LInux Assignment 1 - DAC(sep-2022)

1. Practice the below-mentioned commands with all the possible options:
   cd, cat, ls, mkdir, rmdir, pwd, mv, cp, rm.

```
nischal at anton in ~
$ cd Demo/

nischal at anton in ~/Demo
$ ls
a.out  dir  helloworld.c  myScript.sh  q2  q2.c  q3  q3.c  q4  q4.c  q5  q5.c  test  text

nischal at anton in ~/Demo
$ ls -al
total 116
drwxrwxr-x  3 nischal nischal  4096 Sep 25 00:49 .
drwxr-xr-x 56 nischal nischal  4096 Sep 25 00:49 ..
-rwxrwxr-x  1 nischal nischal 16120 Sep 24 22:38 a.out
drwxrwxr-x  2 nischal nischal  4096 Sep 23 16:35 dir
-rw-rw-r--  1 nischal nischal   123 Sep 24 16:02 helloworld.c
-rwxrw-r--  1 nischal nischal   346 Sep 24 01:47 myScript.sh
-rwxrwxr-x  1 nischal nischal 16072 Sep 24 21:28 q2
-rw-rw-r--  1 nischal nischal   290 Sep 24 21:28 q2.c
-rwxrwxr-x  1 nischal nischal 16160 Sep 24 22:45 q3
-rw-rw-r--  1 nischal nischal   430 Sep 24 22:45 q3.c
-rwxrwxr-x  1 nischal nischal 16160 Sep 24 23:15 q4
-rw-rw-r--  1 nischal nischal   341 Sep 24 23:14 q4.c
-rwxrwxr-x  1 nischal nischal 15992 Sep 25 00:49 q5
-rw-rw-r--  1 nischal nischal   224 Sep 25 00:03 q5.c
-rwxrw-r--  1 nischal nischal     0 Sep 23 17:14 test
-rw-rw-r--  1 nischal nischal     0 Sep 23 17:39 text
```

```
nischal at anton in ~/Demo
$ pwd
/home/nischal/Demo
```

```
nischal at anton in ~/Demo
$ mv test ./dir/
```

```
nischal at anton in ~/Demo
$ cd dir/

nischal at anton in ~/Demo/dir
$ rm file1 file2 test

nischal at anton in ~/Demo/dir
$ cd ..

nischal at anton in ~/Demo
$ rmdir dir
```

```
nischal at anton in ~/Demo
$ mkdir dir

nischal at anton in ~/Demo
$ ls
a.out  dir  helloworld.c  myScript.sh  q2  q2.c  q3  q3.c  q4  q4.c  q5  q5.c  test.txt  text

nischal at anton in ~/Demo
$
```

```
nischal at anton in ~/Demo
$ ls
a.out  helloworld.c  myScript.sh  q2  q2.c  q3  q3.c  q4  q4.c  q5  q5.c  text

nischal at anton in ~/Demo
$ cp text test.txt

nischal at anton in ~/Demo
$ ls
a.out  helloworld.c  myScript.sh  q2  q2.c  q3  q3.c  q4  q4.c  q5  q5.c  test.txt  text
```

2. Enter two commands echo* and ls. What do you think the echo did?
-> echo printed all the files and directories inside the current directory because of the wildcard character which is “*” it selects everything.

3.what does cd do when used without arguments?
-> it takes us to the home directory.

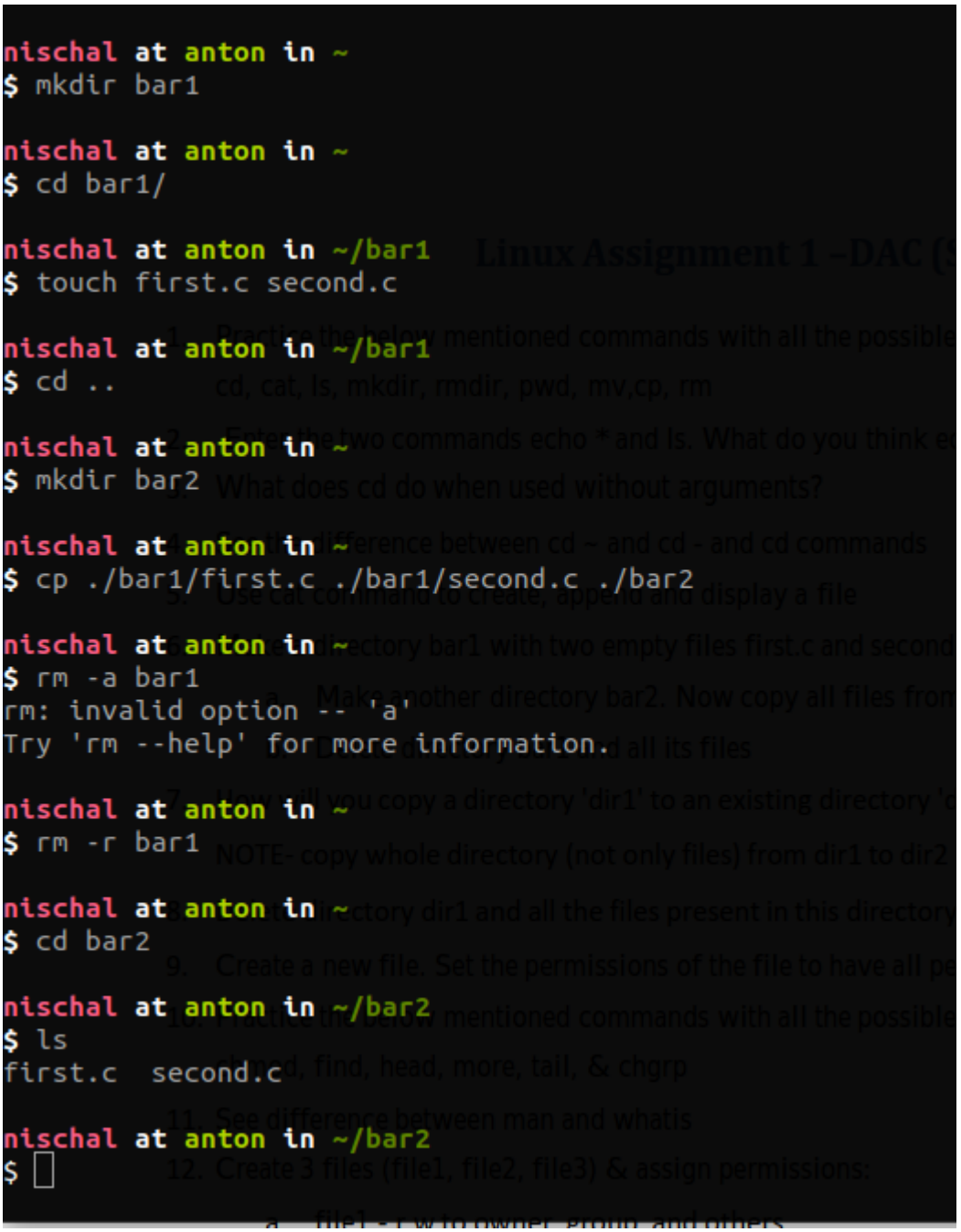
4. See the difference between cd ~ and cd - and cd commands.
-> cd ~ takes you to the home directory
cd - takes to the last directory in your history

5. Use cat command to create append and display a file.

```
~

nischal at anton in ~
$ cat > newfile
hello world

nischal at anton in ~
$ cat >>newfile
another line

nischal at anton in ~
$ cat newfile
hello world
another line

nischal at anton in ~
$
```

6.make a directory bar1 with two empty files first.c and second.c

a. Make another directory bar2 now copy all files form bar1 to bar2.
b. Delete directory bar1 and all its files.



7. How will you copy dir1 to existing dir 2?

Using cp command:

cp dir1 ./dir2

8.delete directory dir1 and all the files present in this directory.

```
nischal at anton in ~
$ mkdir dir1

nischal at anton in ~
$ cd dir1

nischal at anton in ~/dir1
$ touch file file2 file3

nischal at anton in ~/dir1
$ cd ..

nischal at anton in ~
$ rm -r dir1
```

9.create a new file. Set the permissions of the file to have all permissions for yourself.

```
nischal at anton in ~
$ touch newfile

nischal at anton in ~
$ ls
Android                 Documents           Pictures
AndroidStudioProjects   Downloads           Public
backup                  eclipse-workspace   react
bar2                    IdeaProjects        snap
c++                     init                Templates
codes                   Music               'udo systemctl stop mongod'
DataGripProjects        newfile             Videos
Demo                    Node                zoom
Desktop                 OrigDir

nischal at anton in ~
$ chmod u+rwx newfile

nischal at anton in ~
$ ls -l newfile
-rwxrw-r-- 1 nischal nischal 25 Sep 25 13:40 newfile

nischal at anton in ~
$
```

10.practice below commands:

Chmod, find, head, more, tail and chgrp

Chmod is used the modify the file permission
Find is used to search a file in a directory can be use for many other cases too
Head returns first 10 lines of a text file by default you can pass -n flag to modify the lines of output it produces.

Tail command is similar to head but it returns the last 10 lines by default it can also be modifies using -n flag.

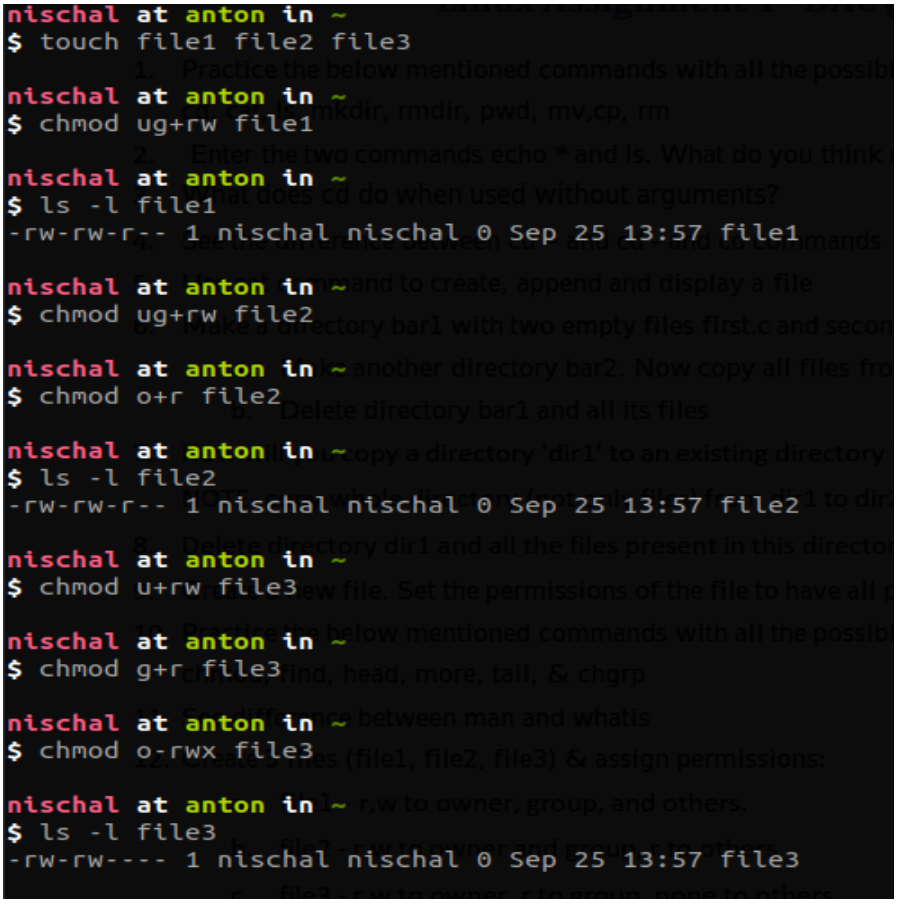
chgrp command is lets you modify the group ownership of a file or directory similar to chwon but its for group instead of user.

11. See the difference between man and whatis.

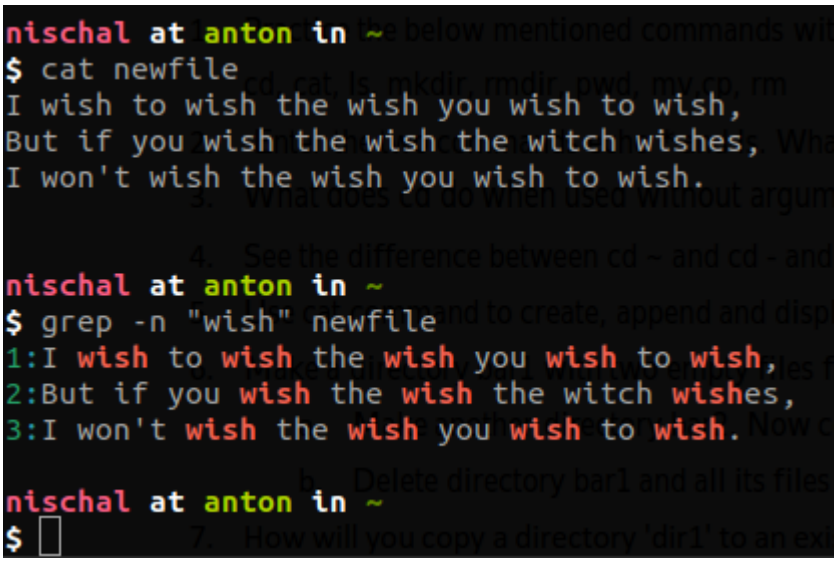
Man is used to see manual of commands.

Whatis is one line manual it does not shows detailed view like man does.

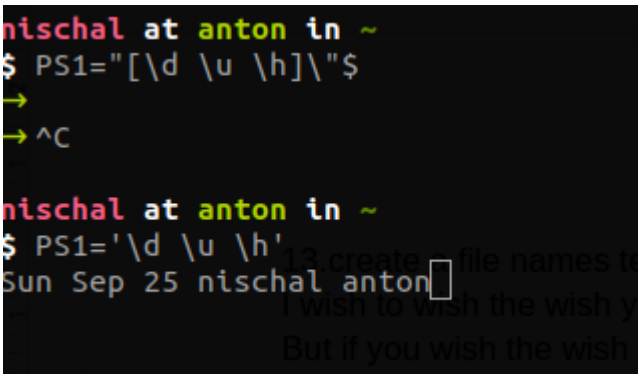
12.create 3 file (file1 file2 file3)& assign permissions.



13.create a file names test.txt having following lines:
I wish to wish the wish you wish to wish,
But if you wish the wish the witch wishes,
I won't wish the wish you wish to wish.
Now use grep command to print the lines matching words wish.



14. Chage the default system variable PS! To a new value containing the current date along with username and hostname.



15.Explore all the commands with all the options taught in today's class
-> Successfully explored all the commands and their variations.

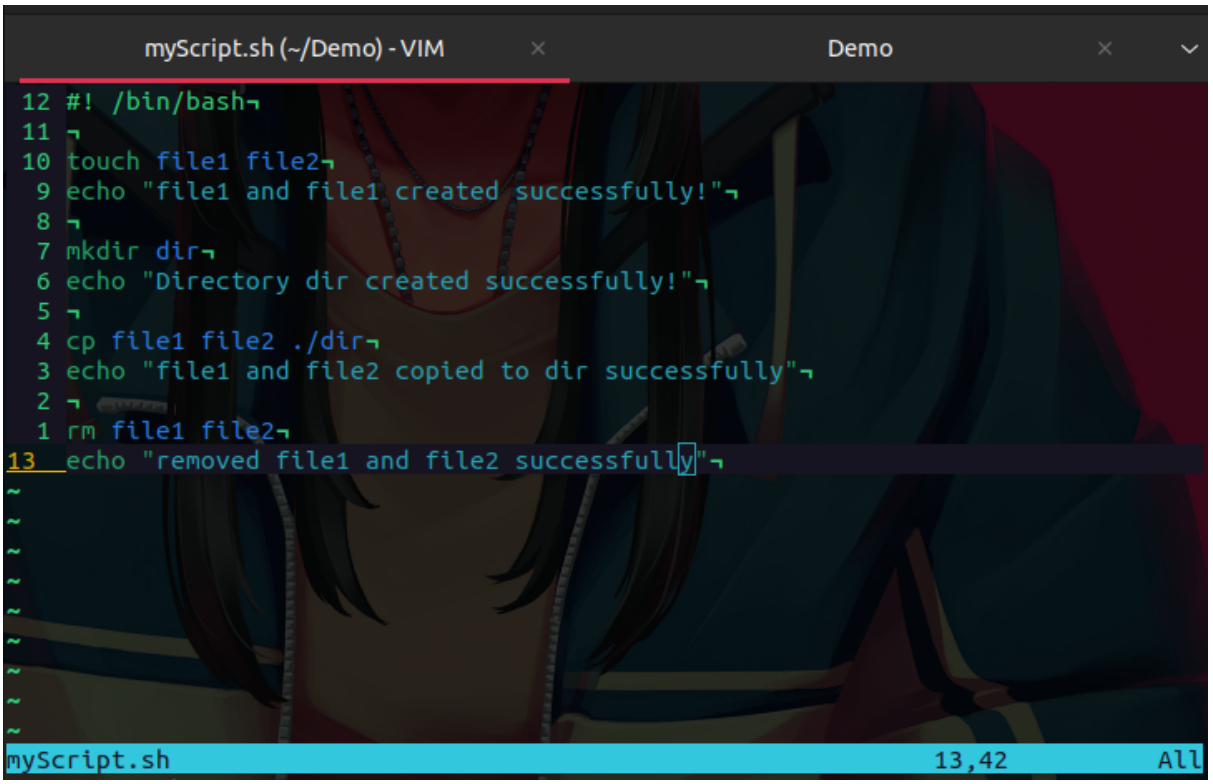
Assignment 2

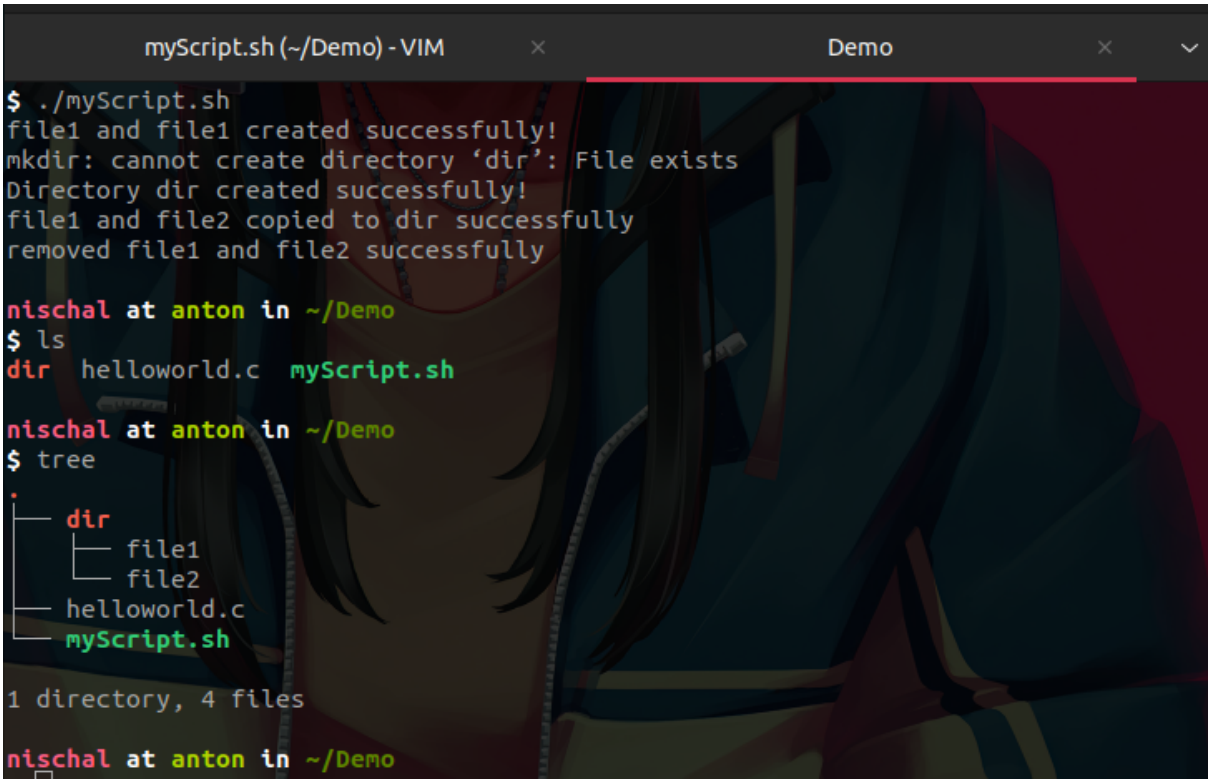
1.Redirect both the output and error of a command to a file

```
$ eccho "error simulation" 2> newfile.txt

nischal at anton in ~/Demo
$ cat newfile.txt
Command 'eccho' not found, did you mean:
  command 'echo' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>

nischal at anton in ~/Demo
$ echo "Hello World!" 2> newfile.txt
Hello World!

nischal at anton in ~/Demo
$ cat newfile.txt

nischal at anton in ~/Demo
$ echo "Hello World!" > newfile.txt

nischal at anton in ~/Demo
$ cat newfile.txt
Hello World!

nischal at anton in ~/Demo
$
```

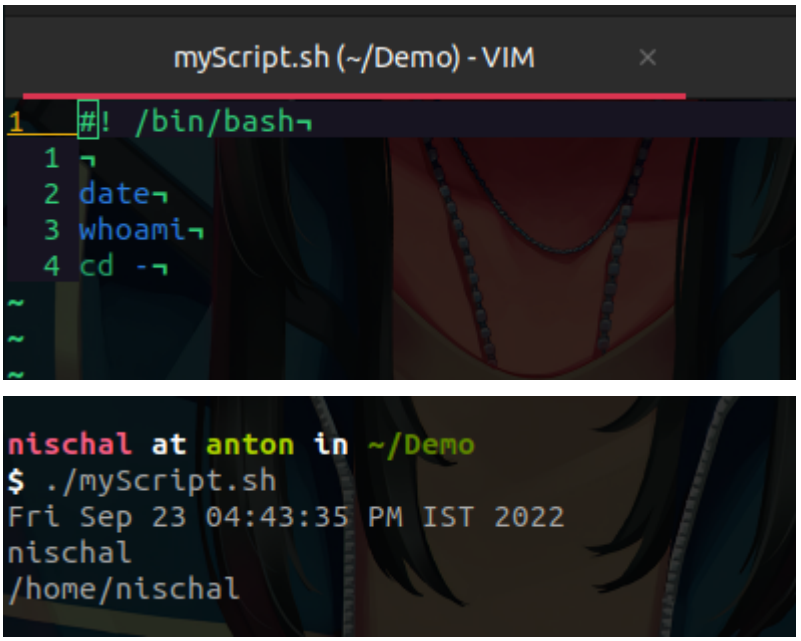
2.Include your current directory in the PATH environment variable using export command

3.Create 2 files “file1” and “file2”. Create a directory “dir”. Copy the files to the directory and then delete the files. Do so using shell script.

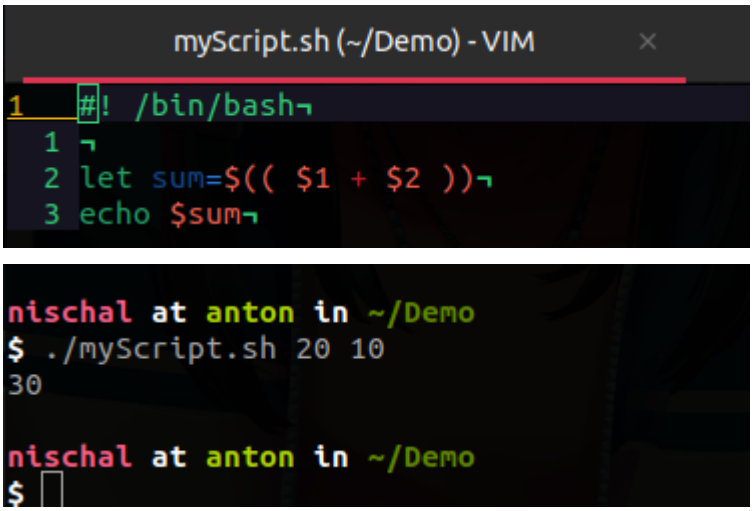




4. Write a script to see current date, time, username and current directory.



5.Write shell script that will add two numbers, which are supplied as command line argument.



6.Write a script to determine whether given file exists or not, the file name is supplied as command line argument, also check for sufficient number of command line arguments.

myScript.sh (~/Demo) - VIM

```
1  #! /bin/bash
  1
  2 if [[ $# > 1 || $# < 1 ]];
  3 then
  4 ▸ echo "Please enter atmost and atleast one file name"
  5 ▸ exit
  6 fi
  7
  8 FILE=$1
  9
 10 if [[ -f "$FILE" ]];
 11 then
 12 ▸ echo "$FILE exists."
 13 else
 14 ▸ echo "$FILE does not exists"
 15 fi
 16
~
~
~
```

```
nischal at anton in ~/Demo
$ ./myScript.sh test test
Please enter atmost and atleast one file name

nischal at anton in ~/Demo
$ ./myScript.sh test
test exists.

nischal at anton in ~/Demo
$ ./myScript.sh helloworld.c
helloworld.c exists.

nischal at anton in ~/Demo
$
```

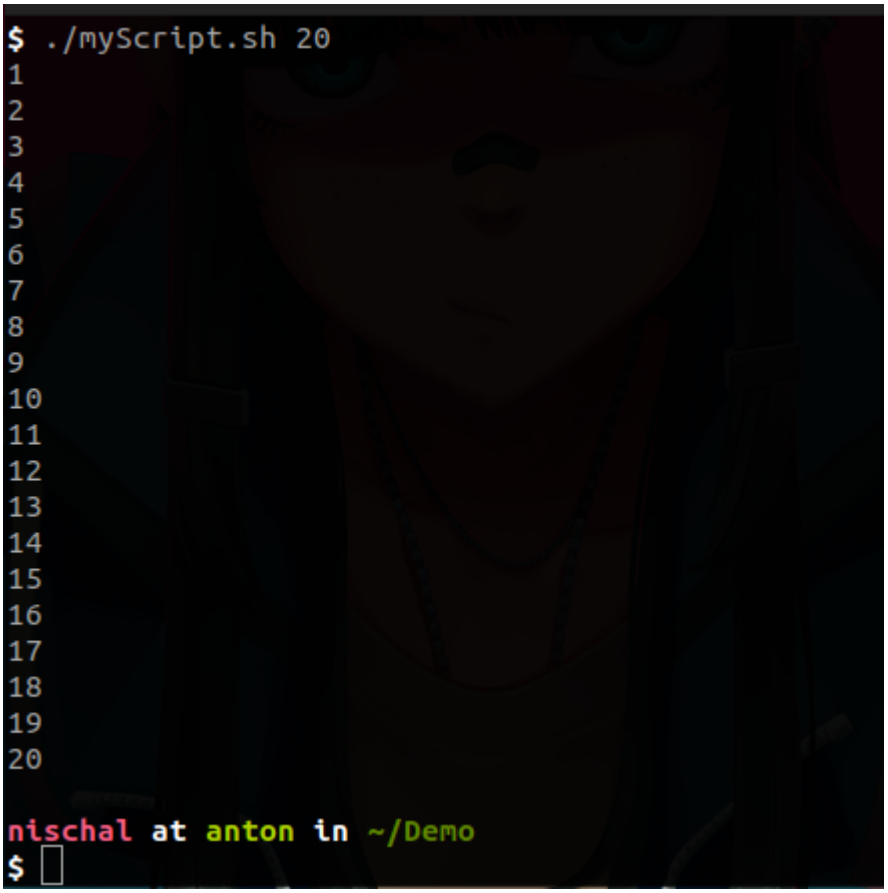
7.Write a script to print nos. as 5, 4, 3, 2, 1 using while loop.

```
1  #! /bin/bash
  1
  2 NUM=5
  3
  4 while [ $NUM -ge 1 ]
  5 do
  6    echo $NUM
  7    NUM=`expr $NUM - 1`
  8 done
```

```
nischal at anton in ~/Demo
$ ./myScript.sh
5
4
3
2
1

nischal at anton in ~/Demo
$
```

8.Take a number as command line and using until loop print value from 1 to till number.

```
#! /bin/bash
NUM=1
while [ $NUM -le $1 ]
do
    echo "$NUM"
    NUM=`expr $NUM + 1`
done
~
~
~
~
~
~
~
~
~
~
~
~
~
myScript.sh
"myScript.sh" 9L, 85B
```



9.Write a script, using case statement to perform basic math operation as follows
+ addition
-subtraction
x multiplication
/ division

NOTE- two numbers will be given as command line arguments.

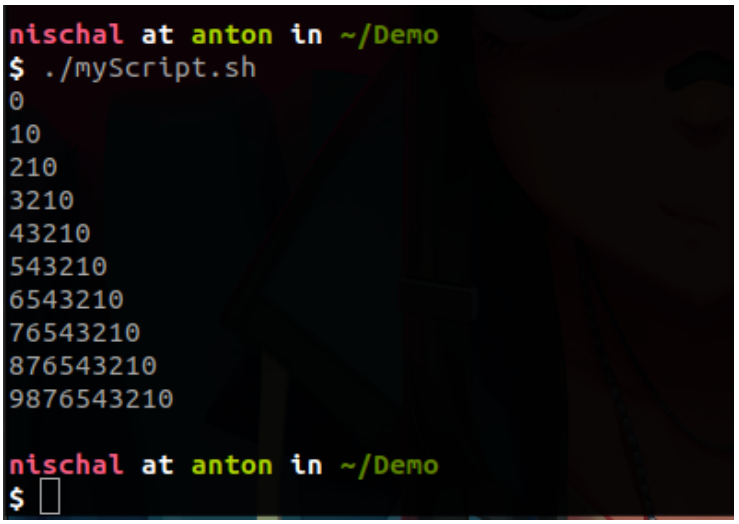
```
1  #! /bin/bash¬
 1 ¬
 2 read -p "Enter 1->add 2->sub 3->mul 4->div :" NUM¬
 3 echo "$1  $2"¬
 4 ¬
 5 case $NUM in¬
 6 ¬
 7 ▶ 1)¬
 8 ▶ ▶ echo "$1 + $2 = $(($1+$2))"¬
 9 ▶ ▶ ;;¬
10 ▶ 2)¬
11 ▶ ▶ echo "$1 - $2 = $(($1-$2))"¬
12 ▶ ▶ ;;¬
13 ▶ 3)¬
14 ▶ ▶ echo "$1 * $2 = $(($1*$2))"¬
15 ▶ ▶ ;;¬
16 ▶ 4)¬
17 ▶ ▶ echo "$1 / $2 = $(($1/$2))"¬
18 ▶ ▶ ;;¬
19 ▶ *)¬
20 ▶ ▶ echo "please enter a valid number between 1 and 4"¬
21 esac¬
~
~
~
myScript.sh                                                    1,1
"myScript.sh" 22L, 311B
```

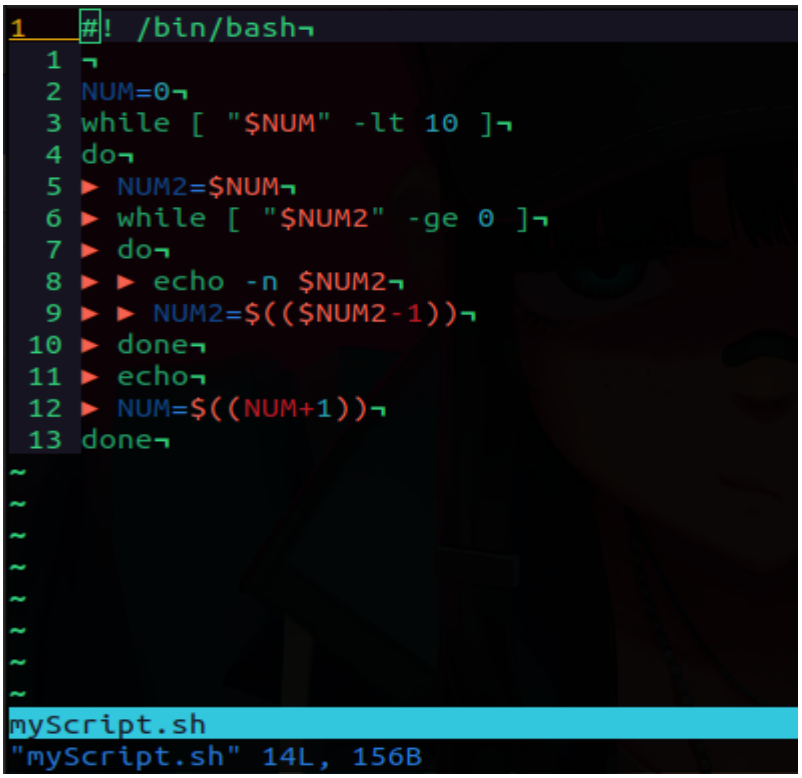
```
nischal at anton in ~/Demo
$ ./myScript.sh 9 9
Enter 1->add 2->sub 3->mul 4->div :4
9  9
9 / 9 = 1

nischal at anton in ~/Demo
$
```

10.Write a script to find out biggest number from three given numbers. Numbers are supplied as command line arguments. Print error if sufficient arguments are not supplied.

11.Write a program using while loop to print

0
1 0
2 1 0
3 2 1 0
4 3 2 1 0
5 4 3 2 1 0
6 5 4 3 2 1 0
7 6 5 4 3 2 1 0
8 7 6 5 4 3 2 1 0
9 8 7 6 5 4 3 2 1 0





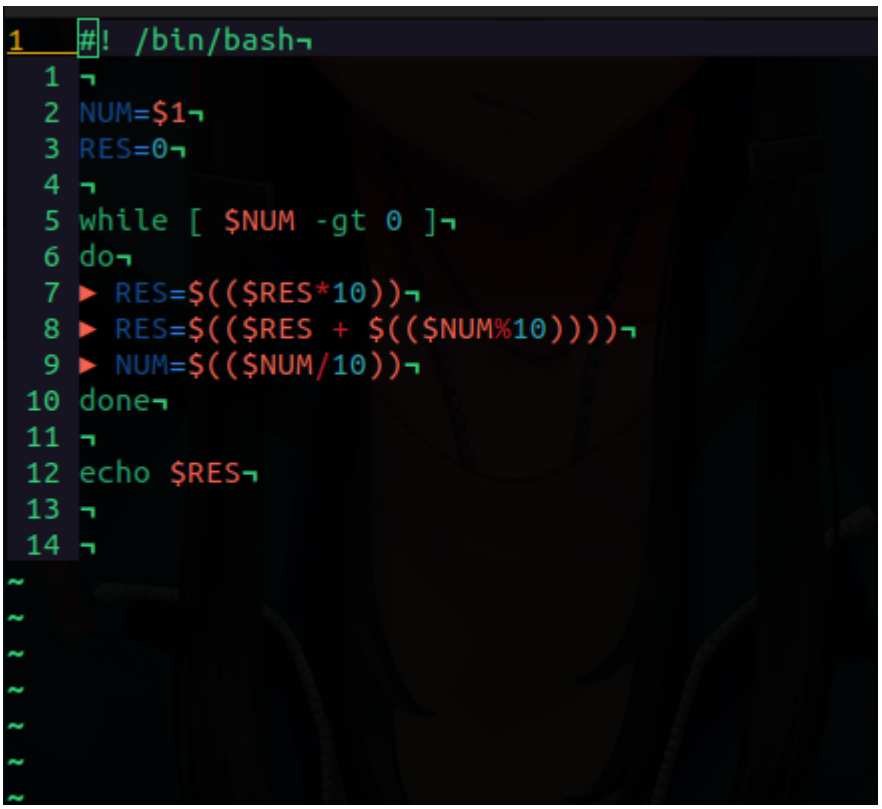
12.Write a program using for loop to print

```
*
*
* *
* * *
* * * *
* * * *
* * *
* *
*
*
```

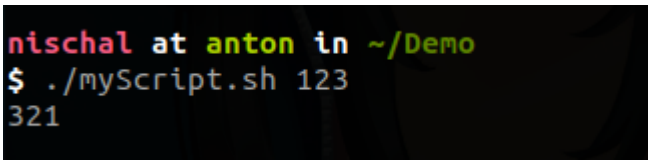
```
#! /bin/bash
for NUMBER in 1 1 2 3 4 4 3 2 1 1
do
    for((  NUM=0; NUM<$NUMBER; NUM++  ))
    do
        echo -n "*"
    done
    echo
done
```

```
nischal at anton in ~/Demo
$ ./myScript.sh
*
*
**
***
****
****
***
**
*
*

nischal at anton in ~/Demo
$
```

13. Write a script to print given number in reverse order, for eg. If number is 123, it must print as 321.



```
#! /bin/bash
NUM=$1
RES=0
while [ $NUM -gt 0 ]
do
    RES=$(($RES*10))
    RES=$(($RES + $(($NUM%10))))
    NUM=$(($NUM/10))
done
echo $RES
```



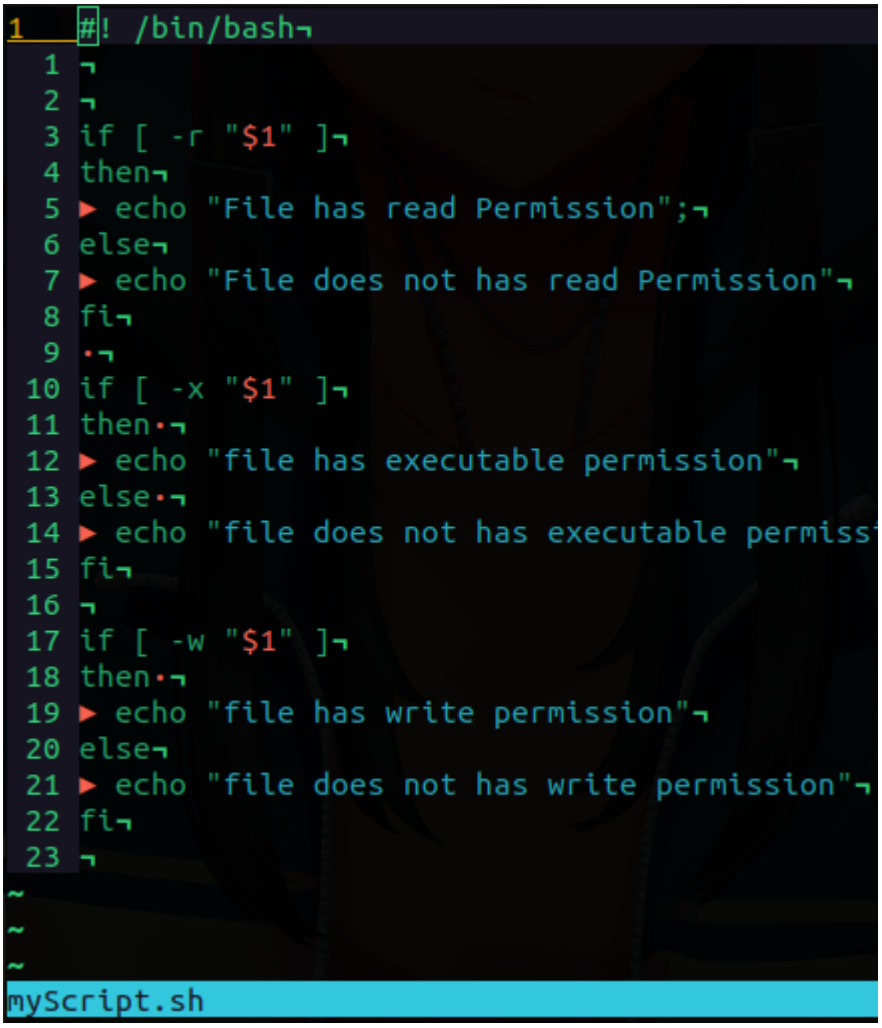
```
nischal at anton in ~/Demo
$ ./myScript.sh 123
321
```

14.Write script to print the sum of all the digits of a given number. For eg. If the number is 123, sum of all the digits will be 1+2+3 = 6.

```
#! /bin/bash
NUM=$1
RES=0

while [ $NUM -gt 0 ]
do
    RES=$(($RES + ($NUM%10)))
    NUM=$(($NUM/10))
done

echo $RES


~
~
~
~
~
~
~
~
```

```
nischal at anton in ~/Demo
$ ./myScript.sh 66
12

nischal at anton in ~/Demo
$
```

15. Create a file named file.txt and write a shell script to check is a file is readable, writable and executable.



```
1   #! /bin/bash¬
  1 ¬
  2 ¬
  3 if [ -r "$1" ]¬
  4 then¬
  5 ▶ echo "File has read Permission";¬
  6 else¬
  7 ▶ echo "File does not has read Permission"¬
  8 fi¬
  9 ·¬
 10 if [ -x "$1" ]¬
 11 then·¬
 12 ▶ echo "file has executable permission"¬
 13 else·¬
 14 ▶ echo "file does not has executable permiss
 15 fi¬
 16 ¬
 17 if [ -w "$1" ]¬
 18 then·¬
 19 ▶ echo "file has write permission"¬
 20 else¬
 21 ▶ echo "file does not has write permission"¬
 22 fi¬
 23 ¬
~
~
~
myScript.sh
```

```
nischal at anton in ~/Demo
$ ./myScript.sh text
File has read Permission
file does not has executable permission
file has write permission

nischal at anton in ~/Demo
$ ./myScript.sh test
File has read Permission
file has executable permission
file has write permission

nischal at anton in ~/Demo
$
```