz"

scp $a kumarp@192.168.56.103:/home/kumarp

sudo rm -r /tmp/backupp.tar.gz

~
```



```
[pranay@localhost ~]$ bash ./scripttb.sh
welcome
enter the path you need to backup:/home/pranay/script
enter the destination ip address:192.168.56.103
tar: Removing leading `/' from member names
/home/pranay/script/
/home/pranay/script/text1.sh
/home/pranay/script/text2.sh
/home/pranay/script/text3.sh
/home/pranay/script/text4.sh
/home/pranay/script/text5.sh
/home/pranay/script/text6.sh
/home/pranay/script/text7.sh
/home/pranay/script/text8.sh
/home/pranay/script/text9.sh
/home/pranay/script/text10.sh
/home/pranay/script/scr.sh
/home/pranay/script/scrr/
kumarp@192.168.56.103's password:
backupp.tar.gz                                                      100%  361    450.0KB/s   00:00
[sudo] password for pranay:
[pranay@localhost ~]$
```



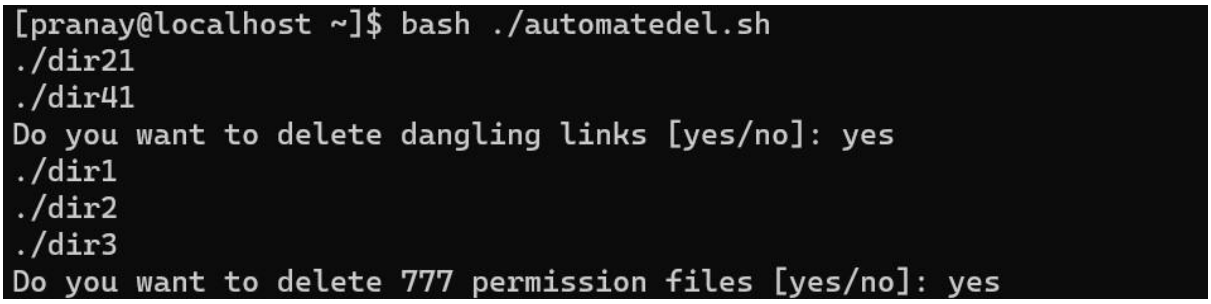
```
localhost login: kumarp
Password:
Last login: Sat Jun 10 09:00:25 on tty1
[kumarp@localhost ~]$ ls
backupp.tar.gz
[kumarp@localhost ~]$ _
```

Shell Scripting Assignment - 3

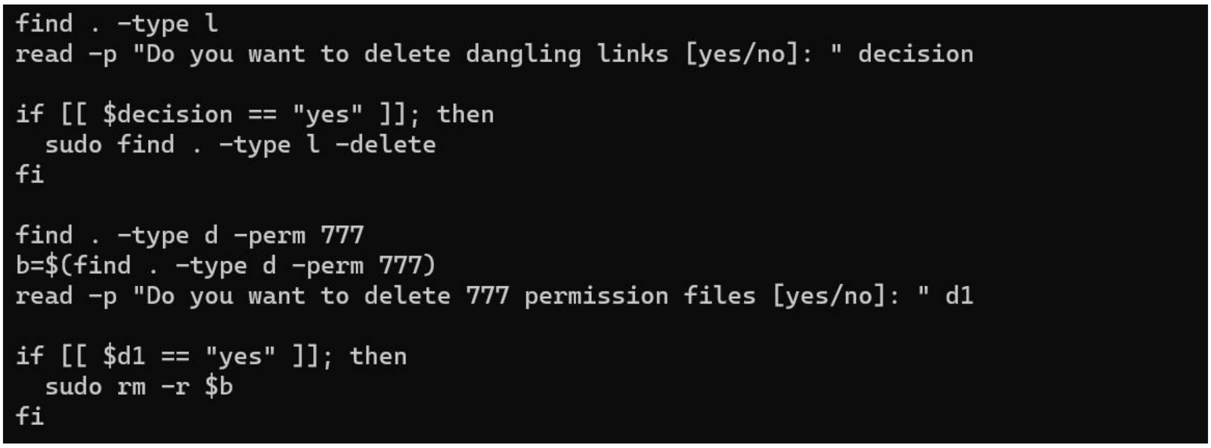
Write a Shell script to (do not perform as root)

1. Find and delete broken symlinks
2. Delete files which has 777 permissions that you have created
3. It should ask for confirmation before deleting files

A:



```
[pranay@localhost ~]$ bash ./automatedel.sh
./dir21
./dir41
Do you want to delete dangling links [yes/no]: yes
./dir1
./dir2
./dir3
Do you want to delete 777 permission files [yes/no]: yes
```



```
find . -type l
read -p "Do you want to delete dangling links [yes/no]: " decision

if [[ $decision == "yes" ]]; then
  sudo find . -type l -delete
fi

find . -type d -perm 777
b=$(find . -type d -perm 777)
read -p "Do you want to delete 777 permission files [yes/no]: " d1

if [[ $d1 == "yes" ]]; then
  sudo rm -r $b
fi
```

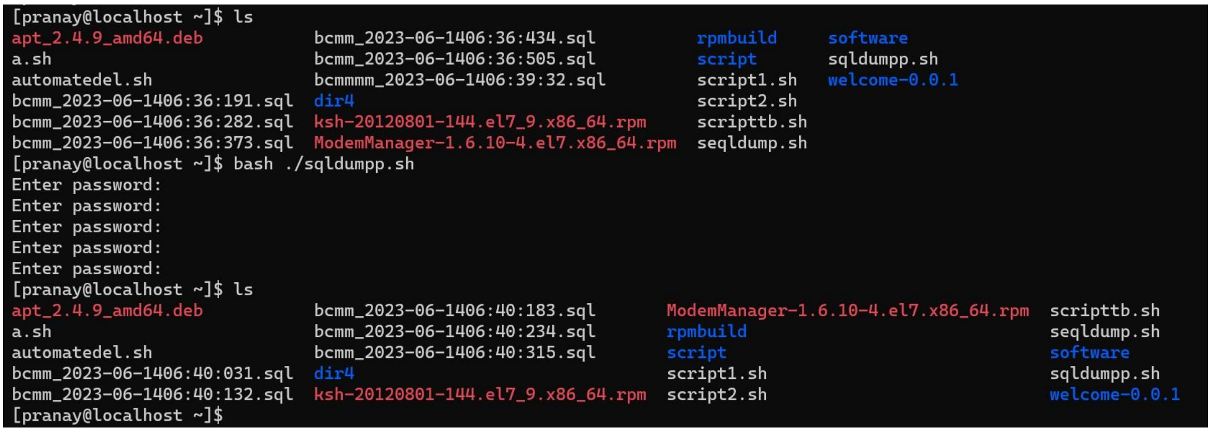
Shell Scripting Assignment - 4 Perform sql dump using the script

1. Perform sql dump 5 times using loop

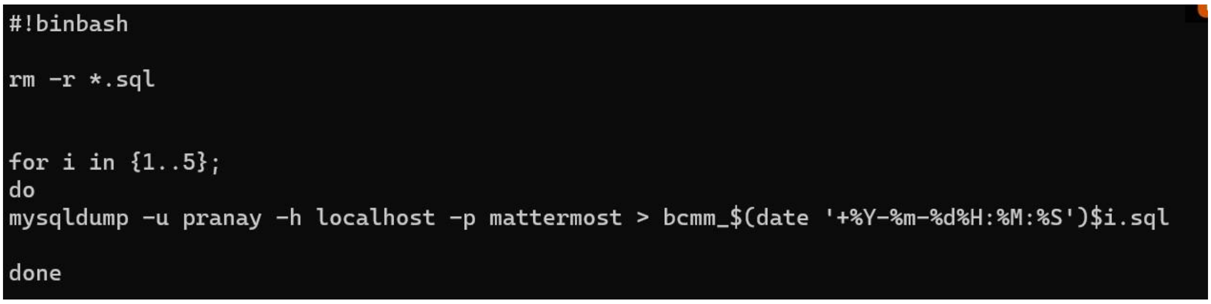
(Note: Before taking the dump you need to check for the oldest dumps and delete)

2. Dump should be in saved in timestamp

A:



```
[pranay@localhost ~]$ ls
apt_2.4.9_amd64.deb          bcmm_2023-06-1406:36:434.sql         rpmbuild     software
a.sh                         bcmm_2023-06-1406:36:505.sql         script       sqldumpp.sh
automatedel.sh               bcmmmm_2023-06-1406:39:32.sql        script1.sh   welcome-0.0.1
bcmm_2023-06-1406:36:191.sql  dir4                                script2.sh
bcmm_2023-06-1406:36:282.sql  ksh-20120801-144.el7_9.x86_64.rpm   scripttb.sh
bcmm_2023-06-1406:36:373.sql  ModemManager-1.6.10-4.el7.x86_64.rpm  seqldump.sh
[pranay@localhost ~]$ bash ./sqldumpp.sh
Enter password:
Enter password:
Enter password:
Enter password:
Enter password:
[pranay@localhost ~]$ ls
apt_2.4.9_amd64.deb          bcmm_2023-06-1406:40:183.sql        ModemManager-1.6.10-4.el7.x86_64.rpm  scripttb.sh
a.sh                         bcmm_2023-06-1406:40:234.sql        rpmbuild                              seqldump.sh
automatedel.sh               bcmm_2023-06-1406:40:315.sql        script                                software
bcmm_2023-06-1406:40:031.sql  dir4                               script1.sh                            sqldumpp.sh
bcmm_2023-06-1406:40:132.sql  ksh-20120801-144.el7_9.x86_64.rpm  script2.sh                            welcome-0.0.1
[pranay@localhost ~]$
```



```
#!binbash

rm -r *.sql


for i in {1..5};
do
mysqldump -u pranay -h localhost -p mattermost > bcmm_$(date '+%Y-%m-%d%H:%M:%S')$i.sql

done
```