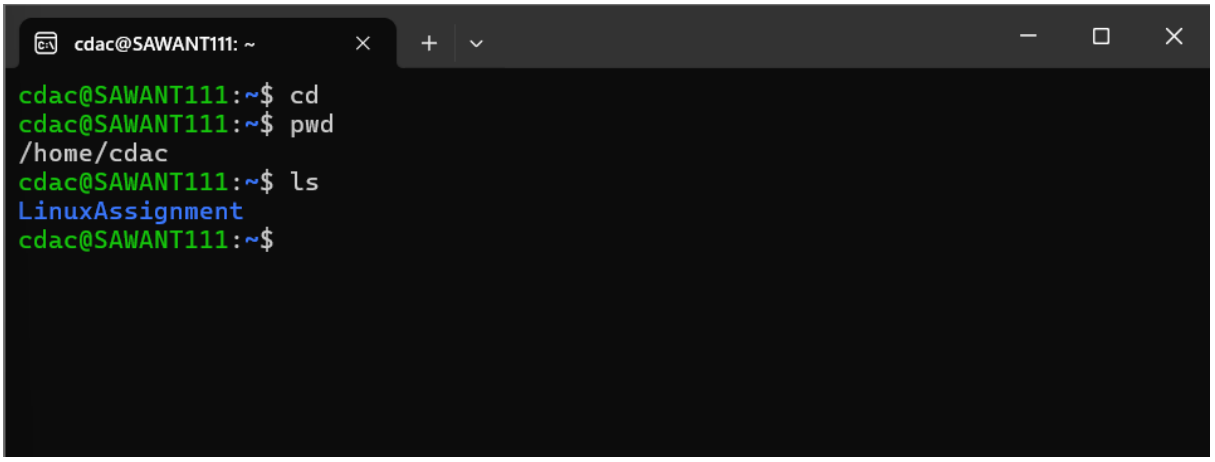
Assignment - 1 COS

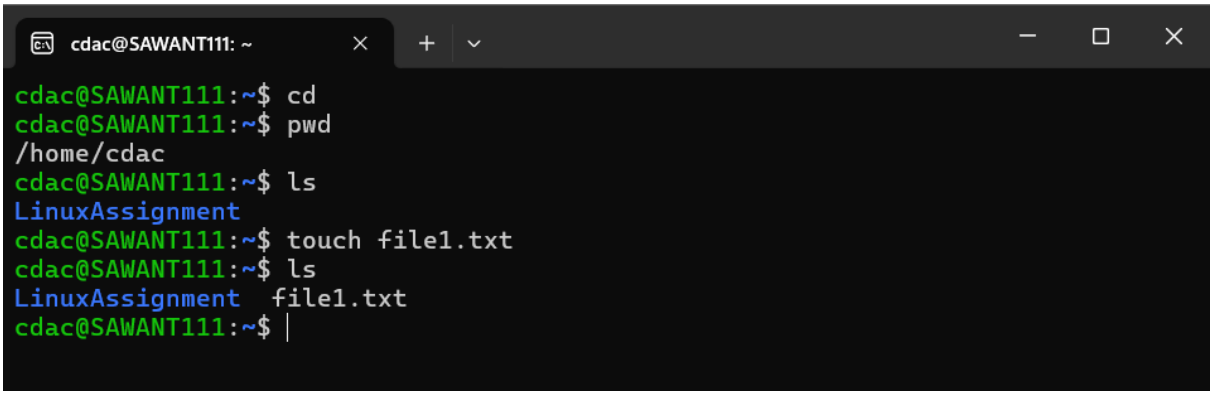
Problem 1 :

a) Navigate and List:
a. Start by navigating to your home directory and list its contents. Then, move into a
directory named "LinuxAssignment" if it exists; otherwise, create it.



To make directory use command `mkdir LinuxAssignment`

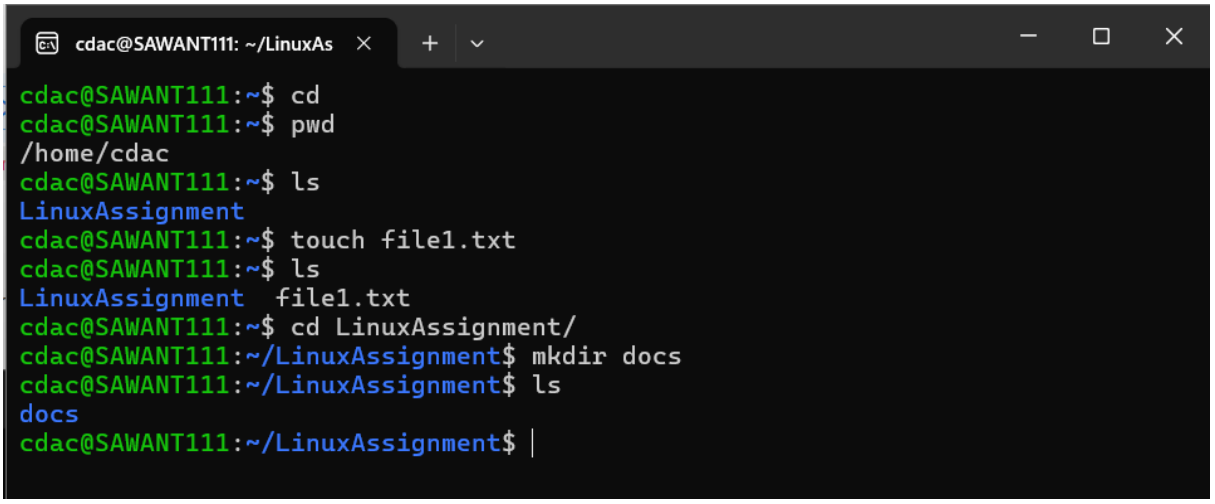
b) File Management:
a. Inside the "LinuxAssignment" directory, create a new file named "file1.txt". Display its
contents.



To create file use `touch file1.txt`

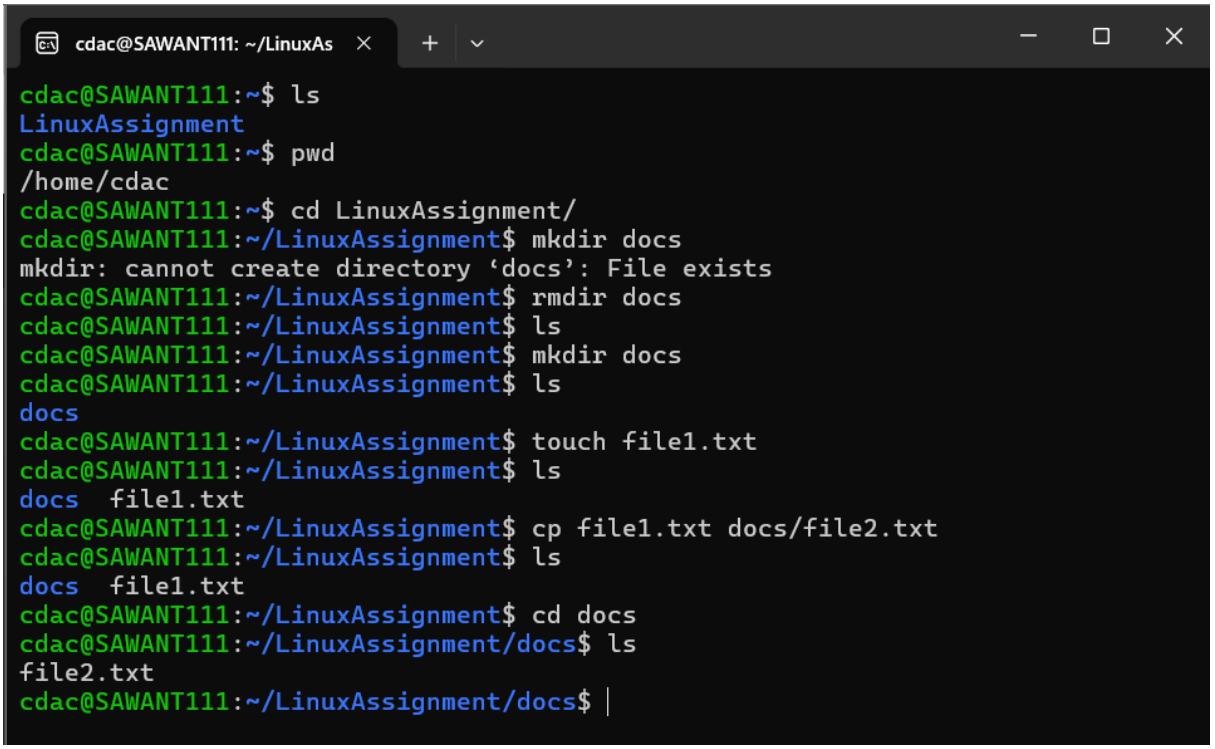
This will create a file named file1.txt.

c) Directory Management:
a. Create a new directory named "docs" inside the "LinuxAssignment" directory.



To go inside LinuxAssignments use `cd LinuxAssignments` command. To make directory use command `mkdir docs`.

d) Copy and Move Files:
a. Copy the "file1.txt" file into the "docs" directory and rename it to "file2.txt".

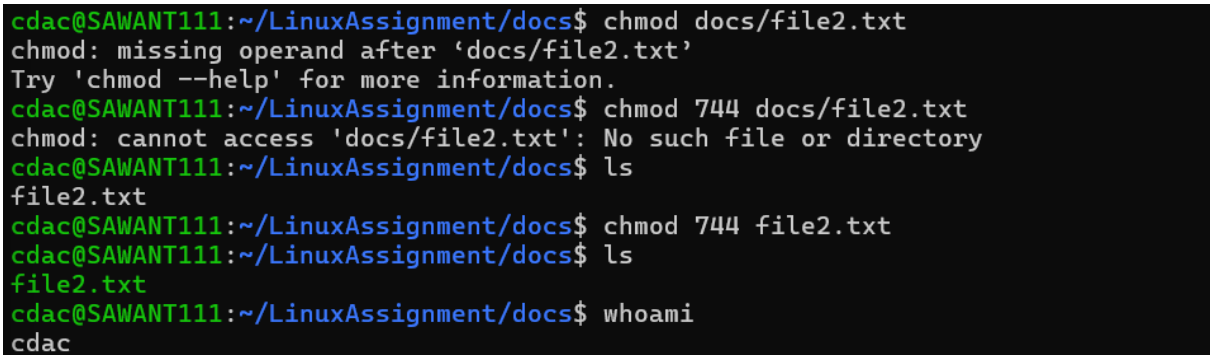


To Copy the "file1.txt" file into the "docs" directory and rename it to "file2.txt" use command

`cp file1.txt docs/file2.txt`.

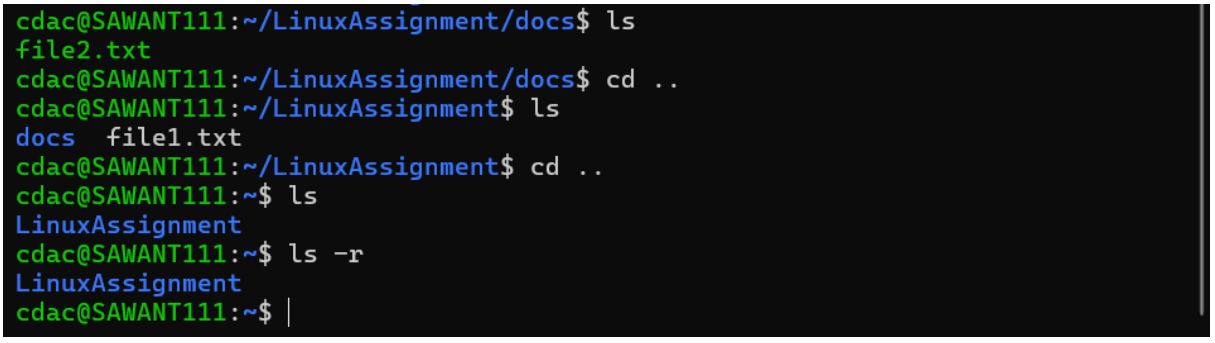
This will copy file1.txt into docs directory and rename it to file2.txt

e) Permissions and Ownership:
a. Change the permissions of "file2.txt" to allow read, write, and execute permissions for
the owner and only read permissions for others. Then, change the owner of "file2.txt" to
the current user.



To Change the permissions of "file2.txt" to allow read, write, and execute permissions for
the owner and only read permissions for others use command `chmod 744 file2.txt`. We can check owner by using `whoami` command.

f) Final Checklist:
a. Finally, list the contents of the "LinuxAssignment" directory and the root directory to
ensure that all operations were performed correctly.



To list the contents use `ls` command.

g) File Searching:
a. Search for all files with the extension ".txt" in the current directory and its subdirectories.
b. Display lines containing a specific word in a file (provide a file name and the specific
word to search).

```
cdac@SAWANT111:~/LinuxAssignment$ nano file1.txt
cdac@SAWANT111:~/LinuxAssignment$ cat file1.txt
Hello
Hii
Good morning
I Love India
Good Evening
I am doing CDAC PG-DAC
My name is Amruta Jadhav Sawant
Hello World

cdac@SAWANT111:~/LinuxAssignment$ find . -name "*.txt"
./docs/file2.txt
./file1.txt
cdac@SAWANT111:~/LinuxAssignment$ grep "India" file1.txt
I Love India
cdac@SAWANT111:~/LinuxAssignment$
```

To Search for all files with the extension ".txt" in the current directory and its subdirectories we can use `find . -name “*.txt”` command.

To Display lines containing a specific word in a file , `grep “specific word” file name` command.

h) System Information:
a. Display the current system date and time.

```
cdac@SAWANT111:~/LinuxAssignment$ date
Wed Aug 28 19:20:54 IST 2024
cdac@SAWANT111:~/LinuxAssignment$
```

By using `date` command we can get current date and time.

i) Networking:
a. Display the IP address of the system.
b. Ping a remote server to check connectivity (provide a remote server address to ping).

```
cdac@SAWANT111:~/LinuxAssignment$ date
Wed Aug 28 19:20:54 IST 2024
cdac@SAWANT111:~/LinuxAssignment$ hostname
SAWANT111
cdac@SAWANT111:~/LinuxAssignment$ hostname -I
172.23.31.66
cdac@SAWANT111:~/LinuxAssignment$ ipconfig
Command 'ipconfig' not found, did you mean:
  command 'ifconfig' from deb net-tools (1.60+git20181103.0eebece-1ubuntu5)
  command 'iconfig' from deb ipmiutil (3.1.8-1)
  command 'iwconfig' from deb wireless-tools (30~pre9-13.1ubuntu4)
Try: sudo apt install <deb name>
cdac@SAWANT111:~/LinuxAssignment$ ping -c 4 amazon.com
PING amazon.com (52.94.236.248) 56(84) bytes of data.
64 bytes from 52.94.236.248 (52.94.236.248): icmp_seq=1 ttl=243 time=313 ms
64 bytes from 52.94.236.248 (52.94.236.248): icmp_seq=2 ttl=243 time=312 ms
64 bytes from 52.94.236.248 (52.94.236.248): icmp_seq=3 ttl=243 time=311 ms
64 bytes from 52.94.236.248: icmp_seq=4 ttl=243 time=311 ms

--- amazon.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4104ms
rtt min/avg/max/mdev = 311.407/311.963/312.547/0.538 ms
cdac@SAWANT111:~/LinuxAssignment$
```

Display the IP address of the system use `ipconfig` or `hostname -I`.

j) File Compression:

a. Compress the "docs" directory into a zip file.

b. Extract the contents of the zip file into a new directory.

```
cdac@SAWANT111:~/LinuxAssignment$ zip -r docs.zip docs
Command 'zip' not found, but can be installed with:
sudo apt install zip
cdac@SAWANT111:~/LinuxAssignment$ sudo apt install zip
[sudo] password for cdac:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  unzip
The following NEW packages will be installed:
  unzip zip
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 350 kB of archives.
After this operation, 929 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 unzip amd64
6.0-26ubuntu3.1 [174 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 zip amd64 3.0-12buil
d2 [176 kB]
Fetched 350 kB in 3s (116 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 24208 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-26ubuntu3.1_amd64.deb ...
Unpacking unzip (6.0-26ubuntu3.1) ...
Selecting previously unselected package zip.
Preparing to unpack .../zip_3.0-12build2_amd64.deb ...
Unpacking zip (3.0-12build2) ...
Setting up unzip (6.0-26ubuntu3.1) ...
Setting up zip (3.0-12build2) ...
Processing triggers for man-db (2.10.2-1) ...
cdac@SAWANT111:~/LinuxAssignment$ zip -r docs.zip docs
  adding: docs/ (stored 0%)
  adding: docs/file2.txt (stored 0%)
cdac@SAWANT111:~/LinuxAssignment$ mkdir newdocs
cdac@SAWANT111:~/LinuxAssignment$ unzip docs.zip -d newdocs
Archive:  docs.zip
   creating: newdocs/docs/
 extracting: newdocs/docs/file2.txt
cdac@SAWANT111:~/LinuxAssignment$
```

Compress the "docs" directory into a zip file use `zip -r docs.zip docs`

To install zip use `sudo apt install zip`

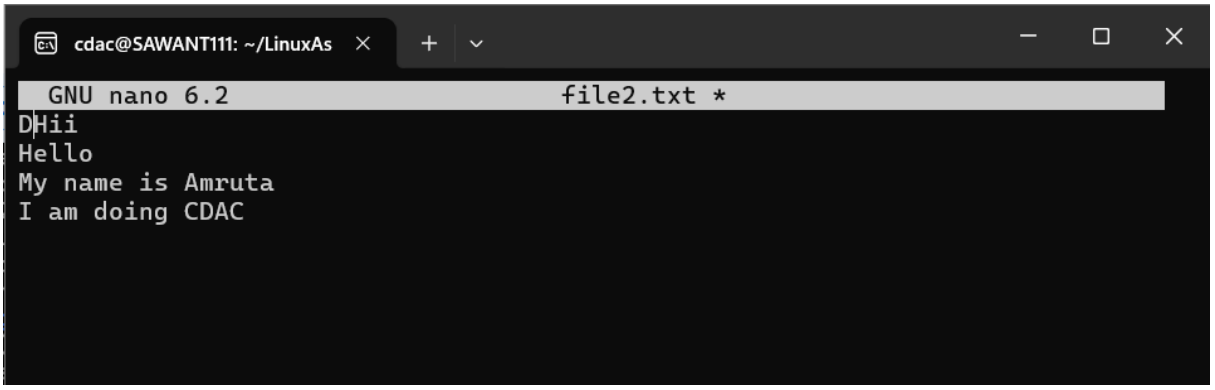
To make new directory `mkdir newdocs`

Extract the contents of the zip file into a new directory `unzip docs.zip -d newdocs`

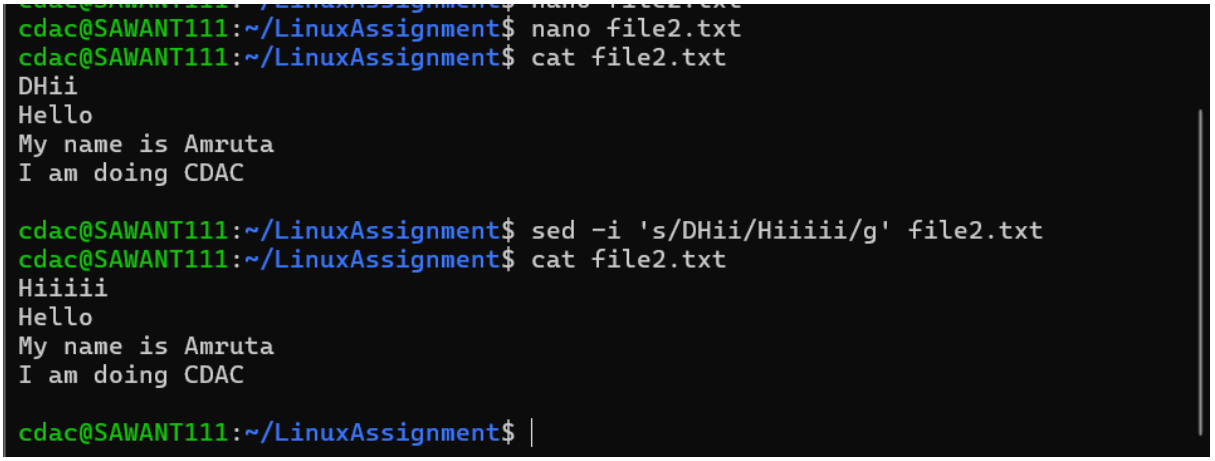
k) File Editing:

a. Open the "file1.txt" file in a text editor and add some text to it.

b. Replace a specific word in the "file1.txt" file with another word (provide the original
word and the word to replace it with).



Above I opened file2.txt using command nano `file2.txt`



Replace a specific word in the "file1.txt" file with another word (provide the original
word and the word to replace it with), use ‘
`sed -i 's/old word/new word/g`' file name.

Problem 2 :

a. Suppose you have a file named "data.txt" containing important information. Display the
first 10 lines of this file to quickly glance at its contents using a command.

```
cdac@SAWANT111:~/LinuxAssignment$ head -n 10 data.txt
Java Tutorial
Our core Java programming tutorial is designed for students and working prof
essionals. Java is an object-oriented, class-based, concurrent, secured and
general-purpose computer-programming language. It is a widely used robust te
chnology.

What is Java?
Java is a programming language and a platform. Java is a high level, robust,
 object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracl
e) in the year 1995. James Gosling is known as the father of Java. Before Ja
va, its name was Oak. Since Oak was already a registered company, so James G
osling and his team changed the name from Oak to Java.

Platform: Any hardware or software environment in which a program runs, is k
nown as a platform. Since Java has a runtime environment (JRE) and API, it i
s called a platform.
Java is a multi-platform, object-oriented, and network-centric language that
 can be used as a platform in itself. It is a fast, secure, reliable program
ming language for coding everything from mobile apps and enterprise software
 to big data applications and server-side.
cdac@SAWANT111:~/LinuxAssignment$ head -n 5 data.txt
Java Tutorial
Our core Java programming tutorial is designed for students and working prof
essionals. Java is an object-oriented, class-based, concurrent, secured and
general-purpose computer-programming language. It is a widely used robust te
chnology.

What is Java?
Java is a programming language and a platform. Java is a high level, robust,
 object-oriented and secure programming language.
```

`head -n 10 data.txt`

b. Now, to check the end of the file for any recent additions, display the last 5 lines of
"data.txt" using another command.

```
cdac@SAWANT111:~/LinuxAssignment$ tail -n 5 data.txt

Java was developed by Sun Microsystems (which is now the subsidiary of Oracl
e) in the year 1995. James Gosling is known as the father of Java. Before Ja
va, its name was Oak. Since Oak was already a registered company, so James G
osling and his team changed the name from Oak to Java.

Platform: Any hardware or software environment in which a program runs, is k
nown as a platform. Since Java has a runtime environment (JRE) and API, it i
s called a platform.
Java is a multi-platform, object-oriented, and network-centric language that
 can be used as a platform in itself. It is a fast, secure, reliable program
ming language for coding everything from mobile apps and enterprise software
 to big data applications and server-side.
cdac@SAWANT111:~/LinuxAssignment$
```

`tail -n 5 data.txt`

c. In a file named "numbers.txt," there are a series of numbers. Display the first 15 lines of
this file to analyze the initial data set.

```
cdac@SAWANT111:~/LinuxAssignment$ la
data.txt  docs  docs.zip  file1.txt  file2.txt  newdocs
cdac@SAWANT111:~/LinuxAssignment$ touch numbers.txt
cdac@SAWANT111:~/LinuxAssignment$ nano numbers.txt
cdac@SAWANT111:~/LinuxAssignment$ head -n 15 numbers.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
cdac@SAWANT111:~/LinuxAssignment$
```

`head -n 15 numbers.txt`

d. To focus on the last few numbers of the dataset, display the last 3 lines of "numbers.txt".

```
cdac@SAWANT111:~/LinuxAssignment$ tail -n 3 numbers.txt
18
19
20
cdac@SAWANT111:~/LinuxAssignment$
```

`tail -n 3 numbers.txt`

e. Imagine you have a file named "input.txt" with text content. Use a command to translate
all lowercase letters to uppercase in "input.txt" and save the modified text in a new file
named "output.txt."

```
cdac@SAWANT111:~/LinuxAssignment$ ls
data.txt  docs.zip   file2.txt  newdocs      output.txt
docs      file1.txt  input.txt  numbers.txt
cdac@SAWANT111:~/LinuxAssignment$ tr 'a-z' 'A-Z' < input.txt > output.txt
cdac@SAWANT111:~/LinuxAssignment$ cat output.txt
TYPES OF JAVA APPLICATIONS
THERE ARE MAINLY 4 TYPES OF APPLICATIONS THAT CAN BE CREATED USING JAVA PROG
RAMMING:

1) STANDALONE APPLICATION
STANDALONE APPLICATIONS ARE ALSO KNOWN AS DESKTOP APPLICATIONS OR WINDOW-BAS
ED APPLICATIONS. THESE ARE TRADITIONAL SOFTWARE THAT WE NEED TO INSTALL ON E
VERY MACHINE. EXAMPLES OF STANDALONE APPLICATION ARE MEDIA PLAYER, ANTIVIRUS
, ETC. AWT AND SWING ARE USED IN JAVA FOR CREATING STANDALONE APPLICATIONS.

2) WEB APPLICATION
AN APPLICATION THAT RUNS ON THE SERVER SIDE AND CREATES A DYNAMIC PAGE IS CA
LLED A WEB APPLICATION. CURRENTLY, SERVLET, JSP, STRUTS, SPRING, HIBERNATE,
JSF, ETC. TECHNOLOGIES ARE USED FOR CREATING WEB APPLICATIONS IN JAVA.3) ENT
ERPRISE APPLICATION
AN APPLICATION THAT IS DISTRIBUTED IN NATURE, SUCH AS BANKING APPLICATIONS,
ETC. IS CALLED AN ENTERPRISE APPLICATION. IT HAS ADVANTAGES LIKE HIGH-LEVEL
SECURITY, LOAD BALANCING, AND CLUSTERING. IN JAVA, EJB IS USED FOR CREATING
ENTERPRISE APPLICATIONS.

4) MOBILE APPLICATION
AN APPLICATION WHICH IS CREATED FOR MOBILE DEVICES IS CALLED A MOBILE APPLIC
ATION. CURRENTLY, ANDROID AND JAVA ME ARE USED FOR CREATING MOBILE APPLICATI
ONS.
cdac@SAWANT111:~/LinuxAssignment$ cat input.txt
Types of Java Applications
There are mainly 4 types of applications that can be created using Java prog
ramming:

1) Standalone Application
Standalone applications are also known as desktop applications or window-bas
ed applications. These are traditional software that we need to install on e
very machine. Examples of standalone application are Media player, antivirus
, etc. AWT and Swing are used in Java for creating standalone applications.

2) Web Application
```

```
tr 'a-z' 'A-Z' < input > output.txt
```

f. In a file named "duplicate.txt," there are several lines of text, some of which are
duplicates. Use a command to display only the unique lines from "duplicate.txt."

```
cdac@SAWANT111:~/LinuxAssignment$ touch duplicate.txt
cdac@SAWANT111:~/LinuxAssignment$ nano duplicate.txt
cdac@SAWANT111:~/LinuxAssignment$ uniq duplicate.txt
Java Tutorial
Our core Java programming tutorial is designed for students and working prof
essionals. Java is an object-oriented, class-based, concurrent, secured and
general-purpose computer-programming language. It is a widely used robust te
chnology.What is Java?
Java is a programming language and a platform. Java is a high level, robust,
 object-oriented and secure programming language.Our core Java programming t
utorial is designed for students and working professionals. Java is an objec
t-oriented, class-based, concurrent, secured and general-purpose computer-pr
ogramming language. It is a widely used robust technology.
cdac@SAWANT111:~/LinuxAssignment$
```

`uniq filename.txt`

g. In a file named "fruit.txt," there is a list of fruits, but some fruits are repeated. Use a
command to display each unique fruit along with the count of its occurrences in "fruit.txt."

```
cdac@SAWANT111:~/LinuxAssignment$ cat fruits.txt
Banana
, Apple,
Strawberry,
Avocado,
Pineapple,
Watermelon,
Mango,
Kiwi,
Orange,
Berry,
Blueberry,
Cherry,
Lemon,
Apricot,
Figs,
Plum,
Papaya,
and
Grapefruit.
Blueberry,
Cherry,
Lemon,
Apricot,
Figs,
Plum,
Papaya,
and
Grapefruit.
cdac@SAWANT111:~/LinuxAssignment$ sort fruits.txt | uniq -c
      1 , Apple,
      2 Apricot,
      1 Avocado,
      1 Banana
      1 Berry,
      2 Blueberry,
      2 Cherry,
      2 Figs,
      2 Grapefruit.
      1 Kiwi,
```

`sort fruits.txt` This commnd sorts the lines in `fruits.txt` alphabetically. Sorting is necessary because `uniq` only detects adjacent duplicates.

`uniq -c` In this command `-c` option tells `uniq` to count the number of occurrences of each unique line. After sorting, `uniq -c` will display each fruit along with its count.