Practical Computing (UCS311)

Assignment No. - 2 & 3

Submitted By-

Name: Ananya Agarwal

Roll No: 102083036

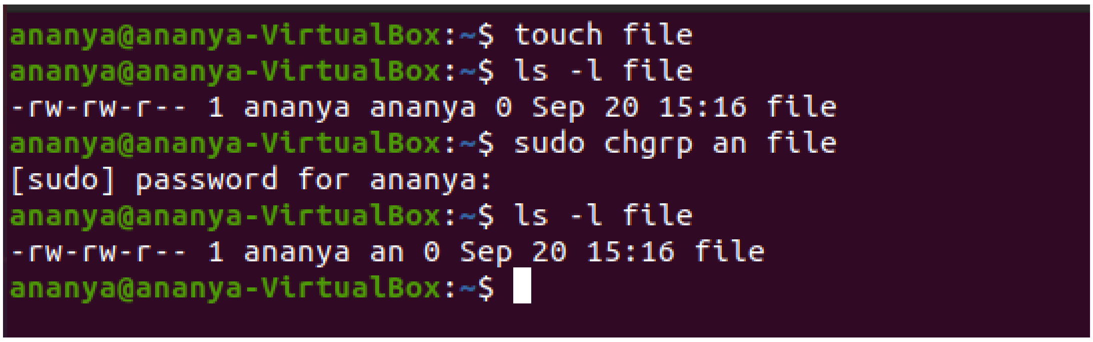
Batch: 2CO14

Assignment-2

Sol 1.

The **chgrp** command (along with sudo command for root login) is used to change the group of a file or directory.

Syntax of command: `$ sudo chgrp groupname filename`

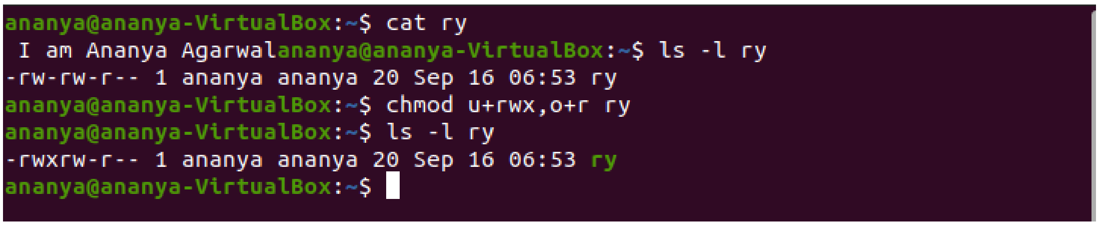


```
ananya@ananya-VirtualBox:~$ touch file
ananya@ananya-VirtualBox:~$ ls -l file
-rw-rw-r-- 1 ananya ananya 0 Sep 20 15:16 file
ananya@ananya-VirtualBox:~$ sudo chgrp an file
[sudo] password for ananya:
ananya@ananya-VirtualBox:~$ ls -l file
-rw-rw-r-- 1 ananya an 0 Sep 20 15:16 file
ananya@ananya-VirtualBox:~$
```

Sol 2.

To provide the **user all** the rights & **others only permissions to read** the file using the command :

```
$ chmod u=rwx,o=r filename
```



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rw-rw-r-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod u+rwx,o+r ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rwxrw-r-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$
```

Sol 3.

Yes

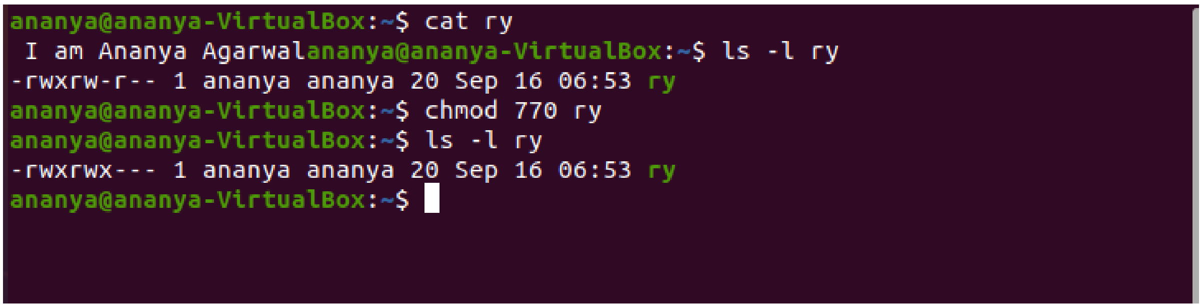
In the octal number system, **4 denotes permissions to read**, **2 denotes permissions to write**, **1 denotes permissions to execute**, **0 denotes no permissions** and similarly their combination denotes other octal numbers.

For **user** the octal number is **7** =4+2+1, which means **rwx**.

For **group** the octal number is **7** =4+2+1, which means **rwx**.

For **others** the octal number is **0**, which means **---**

Thus validating the statement given in the question.



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rwxrw-r-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod 770 ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwx--- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ 
```

Sol 4.

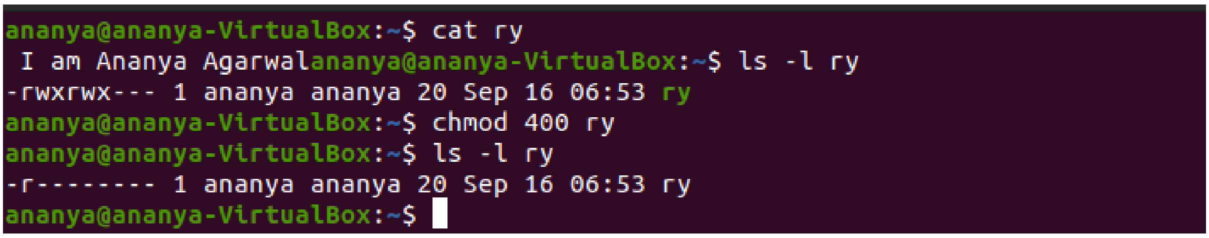
Yes

In the octal number system, **4 denotes permissions to read**, **2 denotes permissions to write**, **1 denotes permissions to execute**, **0 denotes no permissions** and similarly their combination denotes other octal numbers.

For **user** the octal number is **4**, which means **r--**

For **group** the octal number is **0**, which means **---**

For **others** the octal number is **0**, which means **---**



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwx--- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod 400 ry
ananya@ananya-VirtualBox:~$ ls -l ry
-r-------- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ 
```

Sol 5.

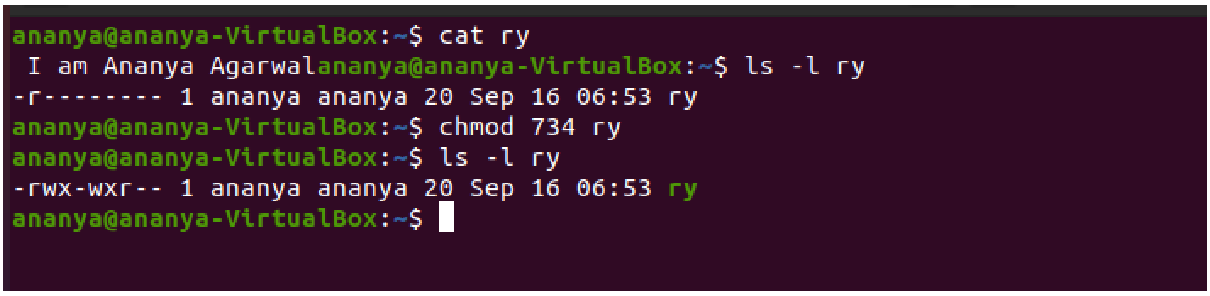
No

In the octal number system, **4 denotes permissions to read**, **2 denotes permissions to write**, **1 denotes permissions to execute**, **0 denotes no permissions** and similarly their combination denotes other octal numbers.

For **user** the octal number is **7** (4+2+1), which means **rwx**.

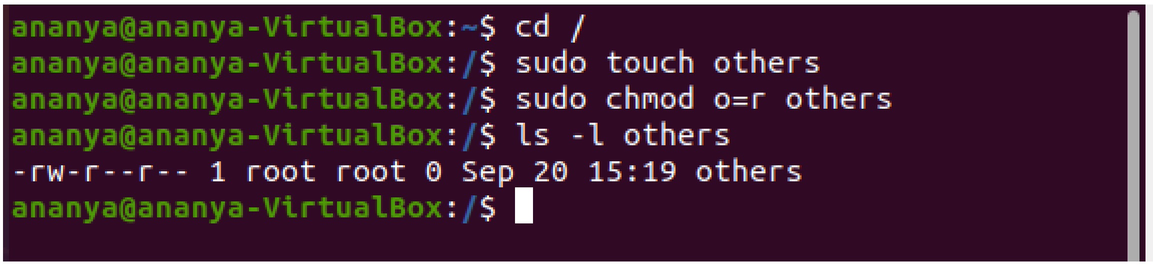
For **group,** the octal number is **3** =2+1, which means **-wx**.

For **others,** the octal number is **4**, which means **r--**.



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-r-------- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod 734 ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rwx-wxr-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ 
```

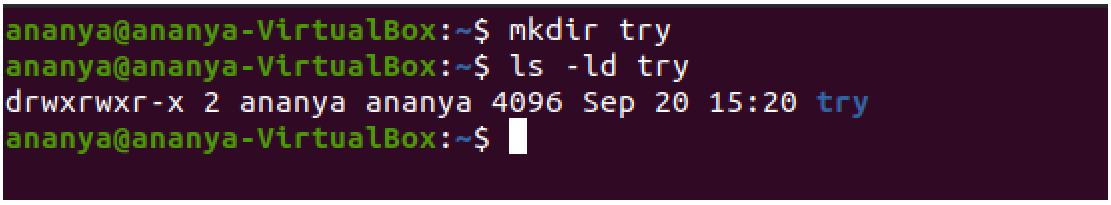
Sol 6. Below is the file to create a file at root and giving read to others.



```
ananya@ananya-VirtualBox:~$ cd /
ananya@ananya-VirtualBox:/$ sudo touch others
ananya@ananya-VirtualBox:/$ sudo chmod o=r others
ananya@ananya-VirtualBox:/$ ls -l others
-rw-r--r-- 1 root root 0 Sep 20 15:19 others
ananya@ananya-VirtualBox:/$ 
```

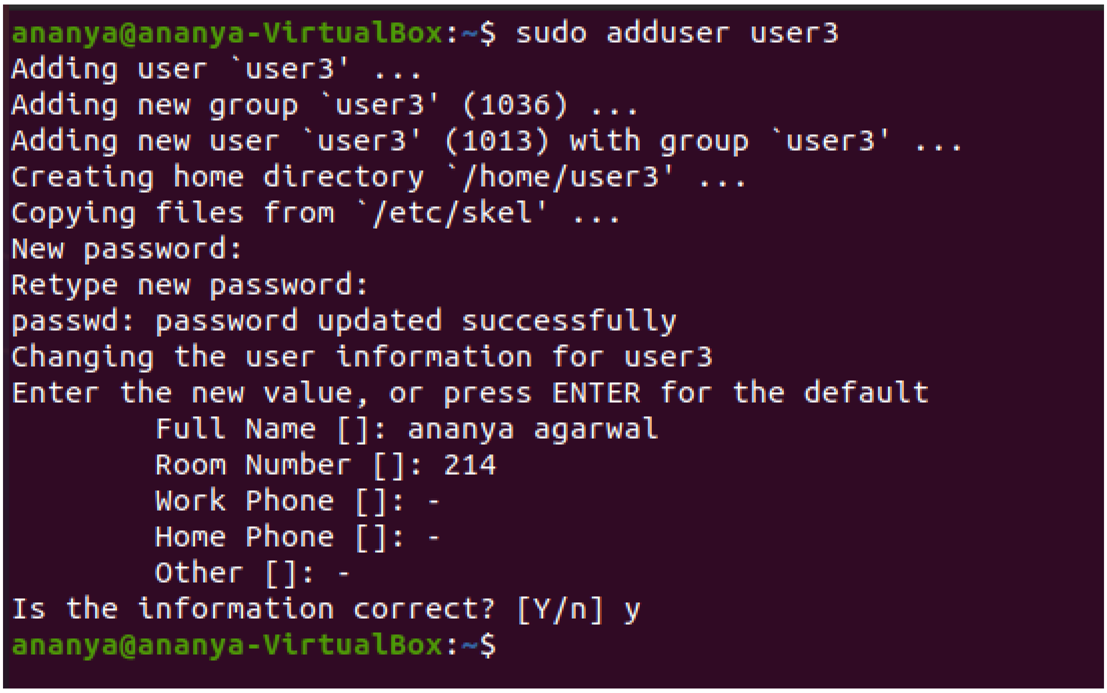
Sol 7.

1) **Make a directory which will by default be created with a default group.**

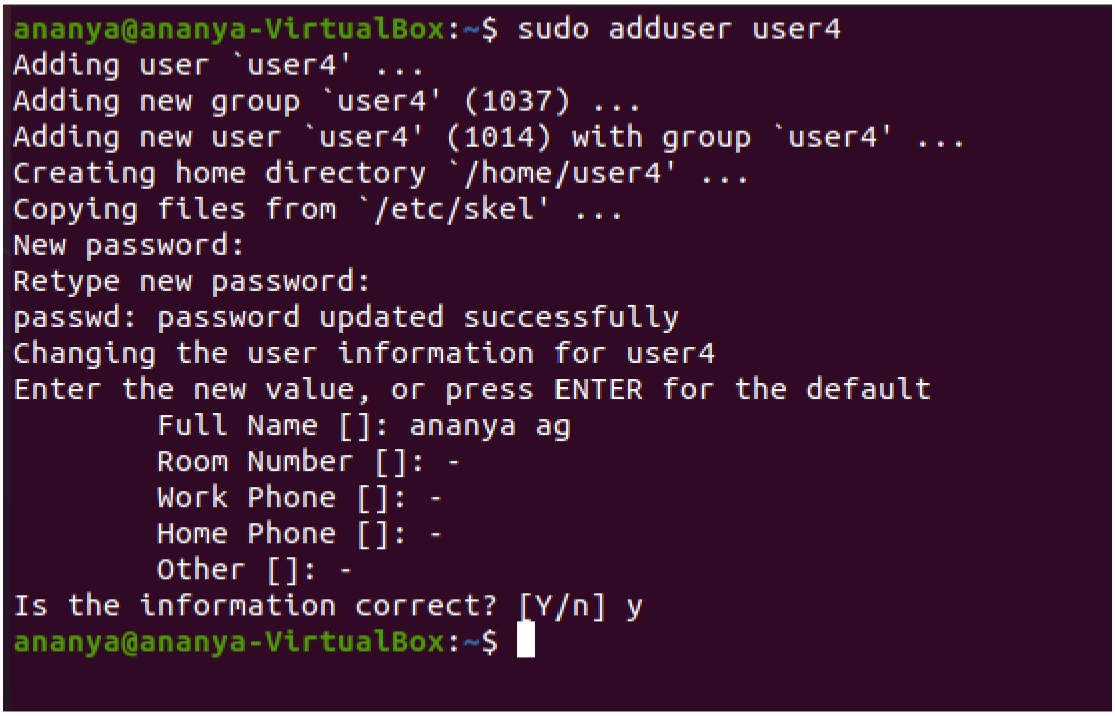


```
ananya@ananya-VirtualBox:~$ mkdir try
ananya@ananya-VirtualBox:~$ ls -ld try
drwxrwxr-x 2 ananya ananya 4096 Sep 20 15:20 try
ananya@ananya-VirtualBox:~$ 
```

2) **Now add 2 users – let it be user3 & user4.**

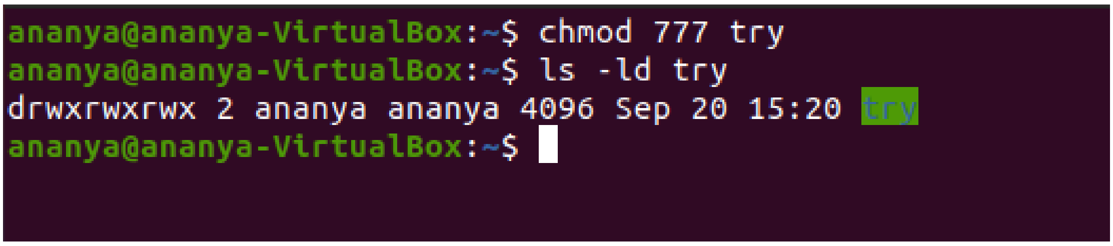


```
ananya@ananya-VirtualBox:~$ sudo adduser user3
Adding user `user3' ...
Adding new group `user3' (1036) ...
Adding new user `user3' (1013) with group `user3' ...
Creating home directory `/home/user3' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user3
Enter the new value, or press ENTER for the default
        Full Name []: ananya agarwal
        Room Number []: 214
        Work Phone []: -
        Home Phone []: -
        Other []: -
Is the information correct? [Y/n] y
ananya@ananya-VirtualBox:~$
```

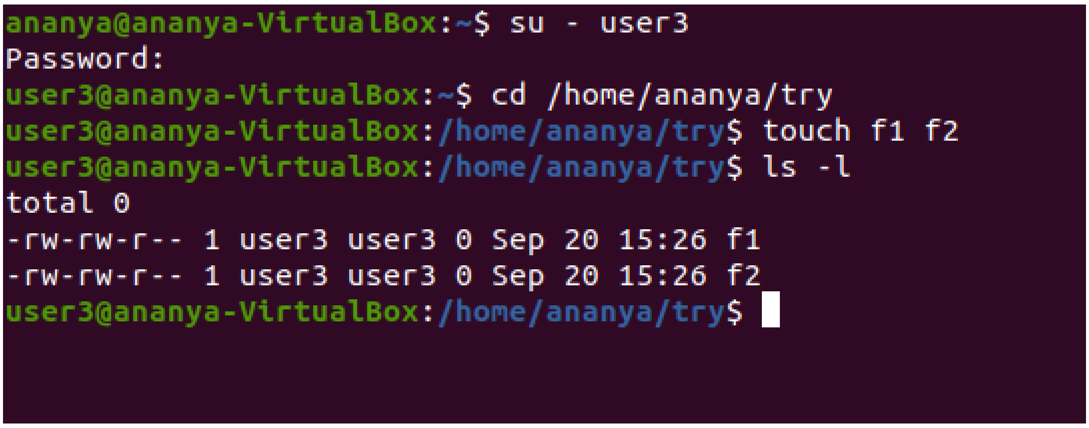


```
ananya@ananya-VirtualBox:~$ sudo adduser user4
Adding user `user4' ...
Adding new group `user4' (1037) ...
Adding new user `user4' (1014) with group `user4' ...
Creating home directory `/home/user4' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user4
Enter the new value, or press ENTER for the default
        Full Name []: ananya ag
        Room Number []: -
        Work Phone []: -
        Home Phone []: -
        Other []: -
Is the information correct? [Y/n] y
ananya@ananya-VirtualBox:~$
```

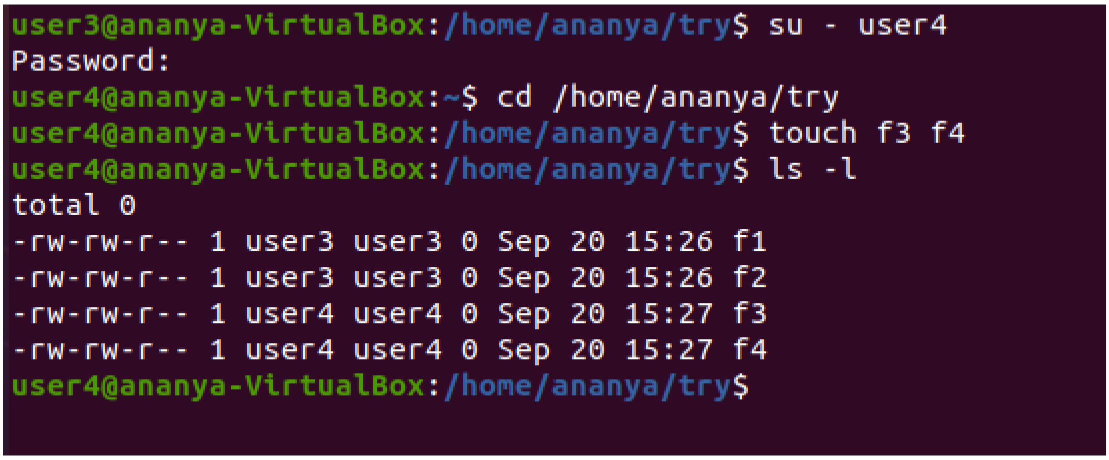
3) **Login as individual users one by one & create files for each user inside a directory.**



```
ananya@ananya-VirtualBox:~$ chmod 777 try
ananya@ananya-VirtualBox:~$ ls -ld try
drwxrwxrwx 2 ananya ananya 4096 Sep 20 15:20 try
ananya@ananya-VirtualBox:~$
```

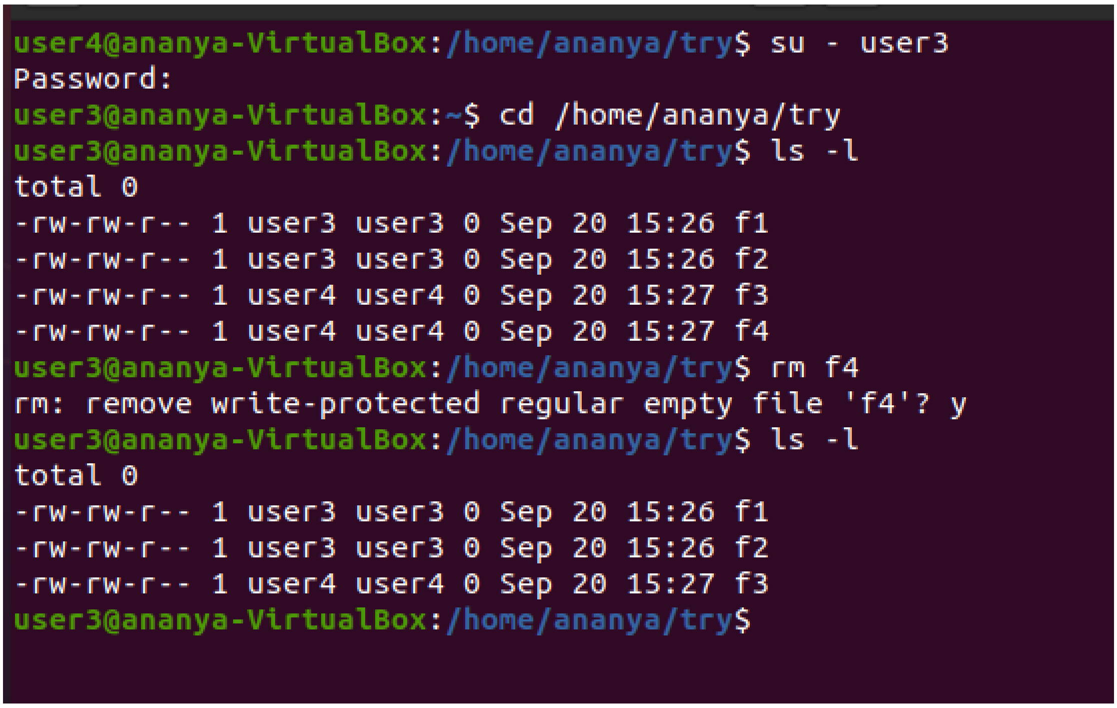


```
ananya@ananya-VirtualBox:~$ su - user3
Password:
user3@ananya-VirtualBox:~$ cd /home/ananya/try
user3@ananya-VirtualBox:/home/ananya/try$ touch f1 f2
user3@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f2
user3@ananya-VirtualBox:/home/ananya/try$
```

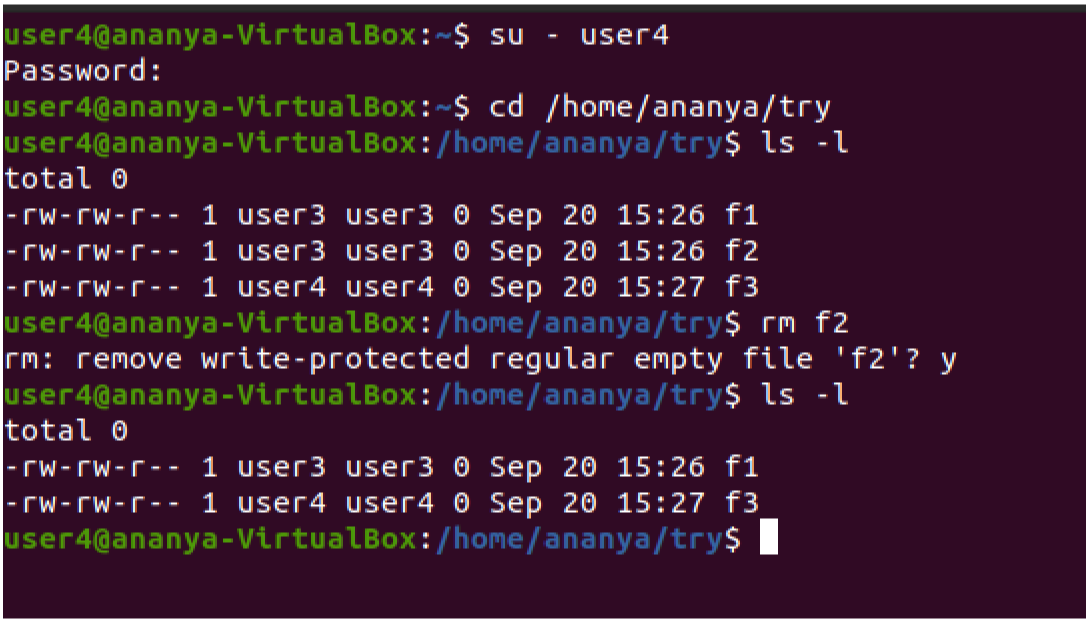


```
user3@ananya-VirtualBox:/home/ananya/try$ su - user4
Password:
user4@ananya-VirtualBox:~$ cd /home/ananya/try
user4@ananya-VirtualBox:/home/ananya/try$ touch f3 f4
user4@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f2
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f4
user4@ananya-VirtualBox:/home/ananya/try$
```

4) **Login with user3 and try to remove file of user2, & vice versa. As expected, files will be deleted.**

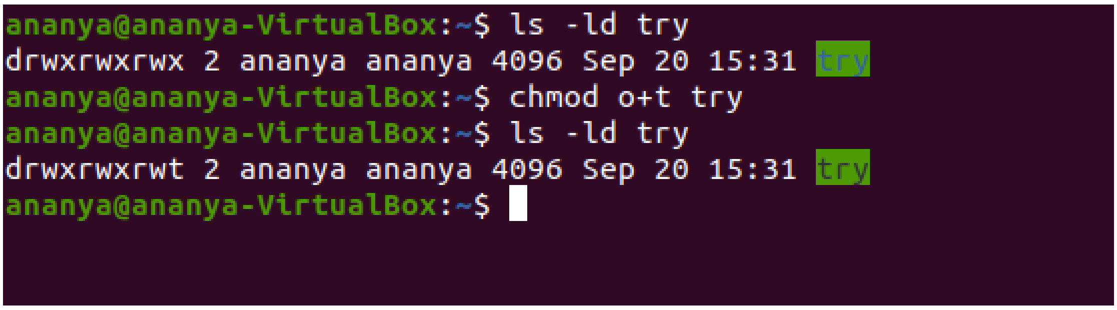


```
user4@ananya-VirtualBox:/home/ananya/try$ su - user3
Password:
user3@ananya-VirtualBox:~$ cd /home/ananya/try
user3@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f2
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f4
user3@ananya-VirtualBox:/home/ananya/try$ rm f4
rm: remove write-protected regular empty file 'f4'? y
user3@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f2
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user3@ananya-VirtualBox:/home/ananya/try$
```

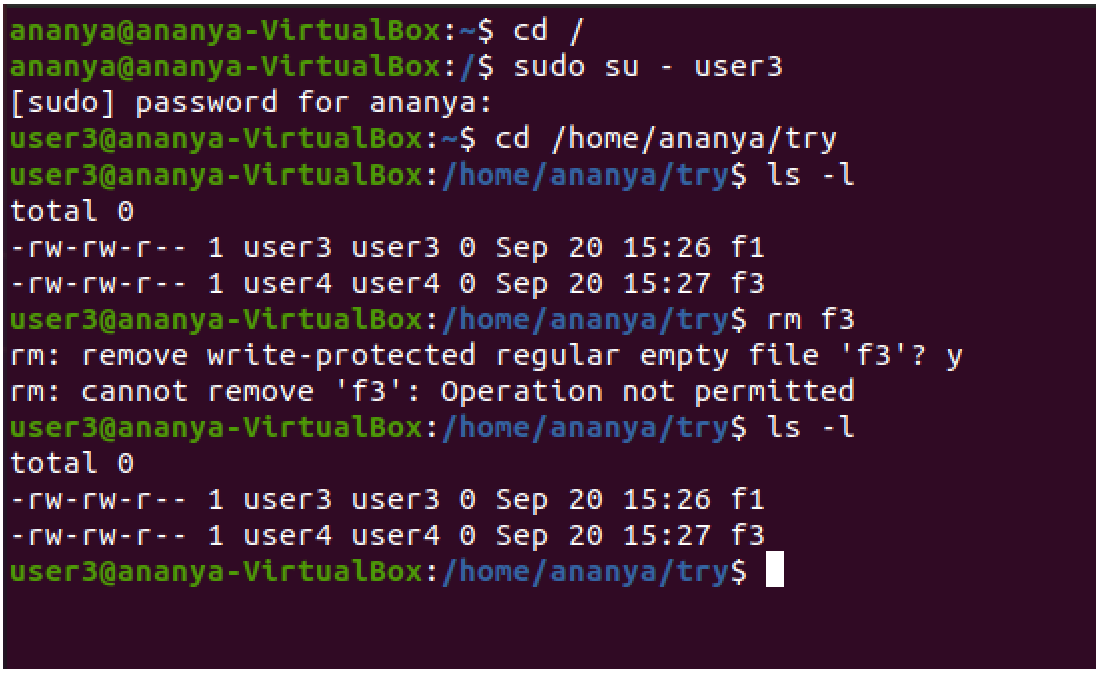


```
user4@ananya-VirtualBox:~$ su - user4
Password:
user4@ananya-VirtualBox:~$ cd /home/ananya/try
user4@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f2
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user4@ananya-VirtualBox:/home/ananya/try$ rm f2
rm: remove write-protected regular empty file 'f2'? y
user4@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user4@ananya-VirtualBox:/home/ananya/try$
```

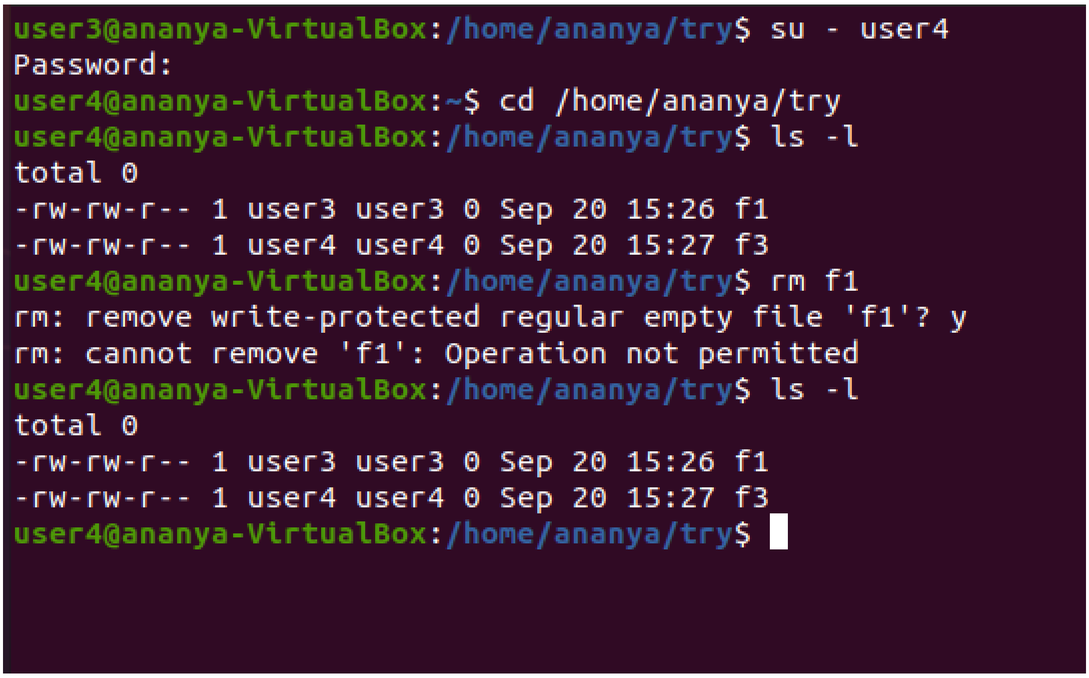
5) **To prevent deletion of one user's file from another user, apply sticky bit by using the command `$ chmod o+t test`. Now login with any user & try to remove file of the other user. Message prompted will be: Operation not permitted.**



```
ananya@ananya-VirtualBox:~$ ls -ld try
drwxrwxrwx 2 ananya ananya 4096 Sep 20 15:31 try
ananya@ananya-VirtualBox:~$ chmod o+t try
ananya@ananya-VirtualBox:~$ ls -ld try
drwxrwxrwt 2 ananya ananya 4096 Sep 20 15:31 try
ananya@ananya-VirtualBox:~$
```



```
ananya@ananya-VirtualBox:~$ cd /
ananya@ananya-VirtualBox:/$ sudo su - user3
[sudo] password for ananya:
user3@ananya-VirtualBox:~$ cd /home/ananya/try
user3@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user3@ananya-VirtualBox:/home/ananya/try$ rm f3
rm: remove write-protected regular empty file 'f3'? y
rm: cannot remove 'f3': Operation not permitted
user3@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user3@ananya-VirtualBox:/home/ananya/try$
```

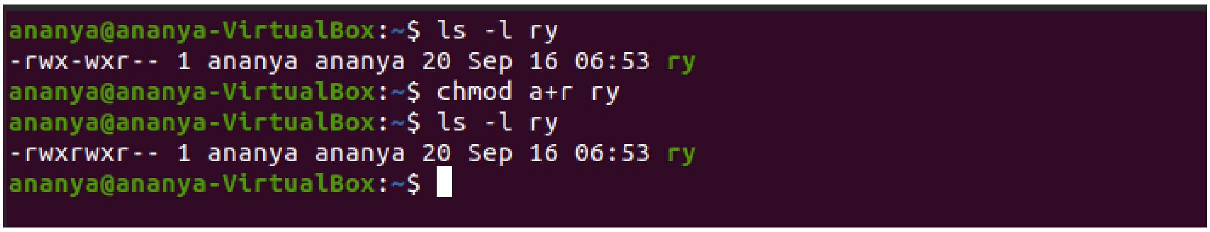


```
user3@ananya-VirtualBox:/home/ananya/try$ su - user4
Password:
user4@ananya-VirtualBox:~$ cd /home/ananya/try
user4@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user4@ananya-VirtualBox:/home/ananya/try$ rm f1
rm: remove write-protected regular empty file 'f1'? y
rm: cannot remove 'f1': Operation not permitted
user4@ananya-VirtualBox:/home/ananya/try$ ls -l
total 0
-rw-rw-r-- 1 user3 user3 0 Sep 20 15:26 f1
-rw-rw-r-- 1 user4 user4 0 Sep 20 15:27 f3
user4@ananya-VirtualBox:/home/ananya/try$
```

Sol 8.

Command used-

`$ chmod a+r filename`



```
ananya@ananya-VirtualBox:~$ ls -l ry
-rwx-wxr-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod a+r ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwxr-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$
```

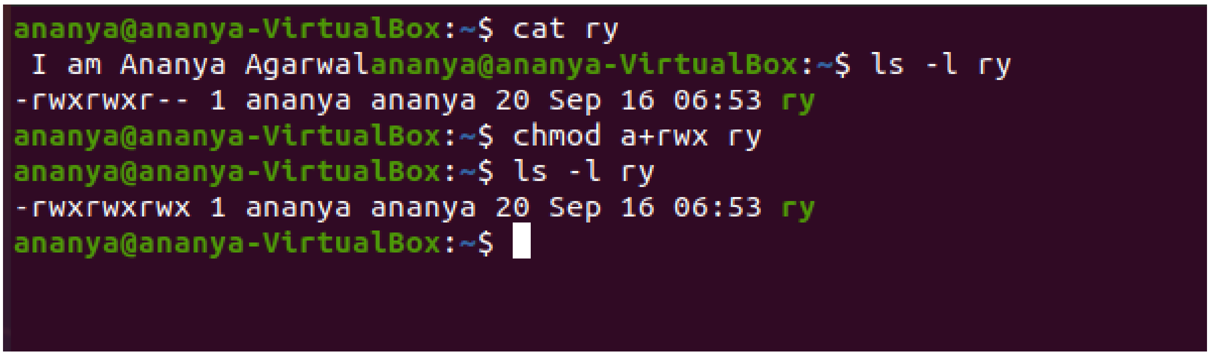
Sol 9.

Command used-

`$ chmod 777 filename`

or

`$ chmod a+rwx filename`



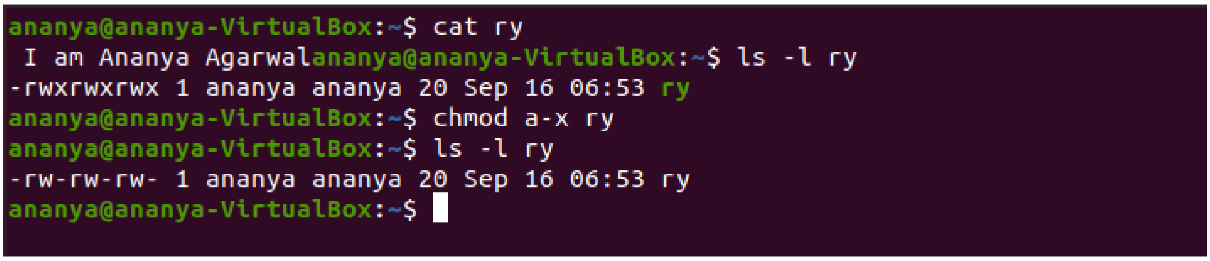
```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwxr-- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod a+rwx ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwxrwx 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ 
```

Sol 10.

The execute permission of any file/directory can be denied from everyone by using the command-

```
$ chmod a-x filename or dirname
```

By default, everyone can’t execute files.



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rwxrwxrwx 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod a-x ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rw-rw-rw- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ 
```

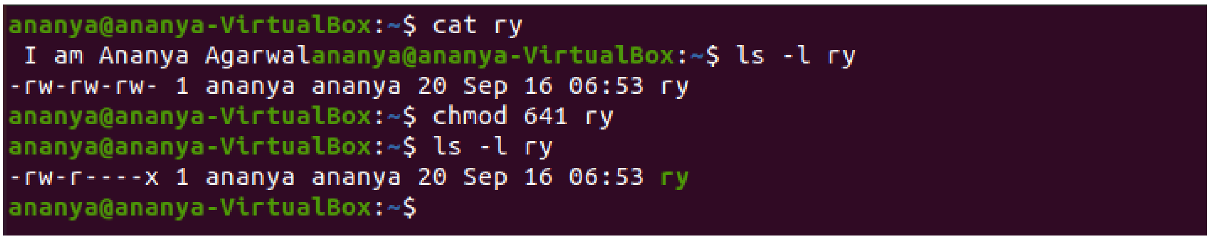
Sol 11.

In the octal number system, **4 denotes permissions to read**, **2 denotes permissions to write**, **1 denotes permissions to execute**, **0 denotes no permissions** and similarly their combination denotes other octal numbers.

For **user** the octal number is **6** =4+2, which means **rw-**.

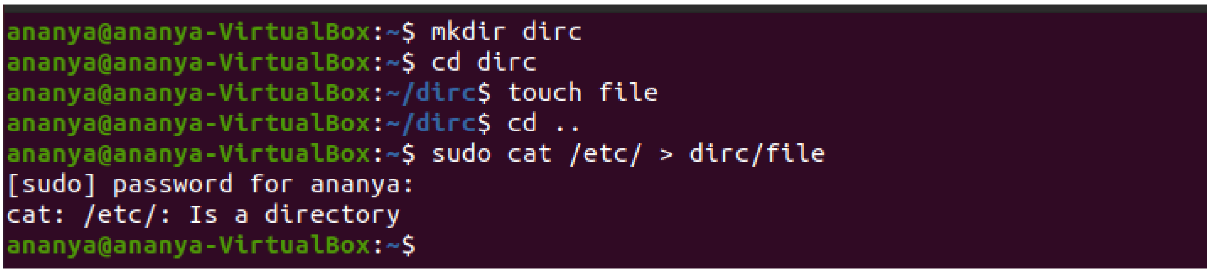
For **group** the octal number is **4**, which means **r--**.

For **others** the octal number is **1**, which means **--x**.



```
ananya@ananya-VirtualBox:~$ cat ry
 I am Ananya Agarwalananya@ananya-VirtualBox:~$ ls -l ry
-rw-rw-rw- 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$ chmod 641 ry
ananya@ananya-VirtualBox:~$ ls -l ry
-rw-r----x 1 ananya ananya 20 Sep 16 06:53 ry
ananya@ananya-VirtualBox:~$
```

Sol 12.



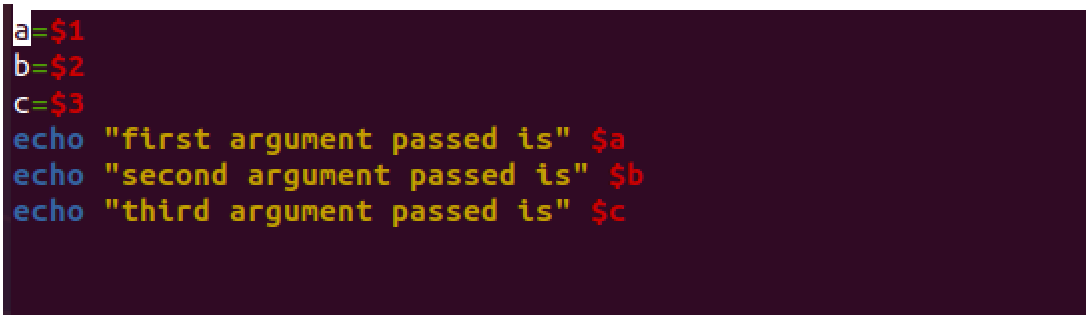
```
ananya@ananya-VirtualBox:~$ mkdir dirc
ananya@ananya-VirtualBox:~$ cd dirc
ananya@ananya-VirtualBox:~/dirc$ touch file
ananya@ananya-VirtualBox:~/dirc$ cd ..
ananya@ananya-VirtualBox:~$ sudo cat /etc/ > dirc/file
[sudo] password for ananya:
cat: /etc/: Is a directory
ananya@ananya-VirtualBox:~$
```

After running the commands, the user is the owner of the copied file.

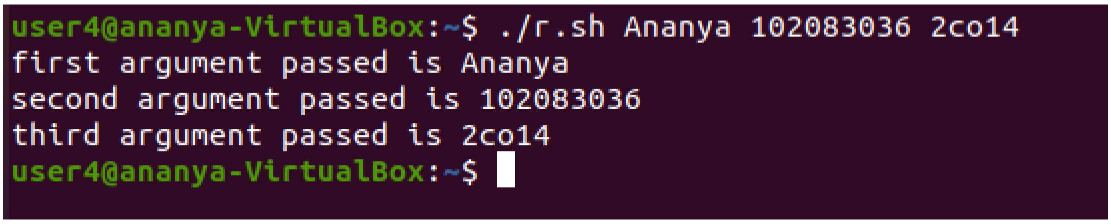
Assignment-3

Sol 1.

To pass command line arguments, write them on the terminal after script name separated with space. Inside the script, the $1 variable is the reference for the first argument in the command line, $2 for the second argument and $3 for the third argument.



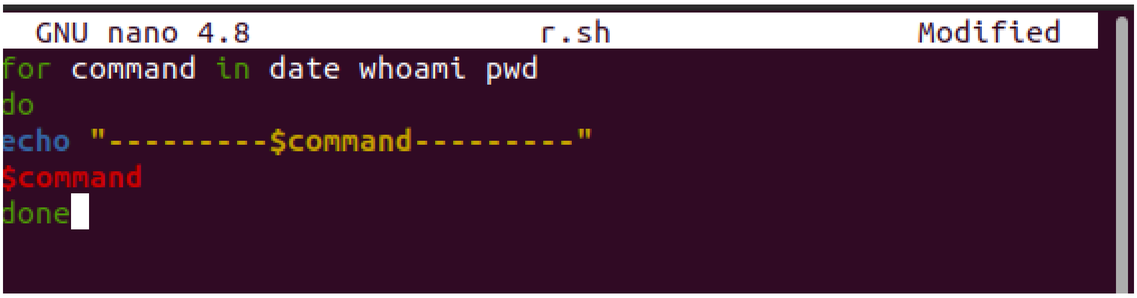
```
a=$1
b=$2
c=$3
echo "first argument passed is" $a
echo "second argument passed is" $b
echo "third argument passed is" $c
```



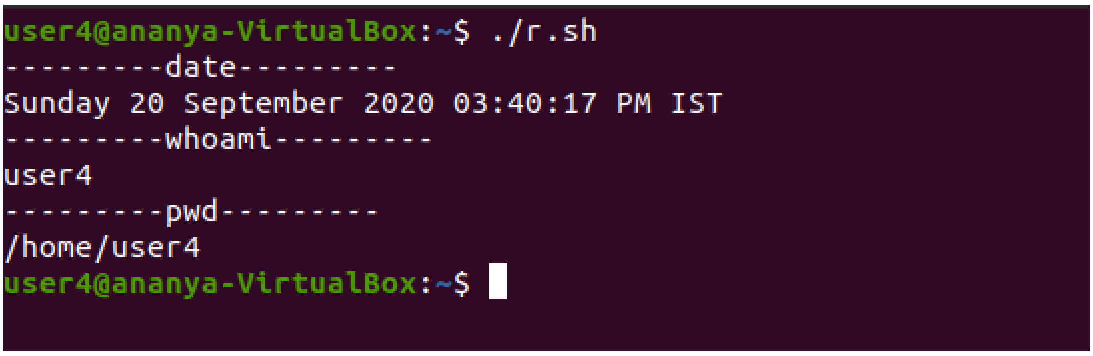
```
user4@ananya-VirtualBox:~$ ./r.sh Ananya 102083036 2co14
first argument passed is Ananya
second argument passed is 102083036
third argument passed is 2co14
user4@ananya-VirtualBox:~$ 
```

Sol 2.

'date' command gives us the current date and time as output, '**whoami**' command prints the username of the current user & '**pwd**' command prints the present working directory.



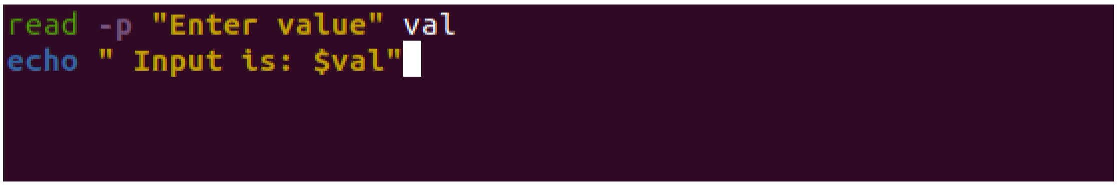
```
  GNU nano 4.8                r.sh                     Modified
for command in date whoami pwd
do
echo "---------$command---------"
$command
done
```



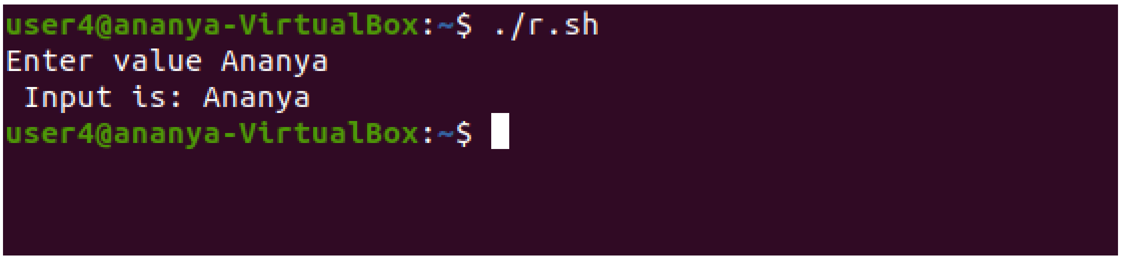
```
user4@ananya-VirtualBox:~$ ./r.sh
---------date---------
Sunday 20 September 2020 03:40:17 PM IST
---------whoami---------
user4
---------pwd---------
/home/user4
user4@ananya-VirtualBox:~$ 
```

Sol 3.

To read the Bash shell user’s input, use the built-in Bash command - **`read`**. It takes input from the user and assigns it to the variable. It reads only a single line from the Bash shell.



```
read -p "Enter value" val
echo " Input is: $val"
```



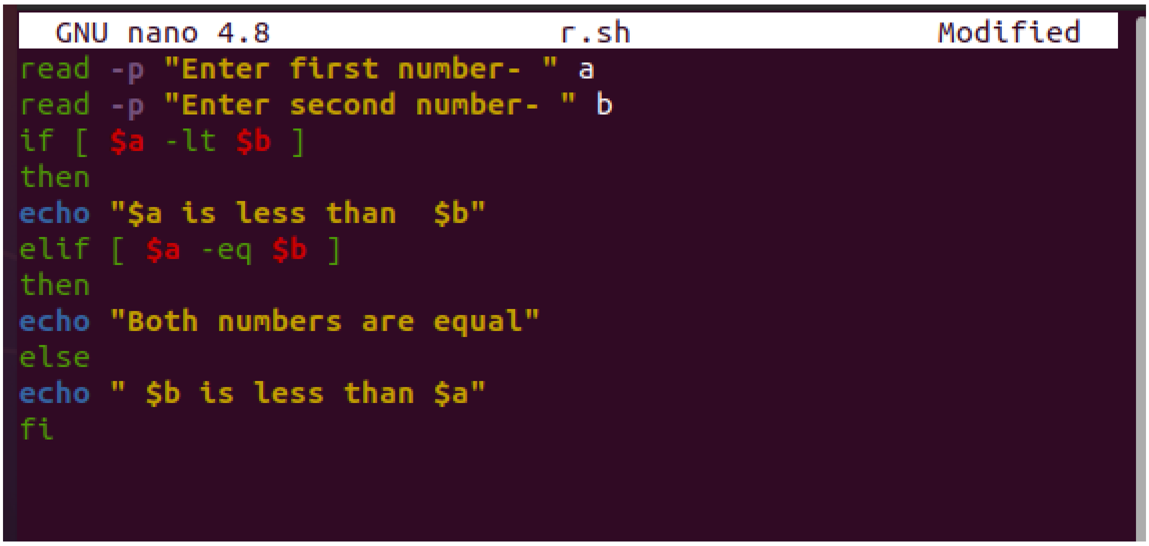
```
user4@ananya-VirtualBox:~$ ./r.sh
Enter value Ananya
 Input is: Ananya
user4@ananya-VirtualBox:~$
```

Sol 4.

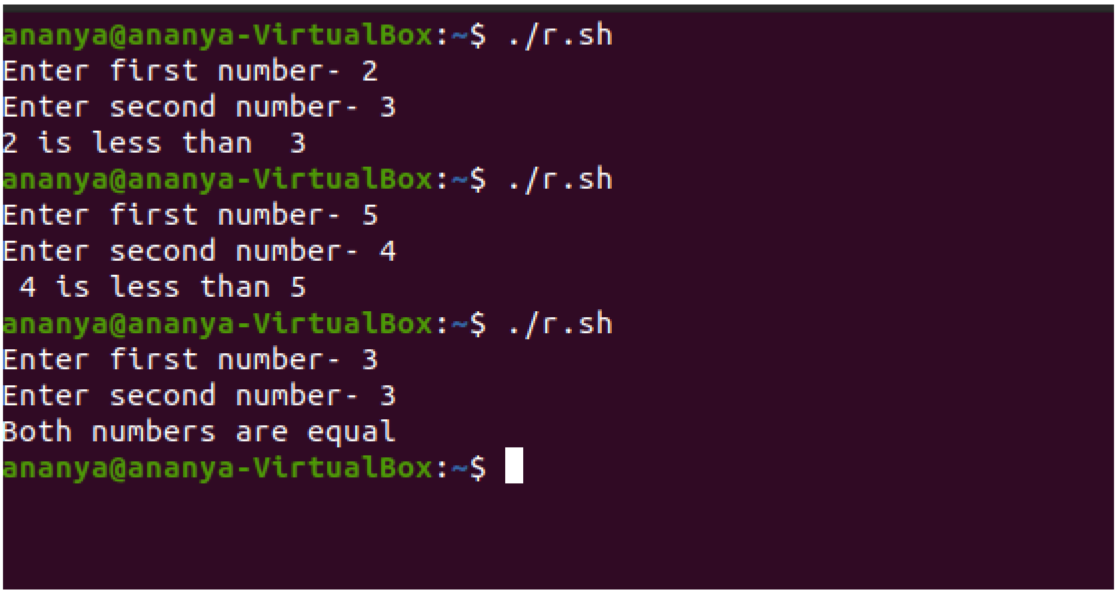
Linux provides a number of relational operators. These can be used to compare numeric values.

- **`-lt`** - less than
- **`-le`** - less than or equal to
- **`-gt`** - greater than
- **`-ge`** - greater than or equal to
- **`-eq`** - equal to
- **`-ne`** - not equal to

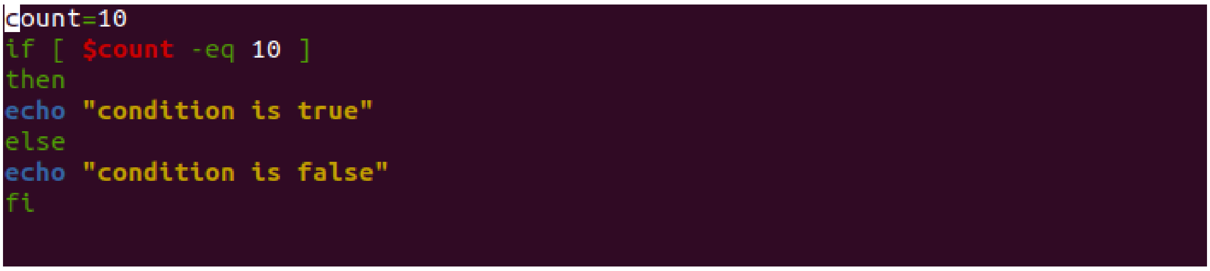
We can also use **`angular operators ( >, <, >=, <= )`** instead of using the above used keywords.



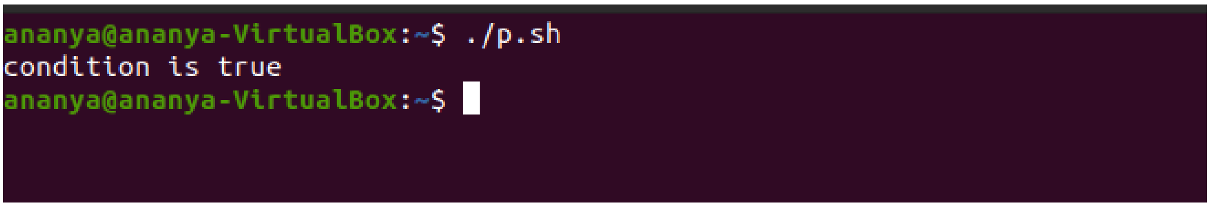
```
  GNU nano 4.8                    r.sh                      Modified
read -p "Enter first number- " a
read -p "Enter second number- " b
if [ $a -lt $b ]
then
echo "$a is less than  $b"
elif [ $a -eq $b ]
then
echo "Both numbers are equal"
else
echo " $b is less than $a"
fi
```



```
ananya@ananya-VirtualBox:~$ ./r.sh
Enter first number- 2
Enter second number- 3
2 is less than  3
ananya@ananya-VirtualBox:~$ ./r.sh
Enter first number- 5
Enter second number- 4
 4 is less than 5
ananya@ananya-VirtualBox:~$ ./r.sh
Enter first number- 3
Enter second number- 3
Both numbers are equal
ananya@ananya-VirtualBox:~$
```



```
count=10
if [ $count -eq 10 ]
then
echo "condition is true"
else
echo "condition is false"
fi
```



```
ananya@ananya-VirtualBox:~$ ./p.sh
condition is true
ananya@ananya-VirtualBox:~$
```

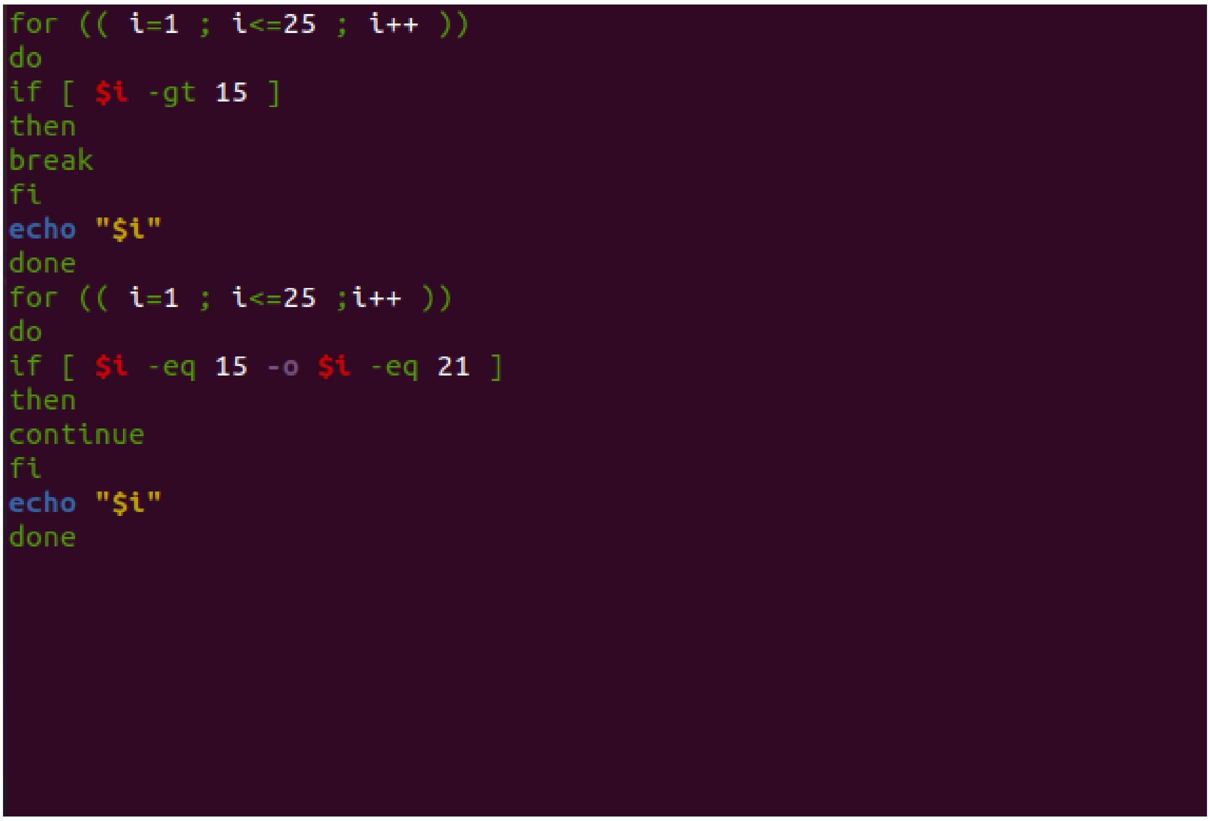
Sol 5.

1) ‘For’ loop

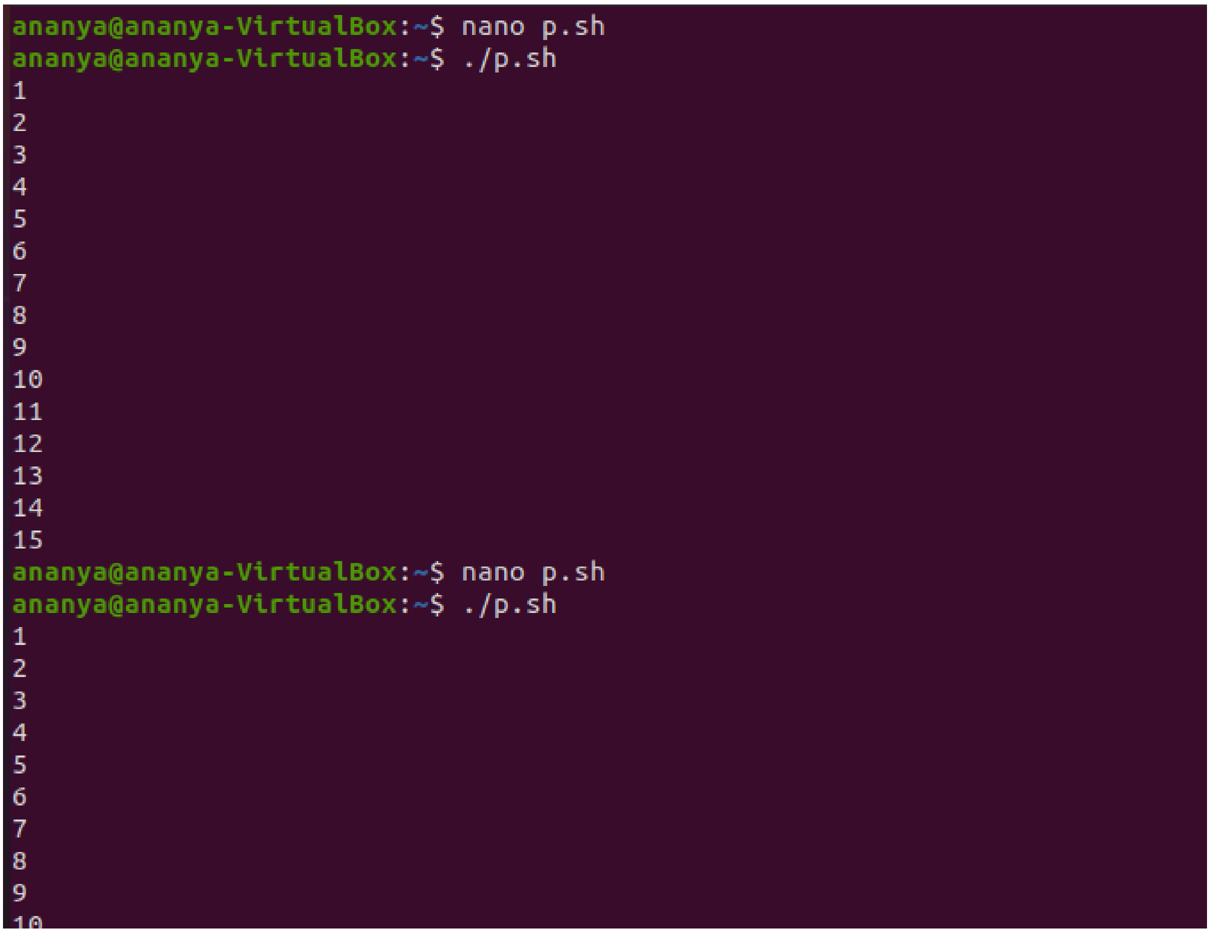
The for loop operates on lists of items. It repeats a set of commands for every item in a list.

Syntax:

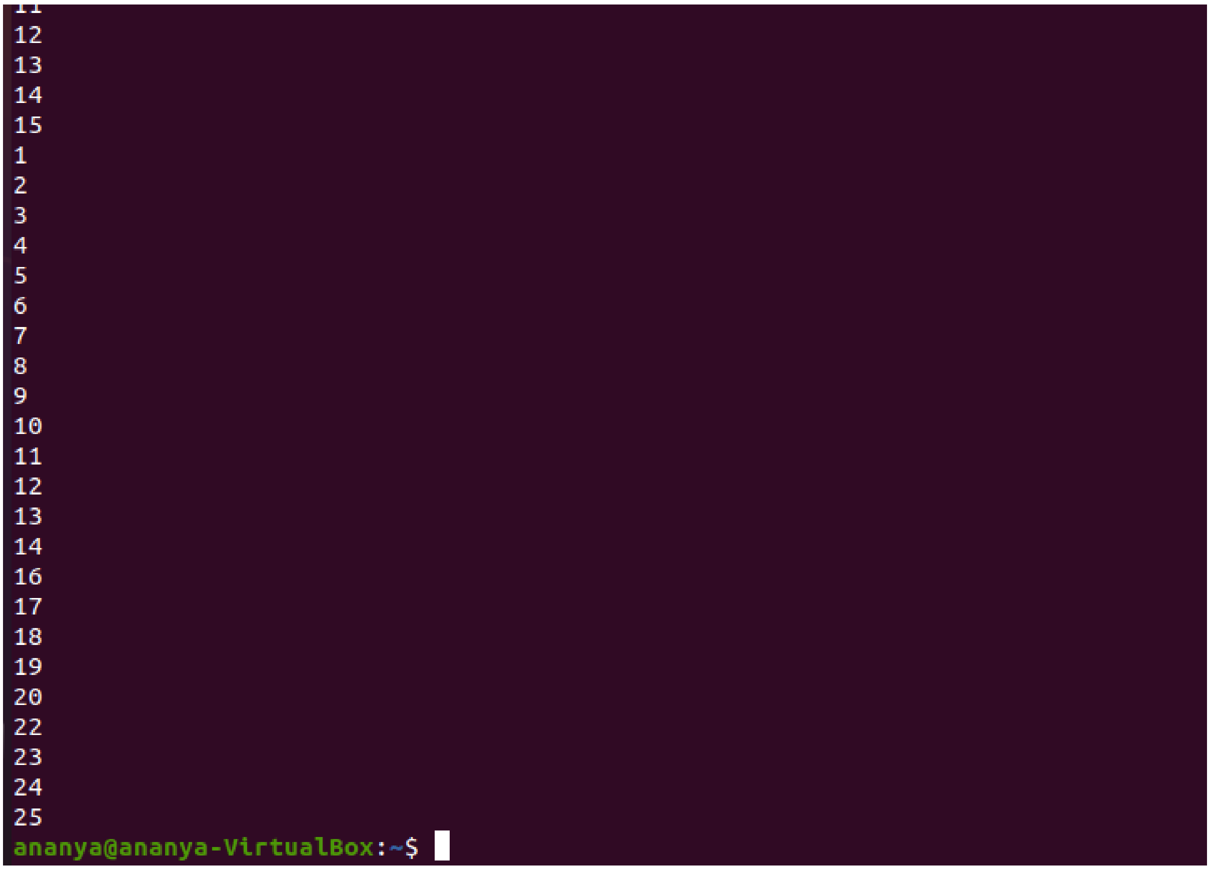
```
for var in word1 word2 ... wordN
do
   Statement(s) to be executed for every word.
done
```



```
for (( i=1 ; i<=25 ; i++ ))
do
if [ $i -gt 15 ]
then
break
fi
echo "$i"
done
for (( i=1 ; i<=25 ;i++ ))
do
if [ $i -eq 15 -o $i -eq 21 ]
then
continue
fi
echo "$i"
done
```



```
ananya@ananya-VirtualBox:~$ nano p.sh
ananya@ananya-VirtualBox:~$ ./p.sh
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
ananya@ananya-VirtualBox:~$ nano p.sh
ananya@ananya-VirtualBox:~$ ./p.sh
1
2
3
4
5
6
7
8
9
10
```



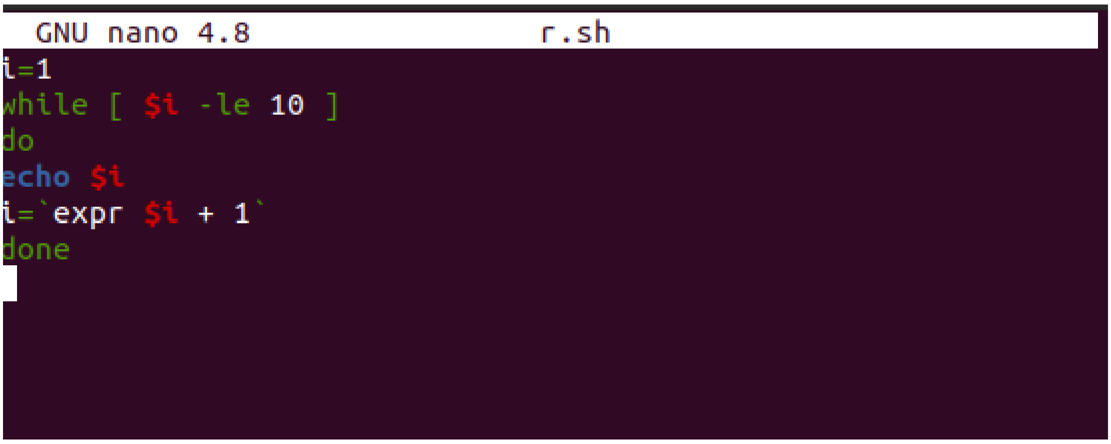
```
11
12
13
14
15
1
2
3
4
5
6
7
8
9
10
11
12
13
14
16
17
18
19
20
22
23
24
25
ananya@ananya-VirtualBox:~$
```

2) **while loop**

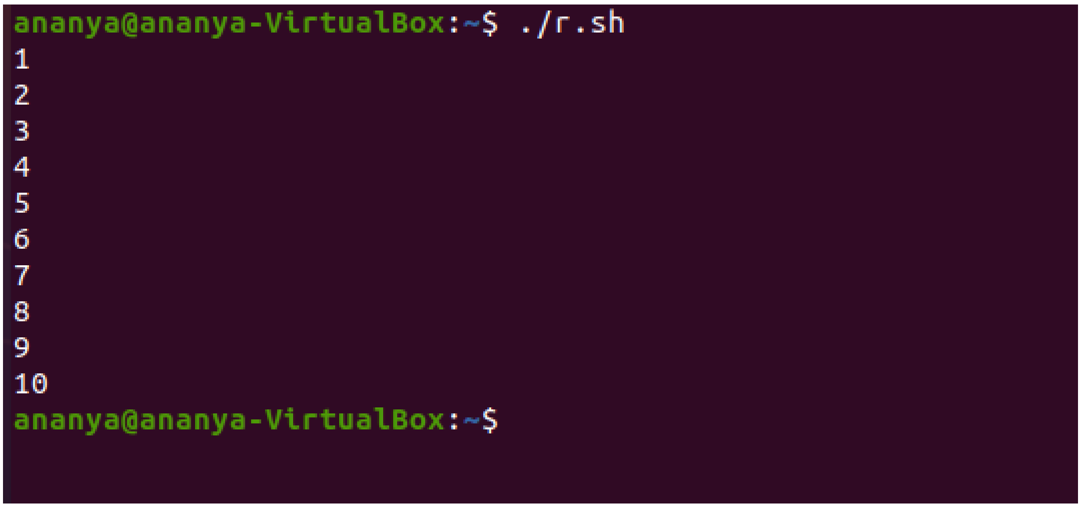
The while loop enables you to execute a set of commands repeatedly until some condition occurs. It is usually used when you need to manipulate the value of a variable repeatedly.

Syntax:

```
while command
do
   Statement(s) to be executed if command is true
Done
```



```
  GNU nano 4.8                    r.sh
i=1
while [ $i -le 10 ]
do
echo $i
i=`expr $i + 1`
done
```



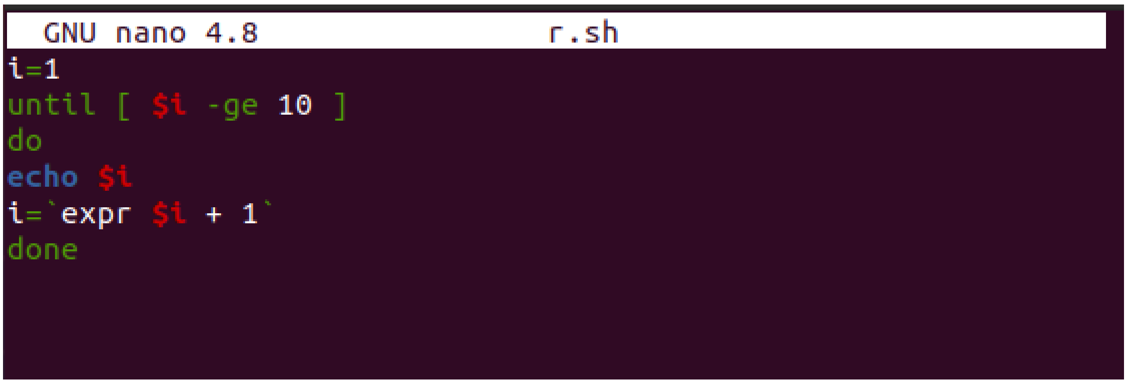
```
ananya@ananya-VirtualBox:~$ ./r.sh
1
2
3
4
5
6
7
8
9
10
ananya@ananya-VirtualBox:~$
```

3) **until loop**
   ***Act as an opposite loop**

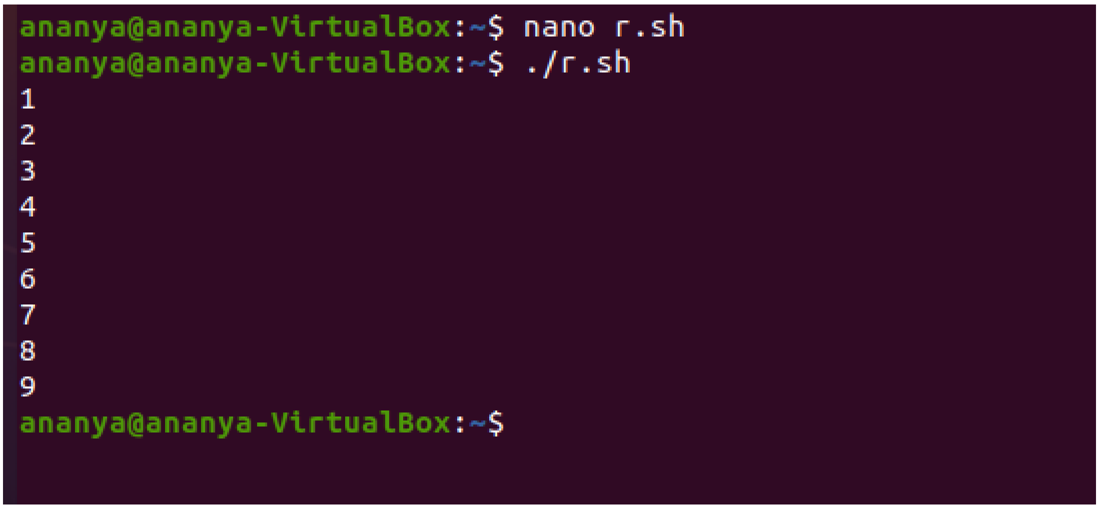
The while loop is perfect for a situation where you need to execute a set of commands while some condition is true. Sometimes you need to execute a set of commands until a condition is true.

Syntax:

```
until command
do
   Statement(s) to be executed until command is true
done
```



```
  GNU nano 4.8                     r.sh
i=1
until [ $i -ge 10 ]
do
echo $i
i=`expr $i + 1`
done
```



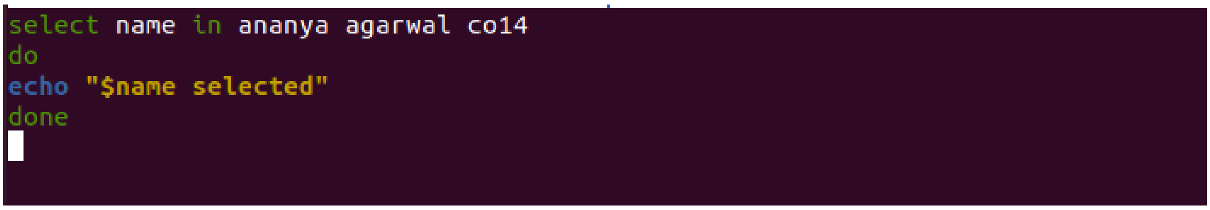
```
ananya@ananya-VirtualBox:~$ nano r.sh
ananya@ananya-VirtualBox:~$ ./r.sh
1
2
3
4
5
6
7
8
9
ananya@ananya-VirtualBox:~$
```

4) select loop

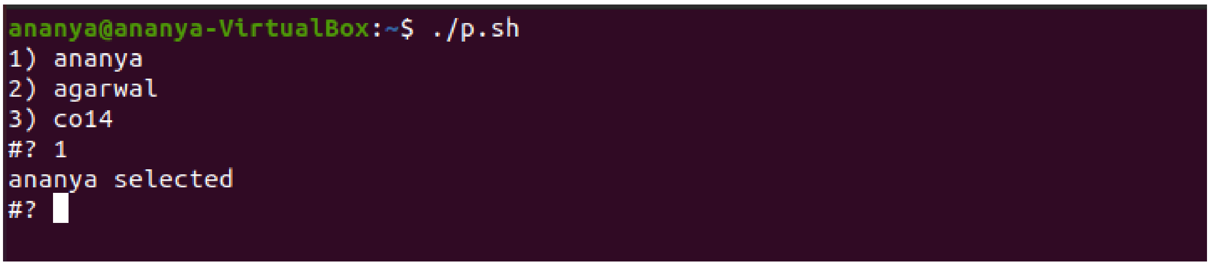
The select loop provides an easy way to create a numbered menu from which users can select options. It is useful when you need to ask the user to choose one or more items from a list of choices. The select loop is an infinite loop that only ends when there's a keyboard interrupt (Ctrl+C or Ctrl+Z) or a break statement is encountered.

Syntax:

```
select var in word1 word2 ... wordN
do
   Statement(s) to be executed for every word.
Done
```

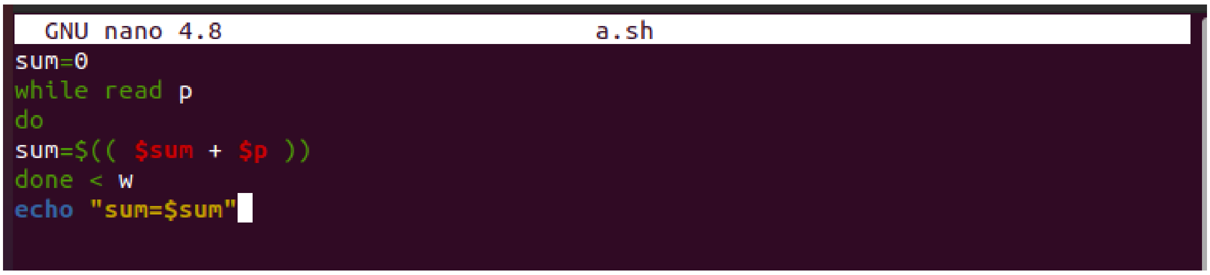


```
select name in ananya agarwal co14
do
echo "$name selected"
done
```

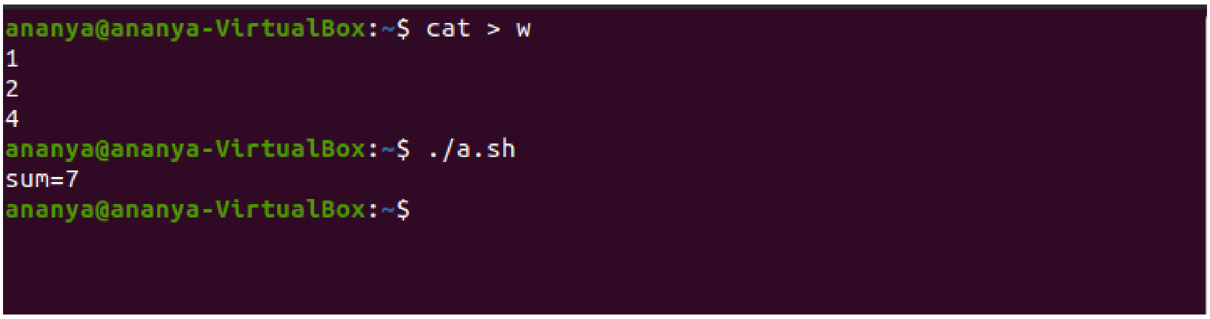


```
ananya@ananya-VirtualBox:~$ ./p.sh
1) ananya
2) agarwal
3) co14
#? 1
ananya selected
#?
```

Sol 6.

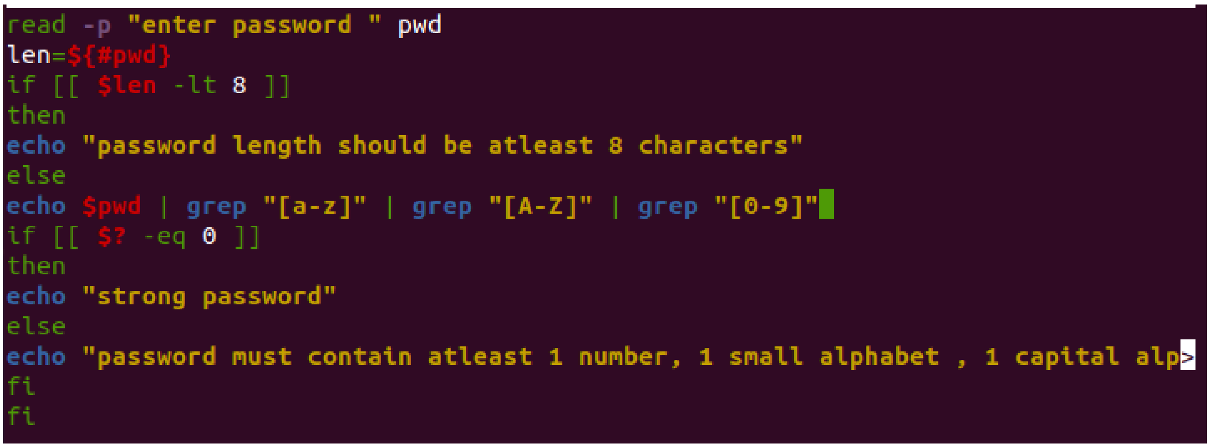


```
  GNU nano 4.8                              a.sh
sum=0
while read p
do
sum=$(( $sum + $p ))
done < w
echo "sum=$sum"
```

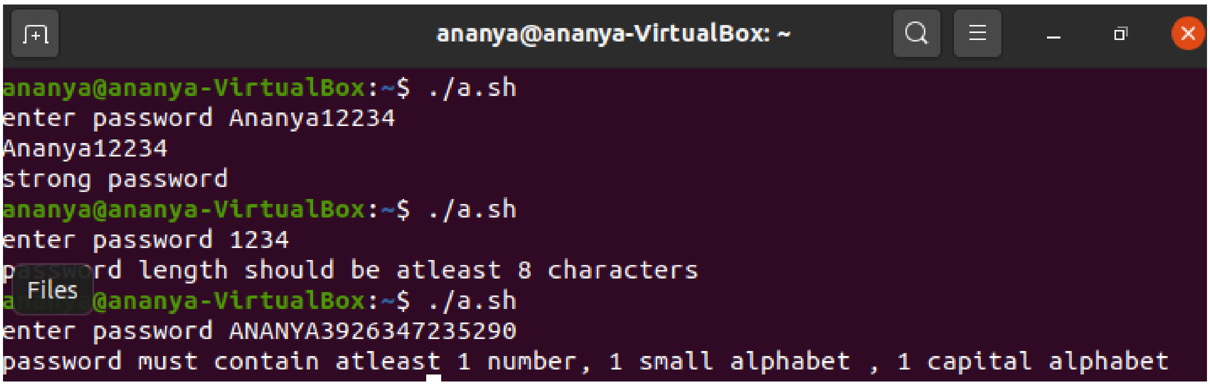


```
ananya@ananya-VirtualBox:~$ cat > w
1
2
4
ananya@ananya-VirtualBox:~$ ./a.sh
sum=7
ananya@ananya-VirtualBox:~$
```

Sol 7.



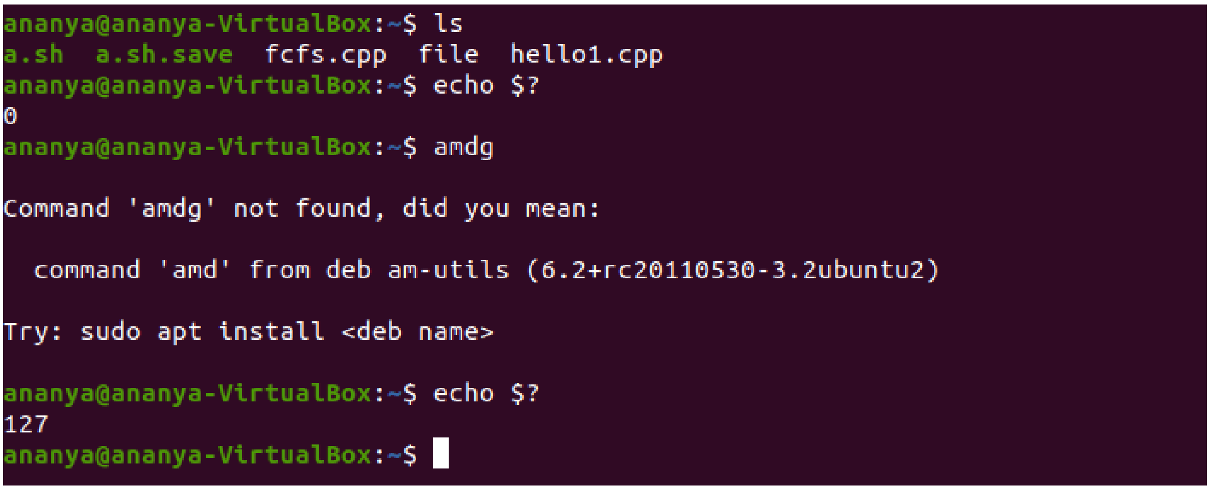
```
read -p "enter password " pwd
len=${#pwd}
if [[ $len -lt 8 ]]
then
echo "password length should be atleast 8 characters"
else
echo $pwd | grep "[a-z]" | grep "[A-Z]" | grep "[0-9]"
if [[ $? -eq 0 ]]
then
echo "strong password"
else
echo "password must contain atleast 1 number, 1 small alphabet , 1 capital alp>
fi
fi
```



```
ananya@ananya-VirtualBox:~$ ./a.sh
enter password Ananya12234
Ananya12234
strong password
ananya@ananya-VirtualBox:~$ ./a.sh
enter password 1234
password length should be atleast 8 characters
ananya@ananya-VirtualBox:~$ ./a.sh
enter password ANANYA3926347235290
password must contain atleast 1 number, 1 small alphabet , 1 capital alphabet
```

Sol 8.

Every command run in bash shell returns a value that's stored in the bash variable **"`$?`"**. To get that value, we run this command. If a command succeeded successfully, the return value will be 0. If the return value is otherwise **mainly 127**, then previous command was not running successfully.

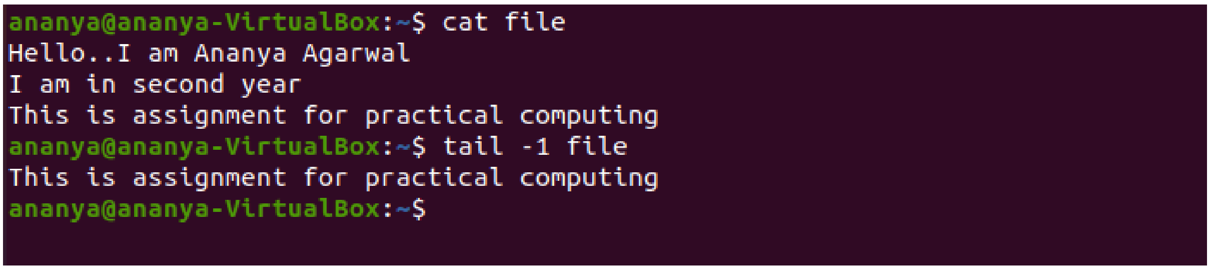


```
ananya@ananya-VirtualBox:~$ ls
a.sh  a.sh.save  fcfs.cpp  file  hello1.cpp
ananya@ananya-VirtualBox:~$ echo $?
0
ananya@ananya-VirtualBox:~$ amdg

Command 'amdg' not found, did you mean:

  command 'amd' from deb am-utils (6.2+rc20110530-3.2ubuntu2)

Try: sudo apt install <deb name>

ananya@ananya-VirtualBox:~$ echo $?
127
ananya@ananya-VirtualBox:~$ 
```

Sol 9.

The tail command can be used to display last 'n' lines from the end of a file. We have to use

`$ tail -1 filename` command to print the last line of a file.

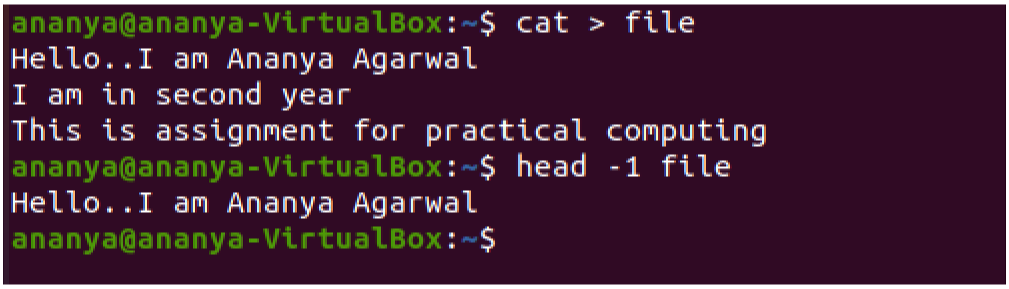


```
ananya@ananya-VirtualBox:~$ cat file
Hello..I am Ananya Agarwal
I am in second year
This is assignment for practical computing
ananya@ananya-VirtualBox:~$ tail -1 file
This is assignment for practical computing
ananya@ananya-VirtualBox:~$
```

Sol 10.

The head command can be used to display first 'n' lines from the beginning of a file. We have to use

`$ head -1 filename` command to print the first line of a file.



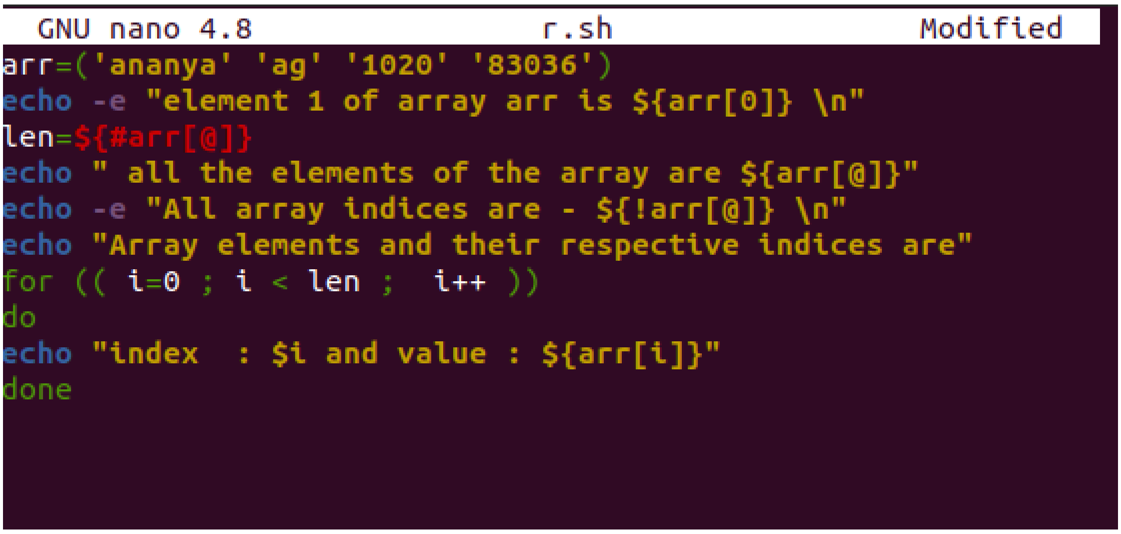
```
ananya@ananya-VirtualBox:~$ cat > file
Hello..I am Ananya Agarwal
I am in second year
This is assignment for practical computing
ananya@ananya-VirtualBox:~$ head -1 file
Hello..I am Ananya Agarwal
ananya@ananya-VirtualBox:~$
```

Sol 11.

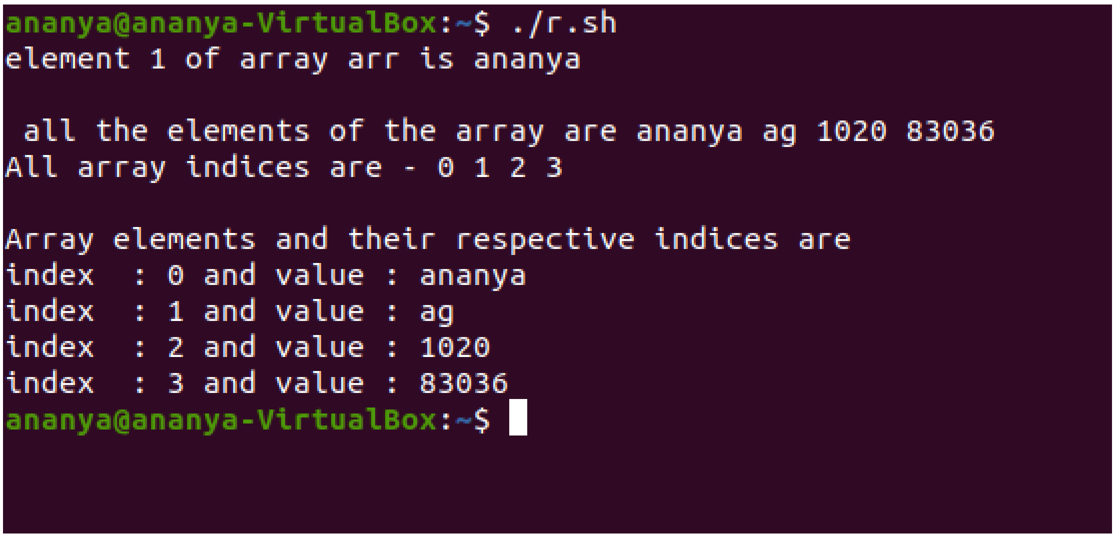
Value of the first element of array by : `${array_name[0]}`

Print values of all array elements by : `echo “${array_name[@]}”`

Print values of all array indices using `echo “${!array_name[@]}”`



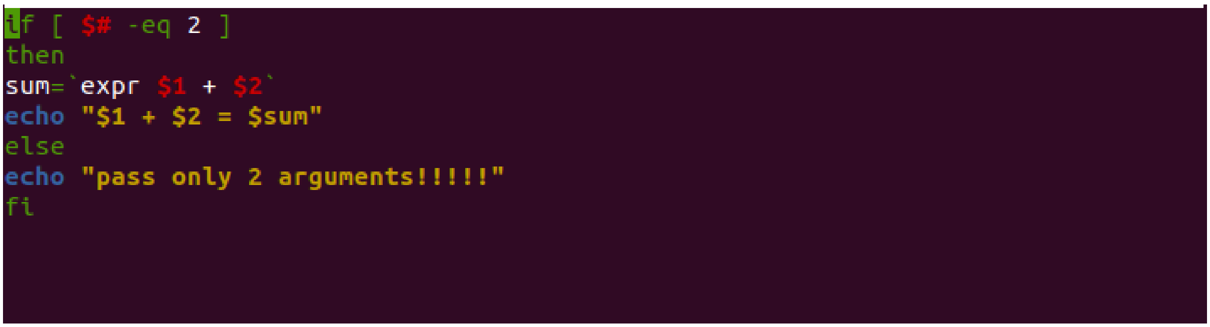
```
  GNU nano 4.8                    r.sh                     Modified
arr=('ananya' 'ag' '1020' '83036')
echo -e "element 1 of array arr is ${arr[0]} \n"
len=${#arr[@]}
echo " all the elements of the array are ${arr[@]}"
echo -e "All array indices are - ${!arr[@]} \n"
echo "Array elements and their respective indices are"
for (( i=0 ; i < len ;  i++ ))
do
echo "index  : $i and value : ${arr[i]}"
done
```



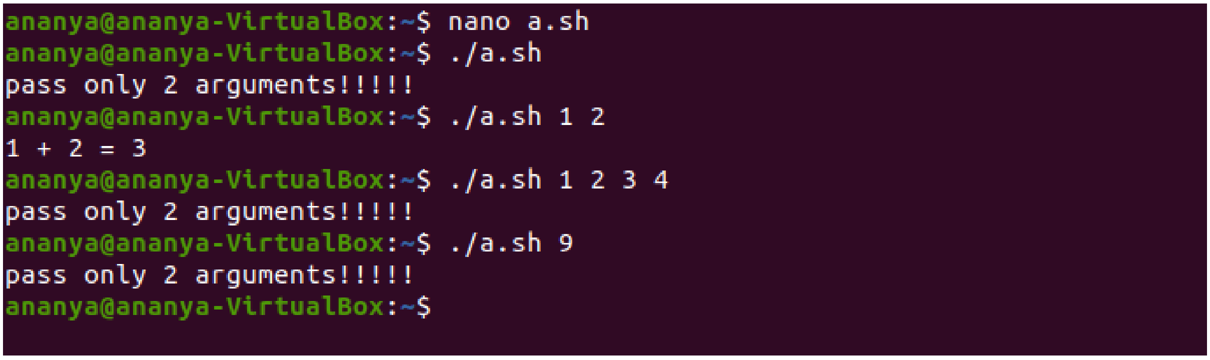
```
ananya@ananya-VirtualBox:~$ ./r.sh
element 1 of array arr is ananya

 all the elements of the array are ananya ag 1020 83036
All array indices are - 0 1 2 3

Array elements and their respective indices are
index  : 0 and value : ananya
index  : 1 and value : ag
index  : 2 and value : 1020
index  : 3 and value : 83036
ananya@ananya-VirtualBox:~$ 
```

Sol 12.

We can check the number of arguments passed to a script using keyword `$#`

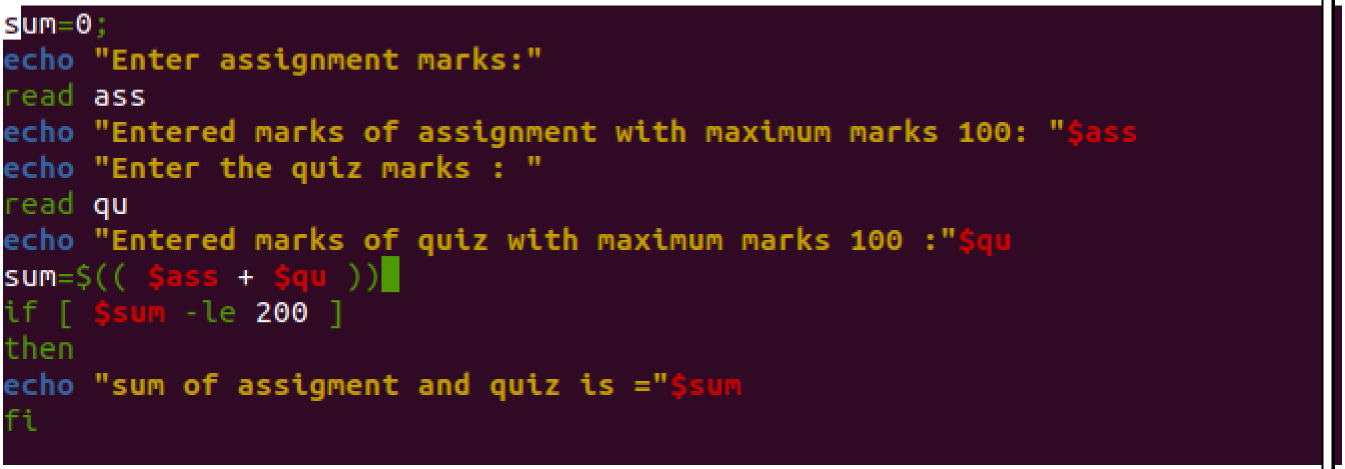


```
if [ $# -eq 2 ]
then
sum=`expr $1 + $2`
echo "$1 + $2 = $sum"
else
echo "pass only 2 arguments!!!!!"
fi
```

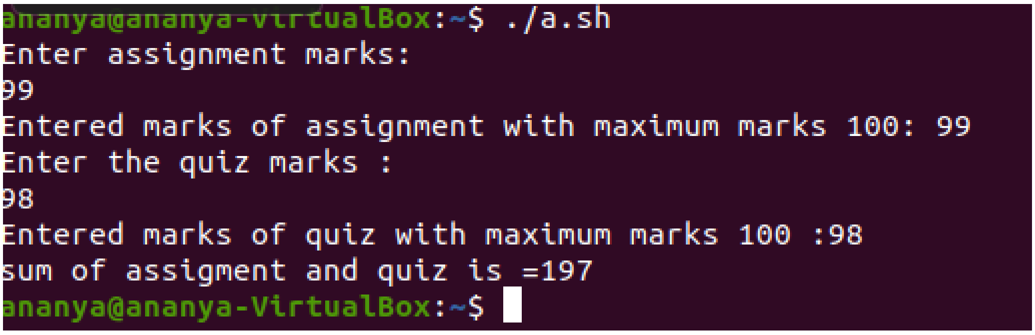


```
ananya@ananya-VirtualBox:~$ nano a.sh
ananya@ananya-VirtualBox:~$ ./a.sh
pass only 2 arguments!!!!!
ananya@ananya-VirtualBox:~$ ./a.sh 1 2
1 + 2 = 3
ananya@ananya-VirtualBox:~$ ./a.sh 1 2 3 4
pass only 2 arguments!!!!!
ananya@ananya-VirtualBox:~$ ./a.sh 9
pass only 2 arguments!!!!!
ananya@ananya-VirtualBox:~$
```

Sol 13.

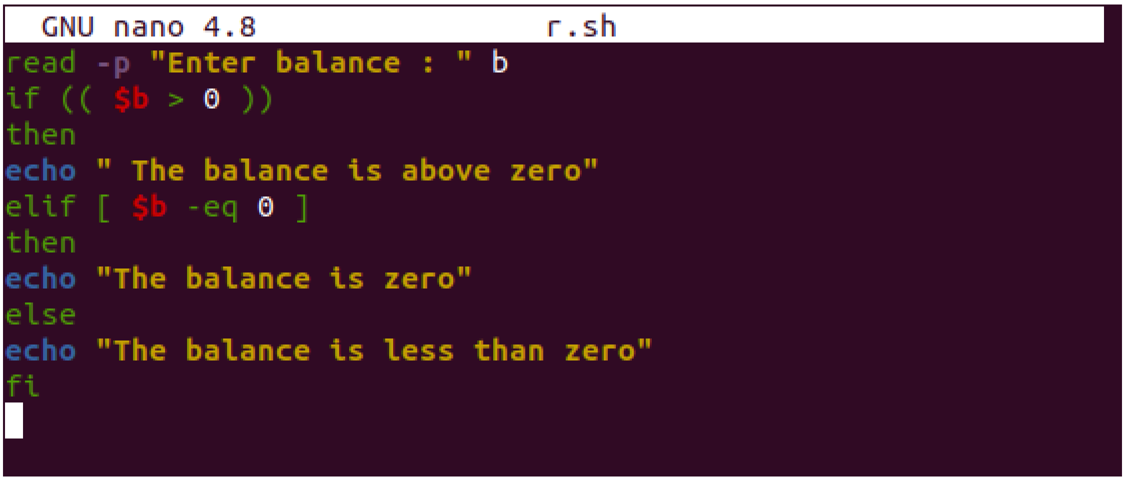


```
sum=0;
echo "Enter assignment marks:"
read ass
echo "Entered marks of assignment with maximum marks 100: "$ass
echo "Enter the quiz marks : "
read qu
echo "Entered marks of quiz with maximum marks 100 :"$qu
sum=$(( $ass + $qu ))
if [ $sum -le 200 ]
then
echo "sum of assigment and quiz is ="$sum
fi
```

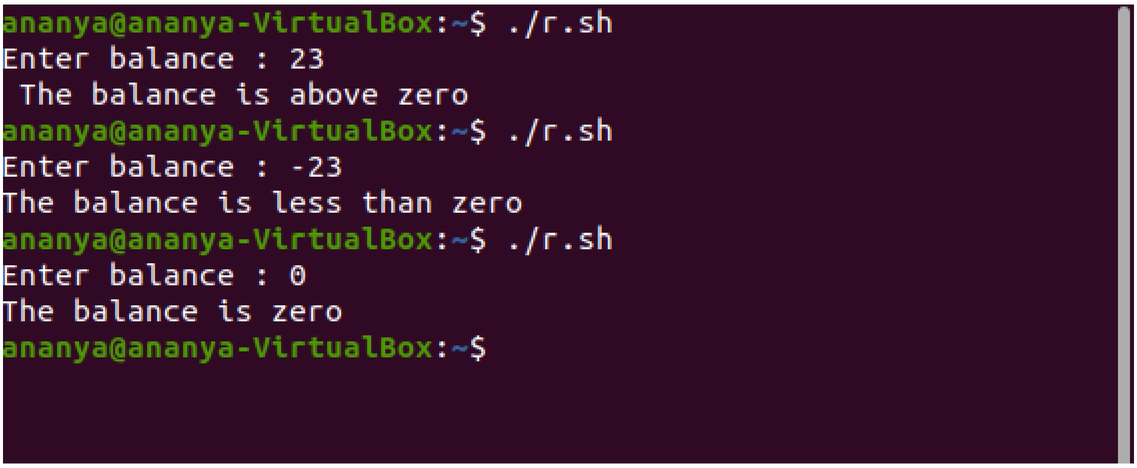


```
ananya@ananya-VirtualBox:~$ ./a.sh
Enter assignment marks:
99
Entered marks of assignment with maximum marks 100: 99
Enter the quiz marks :
98
Entered marks of quiz with maximum marks 100 :98
sum of assigment and quiz is =197
ananya@ananya-VirtualBox:~$
```

Sol 14.



```
  GNU nano 4.8                r.sh
read -p "Enter balance : " b
if (( $b > 0 ))
then
echo " The balance is above zero"
elif [ $b -eq 0 ]
then
echo "The balance is zero"
else
echo "The balance is less than zero"
fi
```



```
ananya@ananya-VirtualBox:~$ ./r.sh
Enter balance : 23
 The balance is above zero
ananya@ananya-VirtualBox:~$ ./r.sh
Enter balance : -23
The balance is less than zero
ananya@ananya-VirtualBox:~$ ./r.sh
Enter balance : 0
The balance is zero
ananya@ananya-VirtualBox:~$
```