

Objective: Practice Exercises for students organized by experiment.

A. Introduction to Linux Commands

Exercise 1:

1. List all the files and directories in your home directory.

Ans1. ls -a ~

```
[sahilsingh@Sahils-MacBook-Air ~ % ls -a ~  
# Applications  
. Cisco Packet Tracer 8.2.1  
.. Create  
.DS_Store Desktop  
.Trash Documents  
.cache Downloads  
.docker Library  
.git Movies  
.gitignore Music  
.lessht Pictures  
.local Public  
.matplotlib PycharmProjects  
.npm Virtual Machines.localized  
.packettracer a  
.ssh environment  
.viminfo myenv  
.vscode new  
.zprofile node_modules  
.zsh_history package-lock.json  
.zsh_sessions package.json  
.zshrc virtual
```

2. Create a new directory called “test” in your home directory.

Ans2. mkdir test

ls -a~ (just to cross check if your file has been created or not.)

```
sahilsingh@Sahils-MacBook-Air ~ % mkdir test
sahilsingh@Sahils-MacBook-Air ~ % ls -a ~
# Cisco Packet Tracer 8.2.1
. Create
.. Desktop
.DS_Store Documents
.Trash Downloads
.cache Library
.docker Movies
.git Music
.gitignore Pictures
.lesshst Public
.local PycharmProjects
.matplotlib Virtual Machines.localized
.npm a
.packettracer environment
.ssh myenv
.viminfo new
.vscode node_modules
.zprofile package-lock.json
.zsh_history package.json
.zsh_sessions test
```

3. Change into the "test" directory you just created.

Ans3. cd test

```
sahilsingh@Sahils-MacBook-Air ~ % cd test
sahilsingh@Sahils-MacBook-Air test %
```

4. Create a new file called "example.txt" in the "test" directory.

Ans4. touch example.txt

ls (just to see if example.txt is created or not)

```
sahilsingh@Sahils-MacBook-Air ~ % cd test
sahilsingh@Sahils-MacBook-Air test % touch example.txt
sahilsingh@Sahils-MacBook-Air test % ls
example.txt
```

5. Open the "example.txt" file and write "Hello, World!" in it.

Ans5. nano example.txt

```
sahilsingh@Sahils-MacBook-Air test % nano example.txt
```

6. Save and exit the file.

Ans. “Ctrl+x” then “Shift+y” then “Enter”

```
Hello, World!
```

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
```

```
Y Yes  
^C Cancel N No
```

7. List all the files in the “text” directory.

Ans7. ls

```
[sahilsingh@Sahils-MacBook-Air test % ls  
example.txt  
sahilsingh@Sahils-MacBook-Air test %
```

8. Rename the “example.txt” file to “sample.txt.”

Ans8. mv example.txt sample.txt

```
[sahilsingh@Sahils-MacBook-Air test % ls
example.txt
[sahilsingh@Sahils-MacBook-Air test % mv example.txt sample.txt
```

9. List all the files in the “test” directory again to verify the file has been renamed.

Ans9. ls

```
[sahilsingh@Sahils-MacBook-Air test % ls
example.txt
[sahilsingh@Sahils-MacBook-Air test % mv example.txt sample.txt
[sahilsingh@Sahils-MacBook-Air test % ls
sample.txt
```

10. Remove the “test” directory and all its contents.

Ans10. rm -r test

```
sahilsingh@Sahils-MacBook-Air ~ % rm -r test
sahilsingh@Sahils-MacBook-Air ~ % ls
#                Public
Applications     PycharmProjects
Cisco Packet Tracer 8.2.1  Virtual Machines.localized
Create           a
Desktop          environment
Documents        myenv
Downloads        new
Library          node_modules
Movies           package-lock.json
Music            package.json
Pictures         virtual
```

11. List all the files and directories in the root directory

Ans11. ls -a /

```
sahilsingh@Sahils-MacBook-Air ~ % ls -a /  
  
.  
..  
.VolumeIcon.icns  
.file  
.vol  
Applications  
Library  
  
System  
Users  
Volumes  
bin  
cores  
dev  
etc  
  
home  
opt  
private  
sbin  
tmp  
usr  
var
```

12. Change into the “/etc” directory.

Ans12. Cd /etc

```
sahilsingh@Sahils-MacBook-Air ~ % cd /etc  
sahilsingh@Sahils-MacBook-Air /etc %
```

13. List all the files and directories in the “/etc” directory.

Ans13. ls

```
sahilsingh@Sahils-MacBook-Air ~ % cd /etc  
sahilsingh@Sahils-MacBook-Air /etc % ls  
afpovertcp.cfg  
aliases  
aliases.db  
apache2  
as1  
as1.conf  
auto_home  
auto_master  
autofs.conf  
bashrc  
bashrc_Apple_Terminal  
com.apple.screensharing.agent.launchd  
csh.cshrc  
csh.login  
csh.logout  
cups  
defaults  
find.codes  
ftpusers  
gettytab  
group  
hosts  
hosts.equiv  
irbrc  
kern_loader.conf  
krb5.keytab  
localtime  
locate.rc  
mail.rc  
man.conf  
manpaths  
manpaths.d  
master.passwd  
networks  
newsyslog.conf  
newsyslog.d  
nfs.conf  
notify.conf  
ntp.conf  
ntp_opendirectory.conf  
openldap  
pam.d  
passwd  
paths  
paths.d  
periodic  
pf.anchors  
pf.conf  
pf.os  
postfix  
ppp  
profile  
protocols  
racoon  
rc.common  
rc.netboot  
resolv.conf  
rmtab  
rpc  
rtadvd.conf  
security  
services  
shells  
snmp  
ssh  
ssl  
sudo_lecture  
sudoers  
sudoers.d  
syslog.conf  
ttsys  
uucp  
wfs  
xtab  
zprofile  
zshrc  
zshrc_Apple_Terminal
```

Exercise 2:

1. Create a new directory called “lab_exercises” in your home directory.

Ans1. mkdir lab_exercises

```
[sahilsingh@Sahils-MacBook-Air ~ % mkdir lab_exercises]
[sahilsingh@Sahils-MacBook-Air ~ % ls]
#                               PycharmProjects
Applications                    Virtual Machines.localized
Cisco Packet Tracer 8.2.1       a
Create                          environment
Desktop                         lab_exercises
Documents                      myenv
Downloads                      new
Library                        node_modules
Movies                        package-lock.json
Music                          package.json
Pictures                      virtual
Public
```

2. Inside the “lab_exercises” directory, create a new file called “file1.txt” and write some text in it.

Ans2.cd lab_exercises

touch file1.txt

nano file1.txt(Use this to add some text in it.)

```
[sahilsingh@Sahils-MacBook-Air ~ % cd lab_exercises]
[sahilsingh@Sahils-MacBook-Air lab_exercises % touch file1.txt]
[sahilsingh@Sahils-MacBook-Air lab_exercises % ls]
file1.txt
```

3. Create a copy of “file1.txt” and name it “file2.txt” using the cp command.

Ans3. Cp file1.txt file2.txt

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % cp file1.txt file2.txt  
[sahilsingh@Sahils-MacBook-Air lab_exercises % ls  
file1.txt      file2.txt
```

4. Verify that “file2.txt” is an exact copy of “file1.txt” by opening both files and comparing their contents.

Ans4. Cat file1.txt file2.txt

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % cat file1.txt file2.txt  
Hello, this is 1st file created in lab_exercises.  
Hello, this is 1st file created in lab_exercises.
```

5. Create a new directory called “backup” inside the ”lab_exercises” directory.

Ans5. mkdir backup

ls (just to see that new directory has been created)

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % mkdir backup  
[sahilsingh@Sahils-MacBook-Air lab_exercises % ls  
backup      file1.txt      file2.txt
```

6. Move “file1.txt” and ”file2.txt” to the “backup” directory using the mv command.

Ans6. mv file1.txt file2.txt backup

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % mv file1.txt file2.txt backup
```


7. Verify that both files have been moved to the “backup” directory by listing its contents.

Ans. Cd backup (To change the directory)

ls

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % cd backup
[sahilsingh@Sahils-MacBook-Air backup % ls
file1.txt      file2.txt
_
```

8. Create a new file called “file3.txt” in the “lab_exercises” directory and write some text in it.

Ans. cd lab_exercises

touch file3.txt

nano file3.txt (To write some text in it)

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % touch file3.txt
[sahilsingh@Sahils-MacBook-Air lab_exercises % nano file3.txt
```

9. Create a new directory “archive” inside the “lab_exercise” directory.

Ans. mkdir archive

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % mkdir archive
[sahilsingh@Sahils-MacBook-Air lab_exercises % ls
archive      backup      file3.txt
_
```

10. Move “file3.txt” to the “archive” directory and rename it as “file3_backup.txt” using the mv command.

Ans. mv file3.txt archive

cd archive

ls

mv file3.txt

file3_backup.txt

ls

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % mv file3.txt archive
[sahilsingh@Sahils-MacBook-Air lab_exercises % cd archive
[sahilsingh@Sahils-MacBook-Air archive % ls
file3.txt
[sahilsingh@Sahils-MacBook-Air archive % mv file3.txt file3_backup.txt
[sahilsingh@Sahils-MacBook-Air archive % ls
file3_backup.txt
_
```

11. Verify that “file3_backup.txt” has been moved to the “archive” directory and that its contents are the same as “file3.txt”.

Ans. We have verified that it has been moved to archive directory by using the above commands only.

12. Create a new directory “temp” inside the “lab_exercises” directory.

Ans. Cd lab_exercises

mkdir temp

```
[sahilsingh@Sahils-MacBook-Air ~ % cd lab_exercises
[sahilsingh@Sahils-MacBook-Air lab_exercises % mkdir temp
_
```

13. Create a new file called "file4.txt" in the "temp" directory and write some text in it.

Ans. Cd temp

touch file4.txt

nano file4.txt

```
sahilsingh@Sahils-MacBook-Air lab_exercises % cd temp
sahilsingh@Sahils-MacBook-Air temp % touch file4.txt
sahilsingh@Sahils-MacBook-Air temp % nano file4.txt
```

14. Move "file4.txt" to the "lab_exercises" directory using the mv command.

Ans. Mv file4.txt lab_exercises

```
[sahilsingh@Sahils-MacBook-Air temp % nano file4.txt
[sahilsingh@Sahils-MacBook-Air temp % mv file4.txt lab_exercises
```

15. Verify that "file4.txt" has been moved to the "lab_exercises" directory and that its contents are the same as before.

Ans. ls

```
[sahilsingh@Sahils-MacBook-Air lab_exercises % ls
archive      backup      file4.txt   temp
[sahilsingh@Sahils-MacBook-Air lab_exercises % cat file4.txt
Hello, this file is file4 created in temp directory
```

B. Shell Programming

Exercise 1:

A. Write a script that displays “Hello, World!” when executed.

Ans.

Step 1:- touch hello_world.sh

Step 2:- nano hello_world.sh

Step 3:- chmod +x hello_world.sh

Step 4:- ./hello_world.sh

```
sahilsingh@Sahils-MacBook-Air desktop % touch hello_world.sh
sahilsingh@Sahils-MacBook-Air desktop % nano hello_world.sh
sahilsingh@Sahils-MacBook-Air desktop % chmod +x hello_world.sh
sahilsingh@Sahils-MacBook-Air desktop % ./hello_world.sh
Hello, World!
```

B. Modify the script to accept a command line argument and display "Hello, <argument>!" instead of "Hello, World!".

Ans.

Step 1:- nano hello_world.sh

Step 2:- chmod +x hello_world.sh

Step 3:- ./hello_world.sh Sahil

```
-----
sahilsingh@Sahils-MacBook-Air desktop % nano hello_world.sh
sahilsingh@Sahils-MacBook-Air desktop % chmod +x hello_world.sh
sahilsingh@Sahils-MacBook-Air desktop % ./hello_world.sh Sahil
Hello, Sahil!
```

Exercise 2: -

Write a script that accepts two command line arguments and display their sum.

Ans.

Step 1:- touch sum.h

Step 2:- nano sum.h

Step 3:- chmod +x sum.h

Step 4:- ./sum.h <value1> <value2>

Code: -

```
#!/bin/bash
```

```
if [ $# -ne 2 ]; then
    echo "Usage: $0 <num1> <num2>"
else
    num1=$1
    num2=$2
    sum=$((num1 + num2))
    echo "Sum: $sum"
fi
```

```
[sahilsingh@Sahils-MacBook-Air desktop % nano sum.h
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x sum.h
[sahilsingh@Sahils-MacBook-Air desktop % ./sum.h 45 76
Sum: 121
```

Exercise 3: -

A. Write a script that accepts a directory name as a command line argument and displays the number of files in that directory.

Ans.

Step 1:- touch countfiles.sh

Step 2:- nano countfiles.sh

Step 3:- chmod +x countfiles.sh

Step 4:- ./countfiles.sh /path/to/your/directory

Code:-

```
#!/bin/bash

if [ $# -ne 1 ]; then
    echo "Usage: $0 <directory>"
    exit 1
fi
directory="$1"
if [ ! -d "$directory" ]; then
    echo "$directory is not a valid directory."
    exit 1
fi
file_count=$(ls -A "$directory" | wc -l)
echo "Number of files in $directory: $file_count"
```

```
[sahilsingh@Sahils-MacBook-Air desktop % touch countfiles.sh
[sahilsingh@Sahils-MacBook-Air desktop % nano countfiles.sh
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x countfiles.sh
[sahilsingh@Sahils-MacBook-Air desktop % pwd
/Users/sahilsingh/desktop
sahilsingh@Sahils-MacBook-Air desktop % ./countfiles.sh /Users/sahilsingh/desktop
Number of files in /Users/sahilsingh/desktop:      11
```

B. Modify the script to display the number of files in the directory and all its subdirectories.

Ans.

Step 1:- nano countfiles.sh

Step 2:- chmod +x countfiles.sh

Step 3:- ./countfiles.sh /path/to/your/directory

Code: -

```
#!/bin/bash
```

```
if [ $# -ne 1 ]; then
    echo "Usage: $0 <directory>"
    exit 1
fi
```

```
directory="$1"
```

```
if [ ! -d "$directory" ]; then
    echo "$directory is not a valid directory."
    exit 1
fi
```

```
file_count=$(find "$directory" -type f | wc -l)
```

```
echo "Number of files in $directory and its subdirectories: $file_count"
```

```
[sahilsingh@Sahils-MacBook-Air desktop % nano countfiles.sh ]
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x countfiles.sh ]
sahilsingh@Sahils-MacBook-Air desktop % ./countfiles.sh /Users/sahilsingh/deskto
p
Number of files in /Users/sahilsingh/desktop and its subdirectories:      27
```

Exercise 4: -

A. Write a script that accepts a filename as a command line argument and displays the number of lines, word and characters in that file.

Ans.

Step 1:- touch countfiles_stats.sh

Step 2:- nano countfiles_stats.sh

Step 3:- chmod +x countfiles_stats.sh

Step 4:- ./countfiles_stats.sh. /path/to/your/.txt/file

Code:-

```
#!/bin/bash
```

```
if [ $# -ne 1 ]; then
    echo "Usage: $0 <filename>"
    exit 1
fi

filename="$1"

if [ ! -f "$filename" ]; then
    echo "$filename is not a valid file."
    exit 1
fi
```

```
line_count=$(wc -l < "$filename")
word_count=$(wc -w < "$filename")
char_count=$(wc -c < "$filename")
```

```
echo "Number of lines: $line_count"
echo "Number of words: $word_count"
echo "Number of characters: $char_count"
```

```
[sahilsingh@Sahils-MacBook-Air desktop % touch abc.txt
[sahilsingh@Sahils-MacBook-Air desktop % nano abc.txt
[sahilsingh@Sahils-MacBook-Air desktop % touch countfiles_stats.sh
[sahilsingh@Sahils-MacBook-Air desktop % nano countfiles_stats.sh
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x countfiles_stats.sh
[sahilsingh@Sahils-MacBook-Air desktop % ./countfiles_stats.sh abc.txt
Number of lines:          2
Number of words:         18
Number of characters:     89
```


B. Modify the script to accept multiple file names as a command line arguments and display the number of lines, words, and characters in each file.

Ans.

Step 1:- nano countfiles.sh

Step 2:- chmod +x count files.sh

Step 3:- ./countfiles.sh file1.txt file2.txt file3.txt

Code:-

```
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Usage: $0 <file1> [<file2> <file3> ...]"
    exit 1
fi

for filename in "$@"; do
    if [ ! -f "$filename" ]; then
        echo "$filename is not a valid file."
    else
        line_count=$(wc -l < "$filename")
        word_count=$(wc -w < "$filename")
        char_count=$(wc -c < "$filename")

        echo "File: $filename"
        echo "Number of lines: $line_count"
        echo "Number of words: $word_count"
        echo "Number of characters: $char_count"
        echo
    fi
done
```

```
[sahilsingh@Sahils-MacBook-Air desktop % touch def.txt
[sahilsingh@Sahils-MacBook-Air desktop % nano def.txt
[sahilsingh@Sahils-MacBook-Air desktop % nano countfiles_stats.sh
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x countfiles_stats.sh
[sahilsingh@Sahils-MacBook-Air desktop % ./countfiles_stats.sh abc.txt def.txt
File: abc.txt
Number of lines:      2
Number of words:     18
Number of characters:      89

File: def.txt
Number of lines:      7
Number of words:     12
Number of characters:      67
```

Exercise 5: -

A. Declare a variable called “name” and assign your name to it. Display the value of variable using the echo command.

Ans.

```
name="Sahil"
echo
$name
```

```
[sahilsingh@Sahils-MacBook-Air desktop % name="Sahil"
[sahilsingh@Sahils-MacBook-Air desktop % echo $name
Sahil
```

B. Declare a variable “age” and assign your age to it. Display the value of the variable using the echo command.

Ans. Age="19"

```
echo $age
```

```
[sahilsingh@Sahils-MacBook-Air desktop % age="19"
[sahilsingh@Sahils-MacBook-Air desktop % echo $age
19
```

C. Declare a variable “color” and assign your favorite color to it. Display the value of the variable using the echo command.

Ans. color="Black"

```
echo $color
```

```
[sahilsingh@Sahils-MacBook-Air desktop % color="Black"
[sahilsingh@Sahils-MacBook-Air desktop % echo $color
Black
```

Exercise 6: -

Declare a variable "num1" and assign value 10 to it. Declare a second variable "num2" and assign the value 5 to it. Add the values of the two variables and display the result using echo command.

Ans. num1=10

num2=5

Let "z = \$num1 + \$num2"

echo \$z

```
[sahilsingh@Sahils-MacBook-Air desktop % num1=10
[sahilsingh@Sahils-MacBook-Air desktop % num2=5
[sahilsingh@Sahils-MacBook-Air desktop % let "z = $num1 + $num2"
[sahilsingh@Sahils-MacBook-Air desktop % echo $z
15
```

Exercise 7: -

A. Declare a variable called “filename” and assign the value “sample.txt” to it. Use the variable to create a new file with that name using the touch command.

Ans. filename="sample.txt"

touch "\$filename"

```
[sahilsingh@Sahils-MacBook-Air desktop % filename="sample.txt"]
[sahilsingh@Sahils-MacBook-Air desktop % touch "$filename"]
[sahilsingh@Sahils-MacBook-Air desktop % ls]
OS LAB FILES                               Screenshot 2023-08-31 at 1.07.36 AM.png
Screenshot 2023-08-31 at 1.03.48 AM.png    Screenshot 2023-08-31 at 1.12.24 AM.png
Screenshot 2023-08-31 at 1.05.36 AM.png    sample.txt
```

B. Declare a variable called “directory” and assign the value “myfolder” to it. Use the variable to create a new directory with that name using the mkdir command.

Ans.

directory="myfolder"

mkdir "\$directory"

```
[sahilsingh@Sahils-MacBook-Air desktop % directory="myfolder"]
[sahilsingh@Sahils-MacBook-Air desktop % mkdir "$directory"]
[sahilsingh@Sahils-MacBook-Air desktop % ls]
OS LAB FILES                               Screenshot 2023-08-31 at 1.12.24 AM.png
Screenshot 2023-08-31 at 1.03.48 AM.png    Screenshot 2023-08-31 at 1.15.37 AM.png
Screenshot 2023-08-31 at 1.05.36 AM.png    myfolder
Screenshot 2023-08-31 at 1.07.36 AM.png    sample.txt
```

Exercise 8: -

Declare a variable called “files” and assign a list of filenames to it. Use a loop to display the contents of each file in the list using the cat command.

Ans. touch file1.txt file2.txt file3.txt

touch display_files.sh

nano display_files.sh

chmod +x display_files.sh

./display_files.sh

cat display_files.sh

Code: -

```
files=("file1.txt" "file2.txt" "file3.txt")
```

```
for file in "${files[@]}; do
```

```
    echo "Contents of $file:"
```

```
    cat "$file"
```

```
    echo "_____"
```

```
done
```

```
sahilsingh@Sahils-MacBook-Air desktop % touch file1.txt file2.txt file3.txt
```

```
sahilsingh@Sahils-MacBook-Air desktop % chmod +x display_files.sh
```

```
sahilsingh@Sahils-MacBook-Air desktop % ./display_files.sh
```

```
Contents of file1.txt:
```

```
-----
```

```
Contents of file2.txt:
```

```
-----
```

```
Contents of file3.txt:
```

```
-----
```

```
sahilsingh@Sahils-MacBook-Air desktop % cat display_files.sh
```

```
files=("file1.txt" "file2.txt" "file3.txt")
```

```
for file in "${files[@]}; do
```

```
    echo "Contents of $file:"
```

```
    cat "$file"
```

```
    echo "-----"
```

```
done
```

Exercise 9: -

A. Write a command that displays the contents of a file called “file1.txt” on the screen.

Ans. touch file1.txt

nano file1.txt

cat file1.txt

```
[sahilsingh@Sahils-MacBook-Air desktop % cat file1.txt
```

```
Hello, it is a normal txt
```

```
file which is made using touch command. And it's content will be shown  
using cat command.
```

—

B. Use input redirection to create a new file called “file2.txt” with the contents of “file1.txt.”

Ans. touch file2.txt

cat < file1.txt > file2.txt

```
[sahilsingh@Sahils-MacBook-Air desktop % nano file1.txt
```

```
[sahilsingh@Sahils-MacBook-Air desktop % cat file1.txt
```

```
Hello, it is a normal txt
```

```
file which is made using touch command. And it's content will be shown  
using cat command.
```

```
[sahilsingh@Sahils-MacBook-Air desktop % cat < file1.txt > file2.txt
```

```
[sahilsingh@Sahils-MacBook-Air desktop % cat file2.txt
```

```
Hello, it is a normal txt
```

```
file which is made using touch command. And it's content will be shown  
using cat command.
```

C. Write a command that appends the contents of “file1.txt” to the end of “file2.txt”

Ans. cat file1.txt >> file2.txt

```
[sahilsingh@Sahils-MacBook-Air desktop % cat file1.txt >> file2.txt
[sahilsingh@Sahils-MacBook-Air desktop % cat file2.txt
Hello, it is a normal txt
file which is made using touch command. And it's content will be shown
using cat command.
Hello, it is a normal txt
file which is made using touch command. And it's content will be shown
using cat command.
```

Exercise 10: -

A. Write a for loop that prints the numbers from 1 to 10 on the screen.

Ans.

Step 1:- touch for_loop.sh

Step 2:- nano for_loop.sh

Step 3:- chmod +x for_loop.sh

Step 4:- ./for_loop.sh

Code: -

```
for i in {1..10}
do
    echo $i
done
```

```
[sahilsingh@Sahils-MacBook-Air desktop % touch for_loop.sh
[sahilsingh@Sahils-MacBook-Air desktop % nano for_loop.sh
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x for_loop.sh
[sahilsingh@Sahils-MacBook-Air desktop % ./for_loop.sh
1
2
3
4
5
6
7
8
9
10
```


B. Modify the loop to print only the even number from 1 to 10.

Ans. nano for_loop.sh

Code: -

```
for i in {1..10}
do
    if [  $((i \% 2)) -eq 0$  ]
    then
        echo $i
    fi
done
```

```
[sahilsingh@Sahils-MacBook-Air desktop % nano for_loop.sh
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x for_loop.sh
[sahilsingh@Sahils-MacBook-Air desktop % ./for_loop.sh
2
4
6
8
10
```

Exercise 11: -

A. Write a loop that displays the names of all the files in the current directory.

Ans.

Step 1:- touch files.sh

Step 2:- nano files.sh

Step 3:- chmod +x files.sh

Step 4:- ./files.sh

Code:-

```
for file in *  
do  
    if [ -f "$file" ]  
    then  
        echo "$file"  
    fi  
done
```

```
sahilsingh@Sahils-MacBook-Air ~ % touch files.sh  
sahilsingh@Sahils-MacBook-Air ~ % nano files.sh  
sahilsingh@Sahils-MacBook-Air ~ % chmod +x files.sh  
sahilsingh@Sahils-MacBook-Air ~ % ./files.sh  
files.sh  
package-lock.json  
package.json
```

Note: - The above code will only show the files present in the working directory, not the directory or sub-directory present in it.

B. Modify the loop to display only the names of the files with the extension “.txt”.

Ans.

Step 1:- nano files.sh

Step 2:- chmod +x files.sh

Step 3:- ./files.sh

Code: -

for file in *

do

if [-f "\$file"] && [["\$file" == *.txt]]

then

echo "\$file"

fi

done

```
sahilsingh@Sahils-MacBook-Air ~ % nano files.sh
sahilsingh@Sahils-MacBook-Air ~ % chmod +x files.sh
sahilsingh@Sahils-MacBook-Air ~ % ./files.sh
```

Note: - It is showing blank because I don't have any file with extension of “.txt” in my present working directory. If you have any file with this extension then this will show the name of the file.

Exercise 12: -

- A. Write a case/sac statement that displays a message on the screen based on the value of a variable called "day". If the value is "Monday", the message should be "It's the start of the week". If the value is "Friday", the message should be "Thank goodness it's Friday". For any other value, the message should be "Just another day".**

Ans.

Step 1:- touch case.sh

Step 2:- nano case.sh

Step 3:- chmod +x case.sh

Step 4:- ./case.sh

Code: -

```
#!/bin/bash
```

```
day="Monday" # You can set the value of the "day" variable here
```

```
case "$day" in
  "Monday")
    echo "It's the start of the week"
    ;;
  "Friday")
    echo "Thank goodness it's Friday"
    ;;
  *)
    echo "Just another day"
    ;;
esac
```

```
sahilsingh@Sahils-MacBook-Air desktop % touch case.sh
sahilsingh@Sahils-MacBook-Air desktop % nano case.sh
sahilsingh@Sahils-MacBook-Air desktop % chmod +x case.sh
sahilsingh@Sahils-MacBook-Air desktop % ./case.sh
It's the start of the week
```

B. Modify the case/esac statement to use a read command to read the value of “day” from the user.

Ans.

Step 1:- nano case.sh

Step 2:- chmod +x case.sh

Step 3:- ./case.sh

Code: -

```
#!/bin/bash
```

```
read -p "Enter a day of the week: " day
```

```
case "$day" in
```

```
  "Monday")
```

```
    echo "It's the start of the week"
```

```
    ;;
```

```
  "Friday")
```

```
    echo "Thank goodness it's Friday"
```

```
    ;;
```

```
  *)
```

```
    echo "Just another day"
```

```
    ;;
```

```
esac
```

```
[sahilsingh@Sahils-MacBook-Air desktop % nano case.sh
```

```
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x case.sh
```

```
[sahilsingh@Sahils-MacBook-Air desktop % ./case.sh
```

```
Enter a day of the week: friday
```

```
Just another day
```

```
[sahilsingh@Sahils-MacBook-Air desktop % ./case.sh
```

```
Enter a day of the week: tuesday
```

```
Just another day
```

```
[sahilsingh@Sahils-MacBook-Air desktop % ./case.sh
```

```
Enter a day of the week: monday
```

```
Just another day
```

```
[sahilsingh@Sahils-MacBook-Air desktop % ./case.sh
```

```
Enter a day of the week: Monday
```

```
It's the start of the week
```

—

Exercise 13: -

Write a case/esac statement that calculates the area of a geometric shape based on the users input. If the input is "square", the statement should ask the user for the length of the side and display the area. If the input is "rectangle", the statement should ask the user for the length and width and display the area. If the input is "circle", the statement should ask the user.

Ans.

Step 1:- touch area.sh

Step 2:- nano area.sh

Step 3:- chmod +x area.sh

Step 4:- ./area.sh

```
sahilsingh@Sahils-MacBook-Air desktop % touch area.sh
sahilsingh@Sahils-MacBook-Air desktop % nano area.sh
sahilsingh@Sahils-MacBook-Air desktop % chmod +x area.sh
sahilsingh@Sahils-MacBook-Air desktop % ./area.sh
Enter a geometric shape (square, rectangle, or circle): sqyare
Invalid input. Please enter 'square', 'rectangle', or 'circle'.
sahilsingh@Sahils-MacBook-Air desktop % ./area.sh
Enter a geometric shape (square, rectangle, or circle): square
Enter the length of the side: 32
The area of the square is 1024 square units.
```

Code:-

```
#!/bin/bash
```

```
# Prompt the user to enter the shape
```

```
read -p "Enter a geometric shape (square, rectangle, or circle): " shape
```

```
case "$shape" in
```

```
"square")
```

```
    read -p "Enter the length of the side: " side
```

```
    area=$(echo "$side * $side" | bc)
```

```
    echo "The area of the square is $area square units."
```

```
;;
```

```
"rectangle")
```

```
    read -p "Enter the length: " length
```

```
    read -p "Enter the width: " width
```

```
    area=$(echo "$length * $width" | bc)
```

```
    echo "The area of the rectangle is $area square units."
```

```
;;
```

```
"circle")
```

```
    read -p "Enter the radius: " radius
```

```
    pi=3.14159265359
```

```
    area=$(echo "$pi * $radius * $radius" | bc)
```

```
    echo "The area of the circle is $area square units."
```

```
;;
```

```
*)
```

```
    echo "Invalid input. Please enter 'square', 'rectangle', or 'circle'."
```

```
;;
```

```
esac
```

Exercise 14: -

A. Write an if statement that checks if a variable called “x” is greater than 10. If it is, display the message “x is greater than 10”.

Ans.

Step 1:- touch greater.sh

Step 2:- nano greater.sh

Step 3:- chmod +x greater.sh

Step 4:- ./greater.sh

Code:-

```
#!/bin/bash
```

```
x=15
```

```
if [ "$x" -gt 10 ]; then
```

```
    echo "x is greater than 10"
```

```
fi
```

```
[sahilsingh@Sahils-MacBook-Air desktop % touch greater.sh
```

```
[sahilsingh@Sahils-MacBook-Air desktop % nano greater.sh
```

```
[sahilsingh@Sahils-MacBook-Air desktop % chmod +x greater.sh
```

```
[sahilsingh@Sahils-MacBook-Air desktop % ./greater.sh
```

```
x is greater than 10
```

B. Modify the if statement to check if “x” is equal to 10 as well. If it is, display the message “x is equal to 10”.

Ans.

Step 1:- nano greater.sh

Step 2:- chmod +x greater.sh

Step 3:- ./greater.sh

Code: -

```
#!/bin/bash
```

```
x=10
```

```
if [ "$x" -gt 10 ]; then
    echo "x is greater than 10"
elif [ "$x" -eq 10 ]; then
    echo "x is equal to 10"
fi
```

```
sahilsingh@Sahils-MacBook-Air desktop % nano greater.sh
```

```
sahilsingh@Sahils-MacBook-Air desktop % chmod +x greater.sh
```

```
sahilsingh@Sahils-MacBook-Air desktop % ./greater.sh
```

```
x is equal to 10
```


C. File Manipulation Using System Calls

Exercise 1: -

Write a program in C language that creates a file “output.txt”, writes the text “Hello, World!” to it, and then closes the file .

Ans.

Step 1: - nano 1.c

Step 2: - Write the code.

Step 3: - gcc 1.c

Step 4: - ./a.out

Step 5: - cat output.txt

Code: -

Output of the program

```
GNU nano 7.2 1.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>

int main() {
    int fd;
    const char *filename = "output.txt";
    const char *text = "Hello, world!";
    fd = open(filename, O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);
    if (fd == -1) {
        perror("Error opening the file");
        return 1;
    }
    ssize_t bytes_written = write(fd, text, strlen(text));
    if (bytes_written == -1) {
        perror("Error writing to the file"); close(fd);
        return 1;
    }
    close(fd);
    printf("Text successfully written to %s\n", filename);
    return 0;
}
```

```
(shreygrg@Linux23) - [~/Shrey_Garg]
$ gcc 1.c

(shreygrg@Linux23) - [~/Shrey_Garg]
$ ./a.out
Text successfully written to output.txt

(shreygrg@Linux23) - [~/Shrey_Garg]
$ cat output.txt
Hello, world!
```

Exercise 2: -

1. Write a program in C that reads the content of file "input.txt" and writes them to a new file "output.txt". You should use system calls like open(),read() and write().

Ans.

Step 1: - cat input.txt

Step 2: - nano 2.c

Step 3: - write the code

Step 4: - gcc 2.c

Step 5: - ./a.out

Step 6: - cat output.txt

Code: -

```
GNU nano 7.2 2.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main() {
    int input_fd, output_fd;
    const char *input_filename="input.txt";
    const char *output_filename = "output.txt";
    char buffer[4096];

    input_fd = open(input_filename, O_RDONLY);

    if (input_fd == -1) {
        perror("Error opening the input file");
        return 1;
    }

    output_fd = open(output_filename, O_WRONLY | O_CREAT | O_TRUNC, 0666);

    if (output_fd == -1) {
        perror("Error opening/creating the output file");
        close(input_fd);
        return 1;
    }

    ssize_t bytes_read;
    while ((bytes_read = read(input_fd, buffer, sizeof(buffer))) > 0) {
        ssize_t bytes_written = write(output_fd, buffer, bytes_read);
        if (bytes_written != bytes_read) {
            perror("Error writing to the output file");
            close(input_fd);
            close(output_fd);
            return 1;
        }
    }

    if (bytes_read == -1) {
        perror("Error reading from the input file");
        close(input_fd);
        close(output_fd);
        return 1;
    }

    close(input_fd);
    close(output_fd);

    printf("Contents of %s have been successfully copied to %s\n", input_filename, output_filename);
    return 0;
}
```

Output of the program

```
(shreygrg@Linux23)~[~/Shrey_Garg]
$ cat input.txt
Hello this is Shrey solving the practise Exercises
12218692

(shreygrg@Linux23)~[~/Shrey_Garg]
$ nano 2.c

(shreygrg@Linux23)~[~/Shrey_Garg]
$ gcc 2.c

(shreygrg@Linux23)~[~/Shrey_Garg]
$ ./a.out
Contents of input.txt have been successfully copied to output.txt

(shreygrg@Linux23)~[~/Shrey_Garg]
$ cat output.txt
Hello this is Shrey solving the practise Exercises
12218692
```

Exercise 3: -

Write a program in C that reads a file called "input.txt" and counts the number of lines in the file. You should use system calls like `open()`, `read()`, and `write()`.

Ans.

Step 1: - `cat input.txt`

Step 2: - `nano 3.c`

Step 3: - write the code

Step 4: - `gcc 3.c`

Step 5: - `./a.out`

Code: -

```
GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
int main() {
    const char *filename = "input.txt";
    int fd;
    fd = open(filename, O_RDONLY);
    if (fd == -1) {
        perror("Error opening the file");
        return 1;
    }
    char buffer[1];
    int line_count = 0;
    ssize_t bytes_read;
    while ((bytes_read = read(fd, buffer,
        sizeof(buffer))) > 0) {
        if (buffer[0] == '\n') {
            line_count++;
        }
    }
    if (bytes_read == -1) {
        perror("Error reading the file");
        close(fd);
        return 1;
    }
    if (bytes_read > 0 && buffer[0] != '\n') {
        line_count++;
    }
    printf("Number of lines in %s: %d\n",
        filename, line_count);
    close(fd);
    return 0;
}
```

Output of the program

```
(shreygrg@Linux23)-[~/Shrey_Garg]
$ cat input.txt
Hello this is Shrey solving the practise Exercises
12218692Adding the more lines in the file
Hello
World
Welcome to the New world.

(shreygrg@Linux23)-[~/Shrey_Garg]
$ nano 3.c

(shreygrg@Linux23)-[~/Shrey_Garg]
$ gcc 3.c

(shreygrg@Linux23)-[~/Shrey_Garg]
$ ./a.out
Number of lines in input.txt: 4
```

Exercise 4: -

- A. Write a C program that creates a file called “numbers.txt” and writes 100 integers to it, one integer per line.
- B. Using lseek() system call, move the file pointer to the beginning of file.
- C. Read the first 10 integers from the file and print them to the console.

Ans.

Step 1: - nano 4.c

Step 2: - Write the code

Step 3: - gcc 4.c

Step 4: - ./a.out

Step 5: - cat number.txt

Code: -

```
GNU nano 7.2 4.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

int main() {
    const char *filename = "number.txt";
    int fd;

    fd = open(filename, O_RDWR | O_CREAT | O_TRUNC, 0666);

    if (fd == -1) {
        perror("Error opening the file");
        return 1;
    }

    for (int i = 1; i <= 100; i++) {
        char num_str[16];
        snprintf(num_str, sizeof(num_str), "%d\n", i);
        write(fd, num_str, strlen(num_str));
    }
    lseek(fd, 0, SEEK_SET);

    char buffer[16];
    for (int i = 0; i < 10; i++) {
        ssize_t bytes_read = read(fd, buffer, sizeof(buffer));
        if (bytes_read <= 0) {
            break;
        }
        buffer[bytes_read] = '\0';
        printf("%s", buffer);
    }

    close(fd);
    return 0;
}
```

Output of the program

```
(shreygrg@Linux23)-[~/Shrey_Garg]
$ nano 4.c
(shreygrg@Linux23)-[~/Shrey_Garg]
$ gcc 4.c
(shreygrg@Linux23)-[~/Shrey_Garg]
$ ./a.out
1
2
3
4
5
6
7
8
9
10
```

Exercise 5: -

Write a C program that prints the last 10 characters of a file named as "input.txt" on the screen. Use open, read, write and lseek system calls.

Ans.

Step 1: - cat input.txt

Step 2: - nano 5.c

Step 3: - Write the code.

Step 4: - gcc 5.c

Step 5: - ./a.out

Code: -

```
GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main() {
    const char *filename = "input.txt";
    int fd;
    fd = open(filename, O_RDONLY);
    if (fd == -1) {
        perror("Error opening the file");
        return 1;
    }

    off_t file_size = lseek(fd, 0, SEEK_END);
    off_t offset = file_size - 10;

    if (offset < 0) {
        offset = 0;
    }
    lseek(fd, offset, SEEK_SET);
    char buffer[11];
    ssize_t bytes_read = read(fd, buffer, sizeof(buffer) - 1);

    if (bytes_read == -1) {
        perror("Error reading the file");
        close(fd);
        return 1;
    }
    buffer[bytes_read] = '\0';
    printf("Last 10 characters of %s:\n%s\n", filename, buffer);
    close(fd);
    return 0;
}
```

Output of the program

```
(shreygrg@ Linux23)-[~/Shrey_Garg]
$ cat input.txt
Hello this is Shrey solving the practise Exercises
12218692Adding the more lines in the file
Hello
World
Welcome to the New world.

(shreygrg@ Linux23)-[~/Shrey_Garg]
$ nano 5.c

(shreygrg@ Linux23)-[~/Shrey_Garg]
$ gcc 5.c

(shreygrg@ Linux23)-[~/Shrey_Garg]
$ ./a.out
Last 10 characters of input.txt:
New world.
```

Exercise 6: -

Write a C program that prints half content of a file named as "input.txt" on the screen. Use open, read, write and lseek system calls. If there are 100 characters written in the file, your program should display the first 50 characters on the screen.

Ans.

Step 1: - cat input.txt

Step 2: - nano 6.c

Step 3: - Write the code.

Step 4: - gcc 6.c

Step 5: - ./a.out

Code: -

```
GNU nano 7.2
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>

#define MAX_BUF_SIZE 50

int main() {
    int fd;
    ssize_t bytesRead;
    char buffer[MAX_BUF_SIZE];

    // Open the file for reading
    fd = open("input.txt", O_RDONLY);
    if (fd == -1) {
        perror("Error opening the file");
        exit(1);
    }

    // Get the file size using lseek
    off_t fileSize = lseek(fd, 0, SEEK_END);

    // Seek back to the beginning of the file
    lseek(fd, 0, SEEK_SET);

    // Calculate the number of characters to read (half of the file)
    off_t halfSize = fileSize / 2;

    // Read and print the first half of the file
    while (halfSize > 0) {
        ssize_t bytesToRead = (halfSize < MAX_BUF_SIZE) ? halfSize : MAX_BUF_SIZE;
        bytesRead = read(fd, buffer, bytesToRead);

        if (bytesRead == -1) {
            perror("Error reading the file");
            exit(1);
        }

        // Write the read data to stdout (screen)
        write(STDOUT_FILENO, buffer, bytesRead);

        halfSize -= bytesRead;
    }

    // Close the file
    close(fd);

    return 0;
}
```

Output of the program

```
File Actions Edit View Help

(shreygrg@Linux23)~[/Shrey_Garg]
$ nano 6.c

(shreygrg@Linux23)~[/Shrey_Garg]
$ cat input.txt
Hello this is Shrey solving the practise Exercises
12218692Adding the more lines in the file
Hello
World
Welcome to the New world.We are learning new things about file manipulation
Sothis is the starting of something new
Zombie process in the fork statementA
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Hello world we are learning the

(shreygrg@Linux23)~[/Shrey_Garg]
$ nano 6.c

(shreygrg@Linux23)~[/Shrey_Garg]
$ gcc 6.c

(shreygrg@Linux23)~[/Shrey_Garg]
$ ./a.out
Hello this is Shrey solving the practise Exercises
12218692Adding the more lines in the file
Hello
World
Welcome to the New world.We are learning new things ab
```


D. Directory Manipulation Using System Calls

Exercise 1: -

- A. Write a C program that opens a directory called "my_directory" and reads all the files and directories inside it.
- B. For each file and directory, print its name and whether it is a file or a directory.
- C. Count the number of files and directories inside the "my directory" directory.
- D. Close the directory.

Ans.

Step 1: - mkdir my_directory

Step 2: - create some files

Step 3: - nano D_1.c

Step 4: - gcc D_1.c

Step 5: - ./a.out

Code: -

```
GNU nano 7.2 D_1.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>

int main() {
    const char *dirname = "my_directory";
    DIR *dir;
    dir = opendir(dirname);
    if (dir == NULL) {
        perror("Error opening the directory");
        return 1;
    }
    int file_count = 0;
    int dir_count = 0;
    struct dirent *entry;
    while ((entry = readdir(dir)) != NULL) {
        if (strcmp(entry->d_name, ".") == 0 || strcmp(entry->d_name, "..") == 0) {
            continue;
        }
        if (entry->d_type == DT_REG) {
            printf("File: %s\n", entry->d_name);
            file_count++;
        } else if (entry->d_type == DT_DIR) {
            printf("Directory: %s\n", entry->d_name);
            dir_count++;
        }
    }
    if (closedir(dir) == -1) {
        perror("Error closing the directory");
        return 1;
    }
    printf("Total files: %d\n", file_count);
    printf("Total directories: %d\n", dir_count);
    return 0;
}
```

Output of the program

```
(shreygrg@Linux23) - [~/Shrey_Garg]
$ gcc D_1.c

(shreygrg@Linux23) - [~/Shrey_Garg]
$ ./a.out
File: 2.txt
File: 4.txt
File: 3.txt
File: 1.txt
File: 5.txt
Total files: 5
Total directories: 0
```

Submitted to: -

Mr. Ashish Kumar Singh

Section: K22QY

Submitted By: -

**Sahil Singh
(12221802)**

**Shrey Garg
(12218692)**

**Abhay P Aneesh
(12201683)**