Practical Computing
(UCS311)
Evaluation Assignment 1

Submitted By-

Name: Ananya Agarwal

Roll No: 102083036

Batch: 2CO14

Q1. A) Create a file named **Rolllist** in your home directory and Insert your roll number and one another roll number (matching pattern of your roll number) into it (**Rolllist**file**).**

B) Write a shell script that:

a) accepts your roll no as command line argument.

b) matches the command line roll no with the **Rolllist** File

c) If script find roll no in **Rolllist** file then **Print** the reverse of therollnumber otherwise **display** the message **"Roll number is not found in the file"**

Write commands for every activity (1, 2 (a, b, c)) and attach screen shot.

Sol 1:

Creation of a file with name Rolllist, changing permissions on the file (giving execution permissions) and inserting my Roll number and matching Roll number into it.

```
ananya@ananya-VirtualBox:~$ cat > Rolllist
102083036
102083034
ananya@ananya-VirtualBox:~$ nano shl.sh
ananya@ananya-VirtualBox:~$ chmod 777 shl.sh
```

My Roll No.

Creating a shell script using nano editor with name shl.sh and performing operations onto it.

```
ananya@ananya-VirtualBox: ~
 GNU nano 4.8                                    shl.sh
roll=$1
flag=0
while read Rno
do
if [[ $Rno -eq $roll ]]
then
echo $roll | rev
flag=1
break
fi
done < Rolllist
if [ $flag -eq 0 ]
then
echo "Roll number is not found in the file"
fi
```

For example;

When the correct Roll number is passed as command line argument, then ATQ the reverse of Roll number is printed.

When the Roll number in the Rolllist file is not same as passed as command line argument, 'Roll number is not found in the file' gets printed.

```
ananya@ananya-VirtualBox:~$ ./shl.sh 102083036
630380201
ananya@ananya-VirtualBox:~$ ./shl.sh 102083033
Roll number is not found in the file
ananya@ananya-VirtualBox:~$
```

Q2. Write a program to create the weekly schedule of your Practical computing class with menu driven option in shell script and save it as **"PCSchedule".**

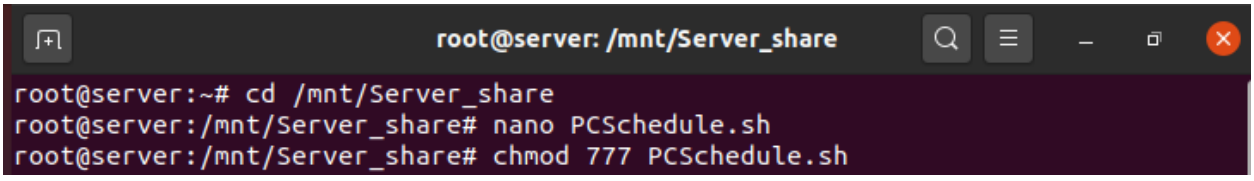
Hint: Menu –1. Monday, 2. Tuesday……3. Saturday

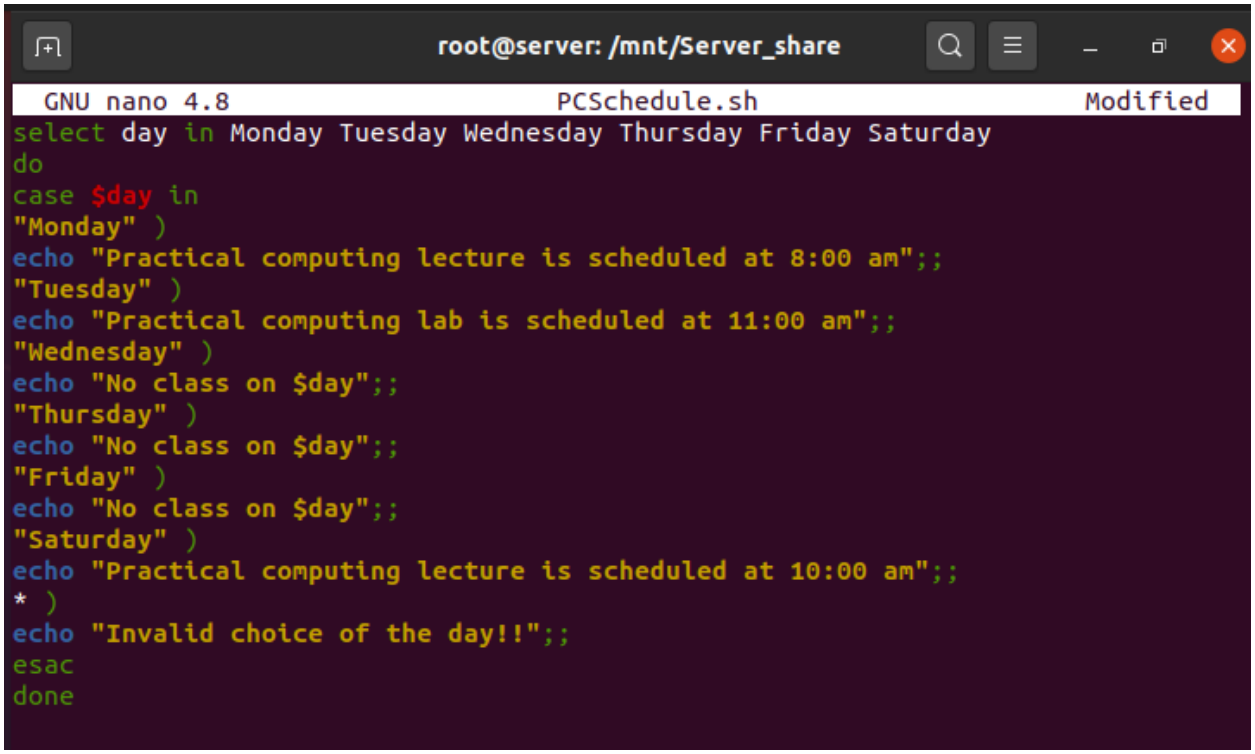
If option 1 (Monday) is selected,
Display: Practical computing is scheduled at 8:00 am.

Save this script in a folder named "Server_share" on the server. Further, access this file (script) from client folder named "Client_share" and execute it on client's terminal.

Sol 2:

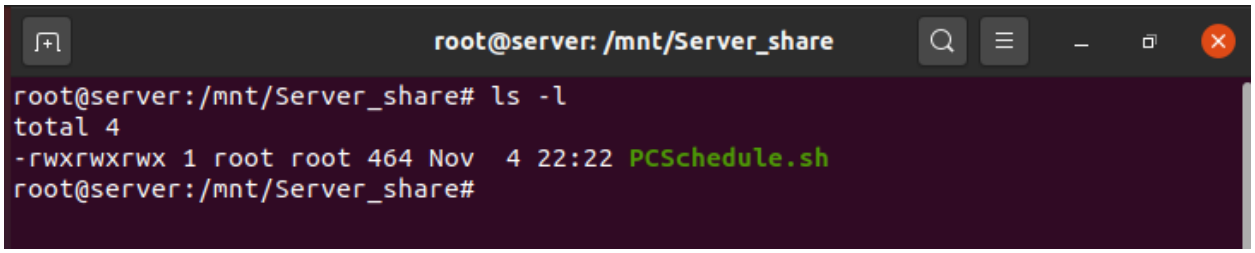
A shell script with name PCSchedule (using nano editor) is created in the folder Server_share on the server side along with changing the permissions. Then, a menu of the 'days' is shown to user so that the user can choose a particular day. Then as per the schedule we had set, the output for that particular day appeared on the terminal.



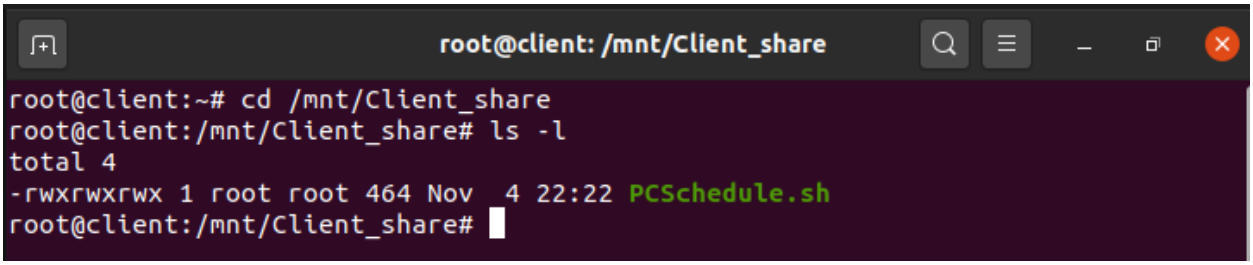



```
root@server:/mnt/Server_share# ./PCSchedule.sh
1) Monday      3) Wednesday   5) Friday
2) Tuesday     4) Thursday    6) Saturday
#? 1
Practical computing lecture is scheduled at 8:00 am
#? ^Croot@server:/mnt/Server_share# ./PCSchedule.sh
1) Monday      3) Wednesday   5) Friday
2) Tuesday     4) Thursday    6) Saturday
#? 4
No class on Thursday
#? ^Croot@server:/mnt/Server_share#
```

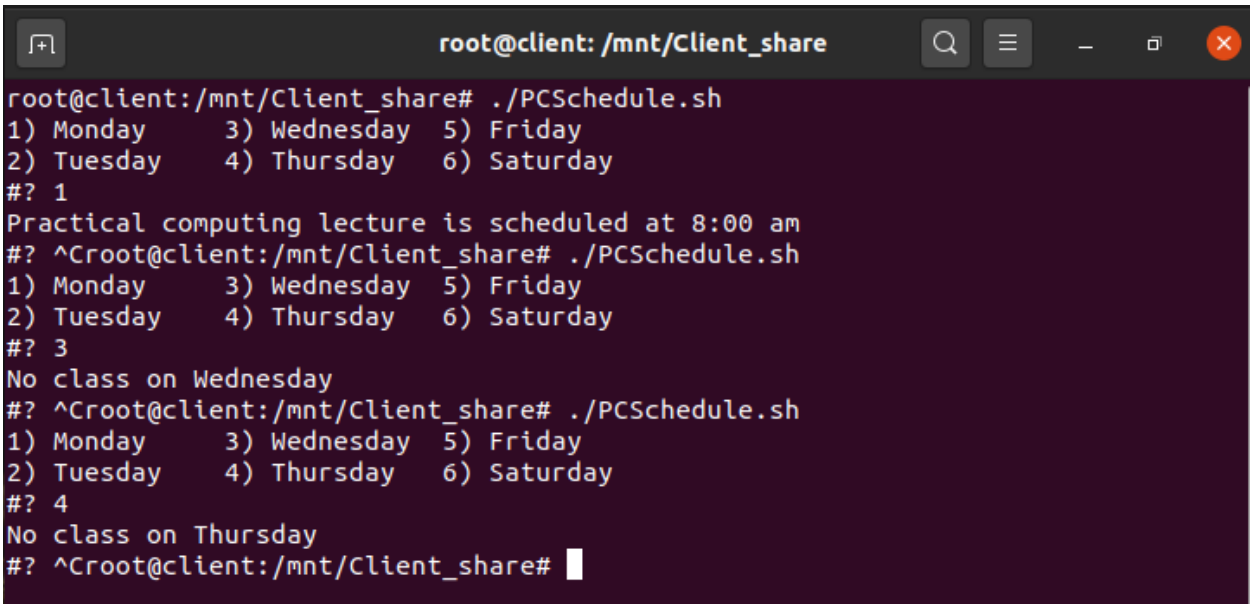
PCSchedule accessible on server side.



The shell script PCSchedule listed on client side as well.



Client can execute the script on the server side as well.



Q3. Take a name as an input from the user. Write a shell script to check whether there is a directory or file with that name. If any of this exists then print "This is a directory" or "This is a file" respectively, otherwise give three options to the user for creating a new directory or file with that input name or do nothing. Create a directory or file or do nothing as per the selected option by user.

Sol 3:

Creating a shell script check.sh using nano editor and changing its permissions.

It is clearly visible that 'ananya' file and 'agarwal' directory both are available in the home directory so these were displayed as file and directory respectively.

```
ananya@ananya-VirtualBox:~$ nano check.sh
ananya@ananya-VirtualBox:~$ chmod 777 check.sh
ananya@ananya-VirtualBox:~$ ls
agarwal  a.out     emp.lst   hello1.cpp  priority_non-preemptive.cpp  q6.sh     rr.cpp  sjf_non-premptive.cpp
ananya   check.sh  fcfs.cpp  matrix.sh   priority_preemptive.cpp      Rolllist  shl.sh  sjf_preemptive.cpp
```

ananya@ananya-VirtualBox: ~

```
 GNU nano 4.8                                                  check.sh
read -p "Enter a name to be searched as a file name or a directory name : " name
if [ -d "$name" ]
then
echo "This is a directory "
elif [ -f "$name" ]
then
echo "This is a file"
else
echo " $name file/directory does not exist"
echo "Options Available: (1)Create a directory $name  (2)Create a file $name  (3)Do nothing"
read -p "Enter your choice : " choice
case $choice in
"1")
mkdir $name
echo "$name directory created!!";;
"2")
touch $name
echo "$name file created!!";;
"3")
;;
*)
echo "Invalid choice!!"
esac
fi
```

ananya file existed thus, 'This is a file' came as an output while Ananya file/directory doesn't existed. Ananya file was created then as per the user's choice.

```
ananya@ananya-VirtualBox:~$ ./check.sh
Enter a name to be searched as a file name or a directory name : ananya
This is a file
ananya@ananya-VirtualBox:~$ ./check.sh
Enter a name to be searched as a file name or a directory name : Ananya
 Ananya file/directory does not exist
Options Available: (1)Create a directory Ananya  (2)Create a file Ananya  (3)Do nothing
Enter your choice : 2
Ananya file created!!
ananya@ananya-VirtualBox:~$ ls
agarwal  a.out     fcfs.cpp    priority_non-preemptive.cpp  Rolllist  sjf_non-premptive.cpp
ananya   check.sh  hello1.cpp  priority_preemptive.cpp      rr.cpp    sjf_preemptive.cpp
Ananya   emp.lst   matrix.sh   q6.sh                        shl.sh
```

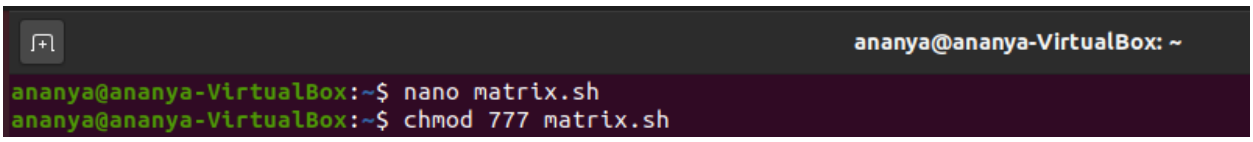
agarwal directory existed thus, 'This is a directory' came as an output while Agarwal file/directory doesn't existed. Agarwal directory was created then as per user's choice.

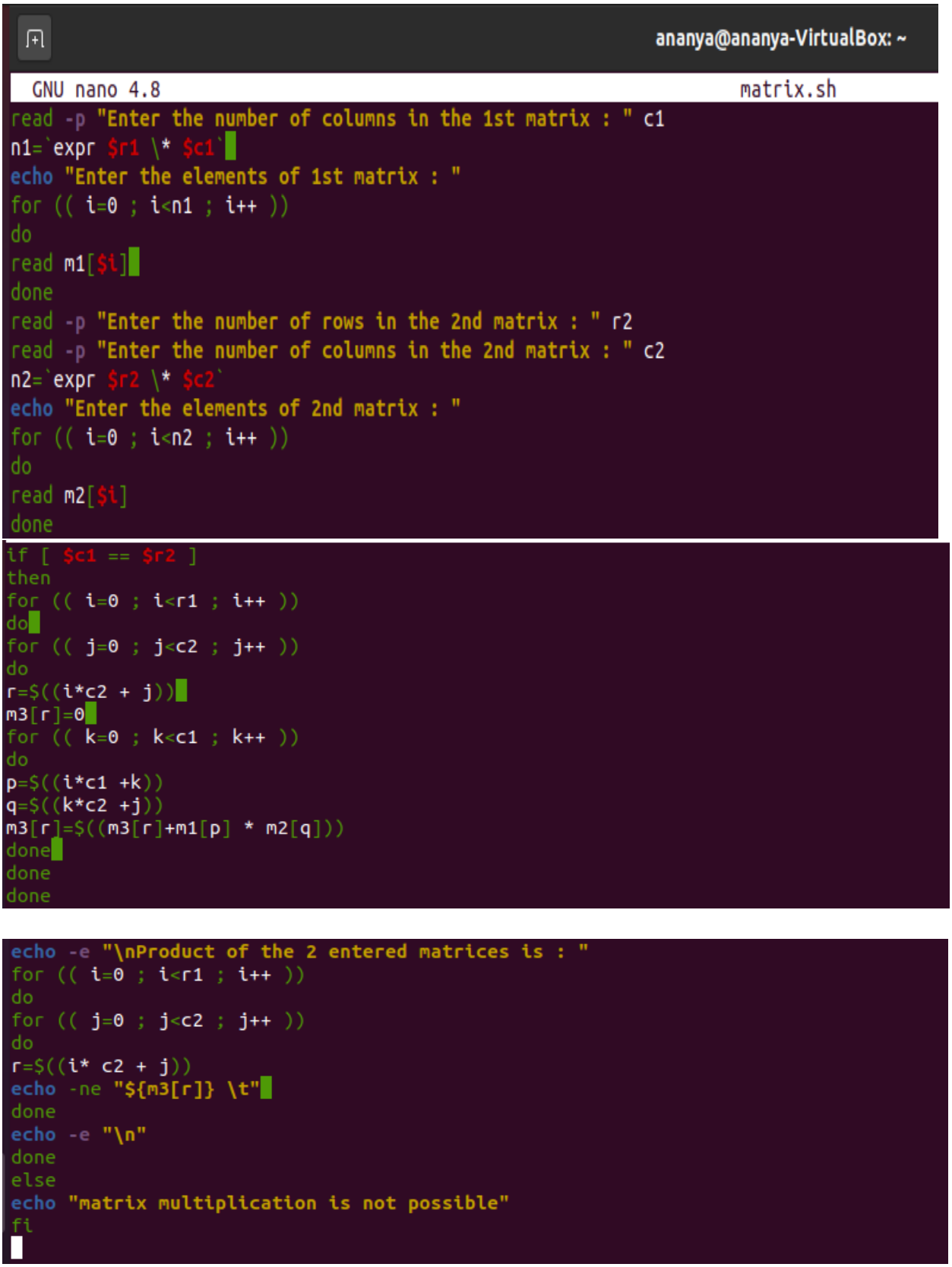
PC file/directory does not exist thus, 'PC file/directory does not exist' came as an output.

Q4. Write a shell script for multiplying 2 matrices and printing the resultant matrix. Here you need to represent a 2-dimensional matrix using a 1-dimensional array.

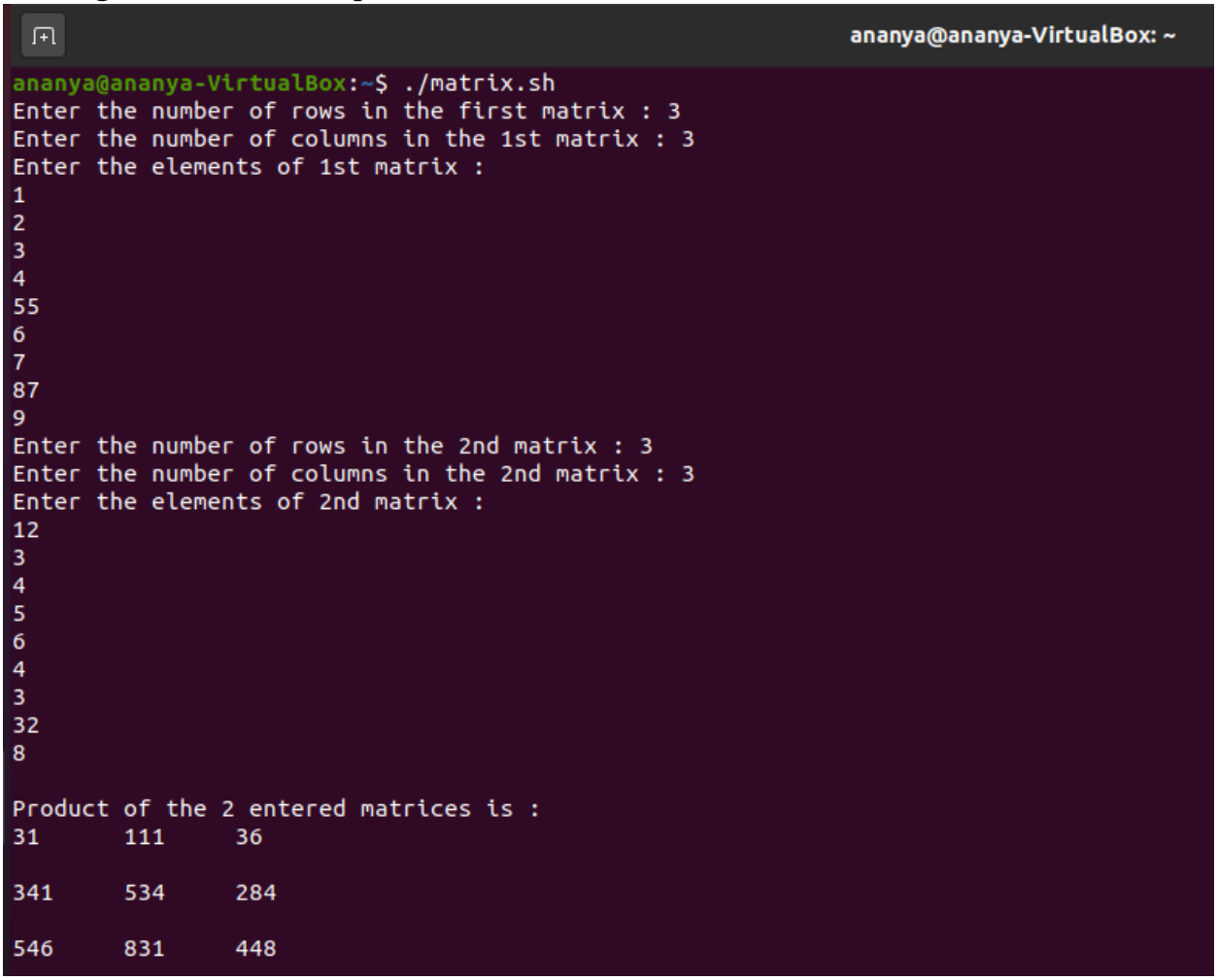
Sol 4:

Creating a shell script with name matrix.sh using nano editor along with changing its permissions and then writing the matrix multiplication code into it.




ananya@ananya-VirtualBox: ~
GNU nano 4.8
matrix.sh
read -p "Enter the number of columns in the 1st matrix : " c1
n1=`expr $r1 \* $c1`
echo "Enter the elements of 1st matrix : "
for ((i=0 ; i<n1 ; i++))
do
read m1[$i]
done
read -p "Enter the number of rows in the 2nd matrix : " r2
read -p "Enter the number of columns in the 2nd matrix : " c2
n2=`expr $r2 \* $c2`
echo "Enter the elements of 2nd matrix : "
for ((i=0 ; i<n2 ; i++))
do
read m2[$i]
done
if [$c1 == $r2]
then
for ((i=0 ; i<r1 ; i++))
do
for ((j=0 ; j<c2 ; j++))
do
r=$((i*c2 + j))
m3[r]=0
for ((k=0 ; k<c1 ; k++))
do
p=$((i*c1 +k))
q=$((k*c2 +j))
m3[r]=$((m3[r]+m1[p] * m2[q]))
done
done
done
echo -e "\nProduct of the 2 entered matrices is : "
for ((i=0 ; i<r1 ; i++))
do
for ((j=0 ; j<c2 ; j++))
do
r=$((i* c2 + j))
echo -ne "${m3[r]} \t"
done
echo -e "\n"
done
else
echo "matrix multiplication is not possible"
fi

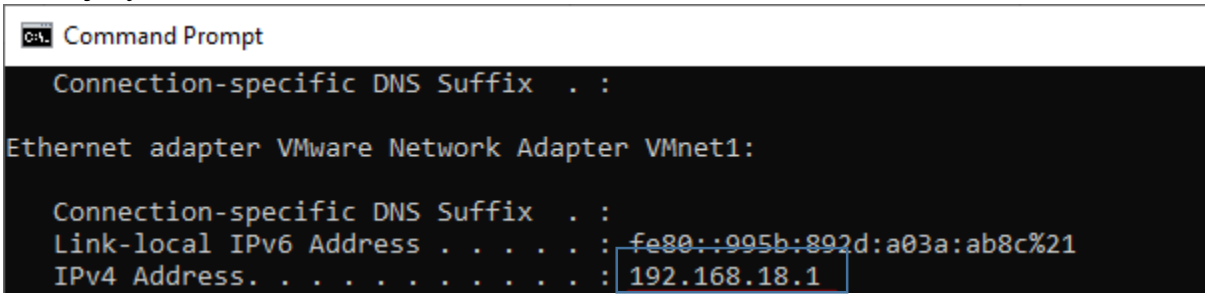
Getting the matrix multiplication:



Q5.Conversion:

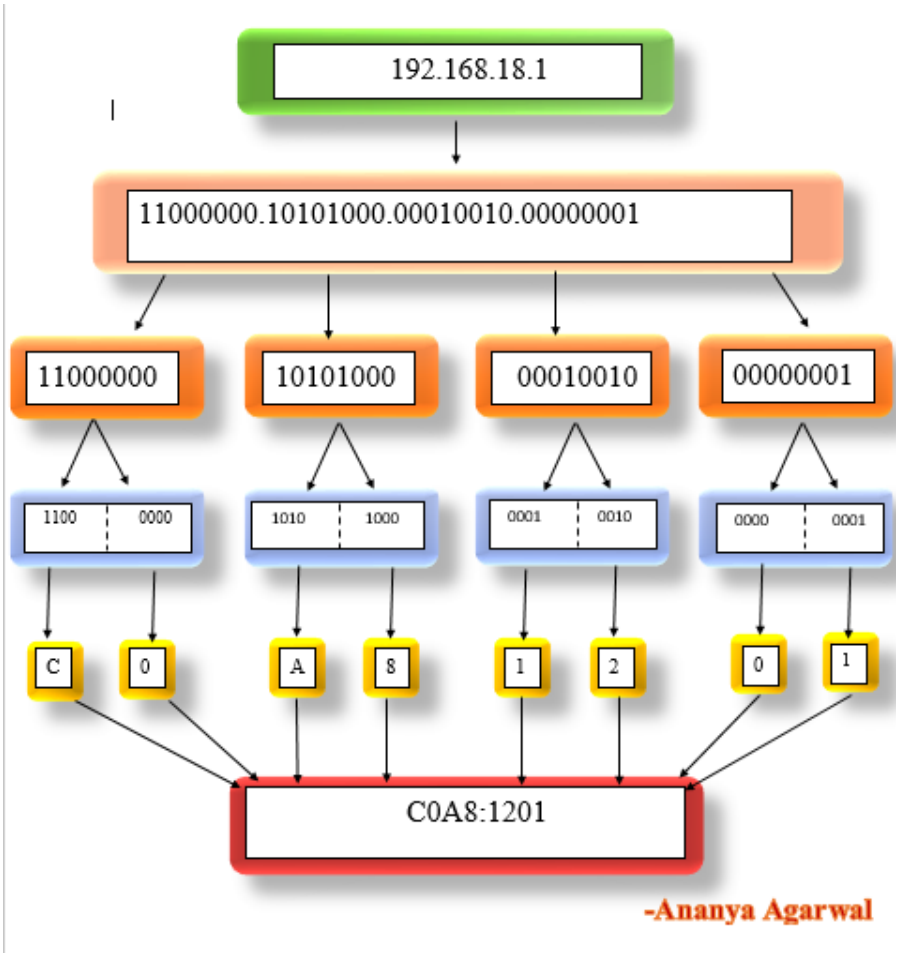
a) IPV4 to IPV6
b) IPV6 to IPV4

My system's IPV4 address



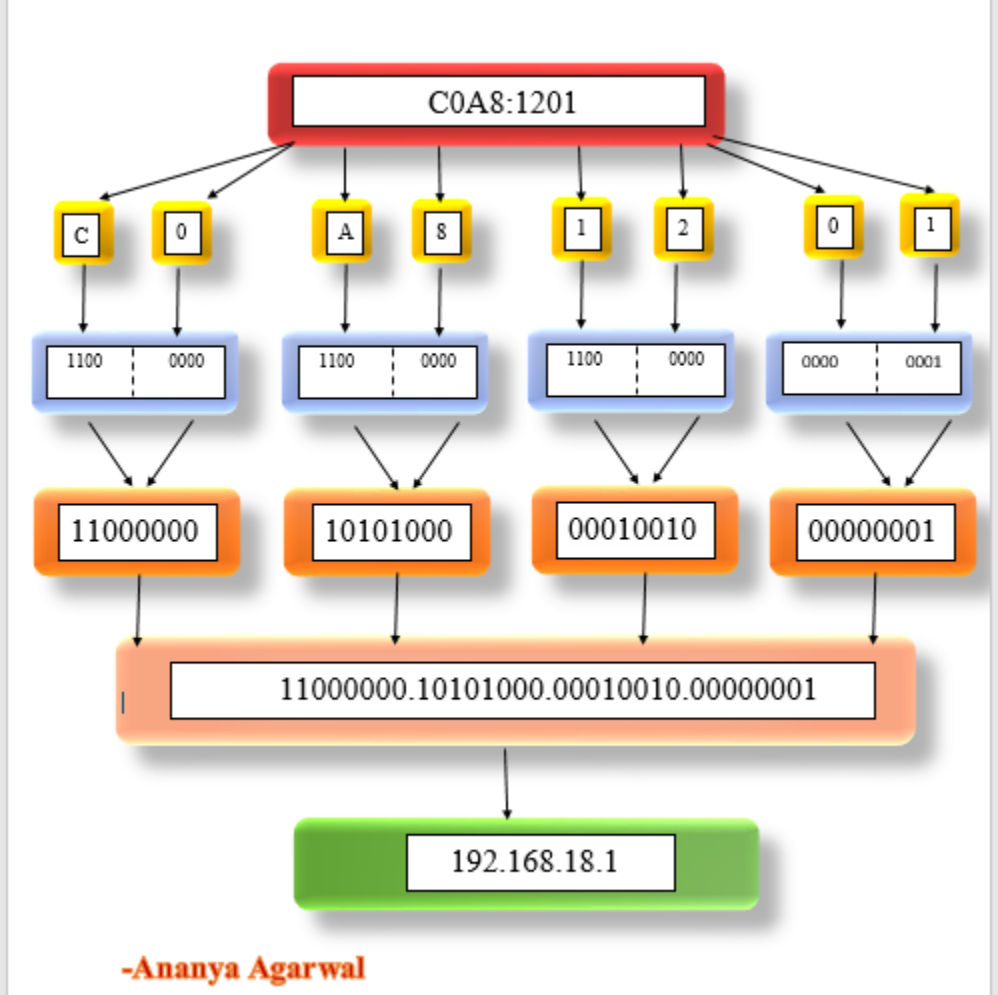
Sol 5 a) IPV4 to IPV6

- First the IPV4 address (decimal equivalent) is converted to its binary equivalent.
- Then, the binary representation is sub-divided into 4 groups of 8-bit each.
- Binary number is further divided into 4-bit binary representation each.
- Now this is converted to hexadecimal number representation. Thus, we are left with 8 hexadecimal numbers.
- Then, the 8 hexadecimal numbers are divided into 2 groups of 4 hexadecimal numbers each and combined into a single address (IPV6 address) separated by a colon.



Sol 5 b) IPV6 to IPV4

- First the IPV6 address (hexadecimal equivalent) is separated into 8 different hexadecimal numbers.
- Then, for every hexadecimal number, a 4-bit binary equivalent is written.
- These 4-bit binary number is further paired consecutively to make 4 such 8-bit binary representations.
- All these 8-bit binary representations are joined together separated by a ".".
- Write decimal representation for each of this 8-bit binary representation separated by a ".", we get the required IPV4 address (decimal equivalent).



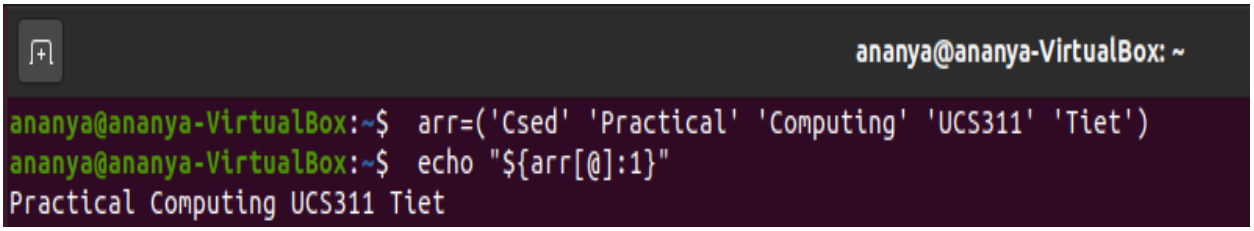
-Ananya Agarwal

Q6..#!/bin/bash
Array= (Csed, Practical, Computing, UCS311,Tiet,)
a.Write a command to display all elements except first one
b. what will be the output of
i)echo ${arr[0]:1}
ii) echo ${array[@]:1:3}
iii)echo ${array[1]:5:5}

Sol 6:

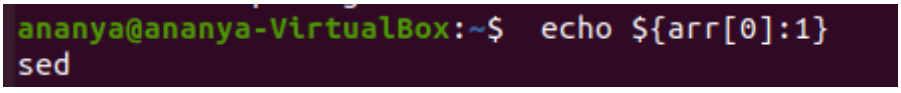
1) The contents present at all the indices are printed using @ and using :1(here) all the elements from the array index 1 are printed i.e. Csed is excluded. This can be done by using the command:

   echo "${arr[@]:1}"



b) 1) arr[0] means the element at the 0th index which is Csed.
:1 (here) signifies that the elements from position 1 of the 0th index element are printed thus, C is not printed in Csed.
Thus, the required output is:

sed



b) 2) arr[@] means that the contents present at all the indices of the array gets printed, but since here :1:3 was mentioned, 3 elements starting from the element at the 1st index of array gets printed.
Thus, the required output is:

Practical Computing UCS311

```
ananya@ananya-VirtualBox:~$ echo ${arr[@]:1:3}
Practical Computing UCS311
```

b) 3) arr[1] means the element at the 1st index of the array which is Practical. We have to display the content beginning from the 5th position (considering 0 as the start) and printing the next 5 characters. However, here the element Practical is over (length of Practical after t is left 4 only) before displaying the next 5 characters.

Thus, the required output is:

ical

```
ananya@ananya-VirtualBox:~$ echo ${arr[1]:5:5}
ical
```

END