**Concepts of Operating System**

***Assignment 2***

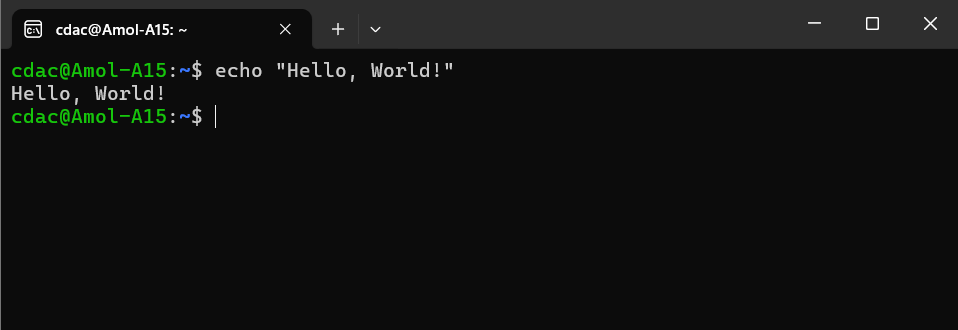
***Part A***

**What will the following commands do?**

1. echo "Hello, World!"

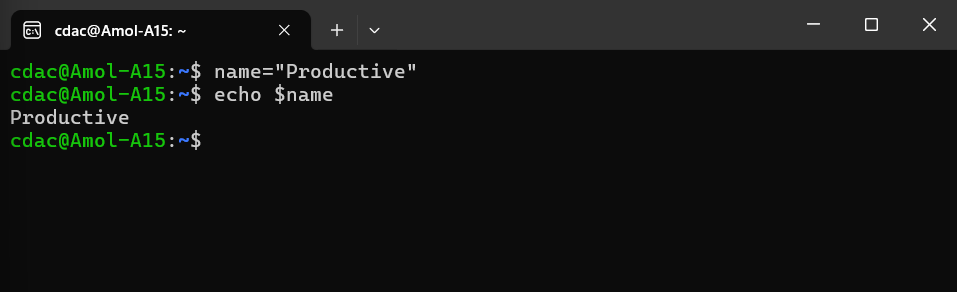
Display message on screen

Hello, World



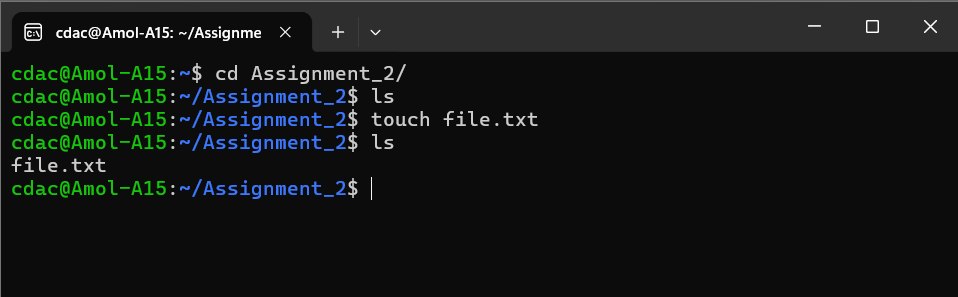
1. name="Productive"

Create a variable with the value as string {Productive}



1. touch file.txt

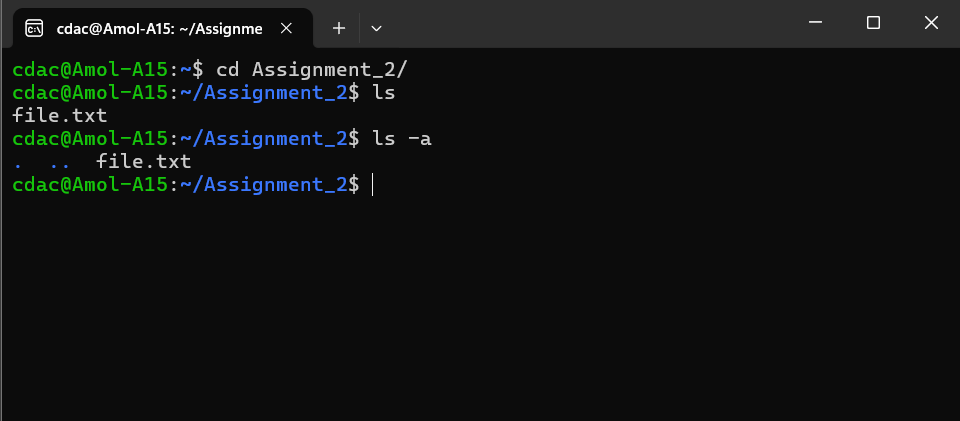
is use to create a new file



1. ls -a

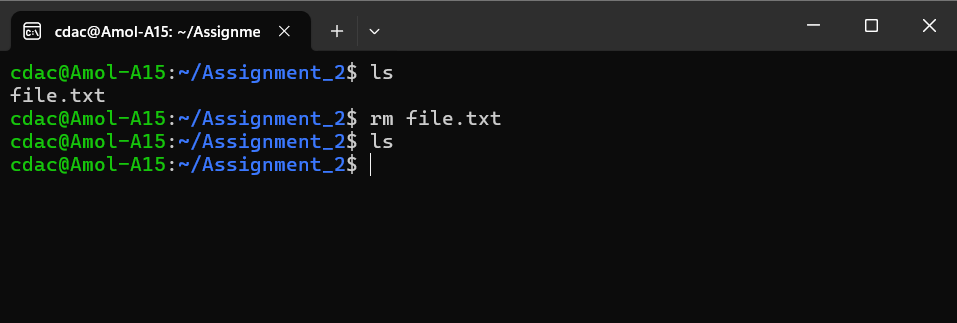
list directory contents

-a is use to not ignore entries starting with { . }



1. rm file.txt

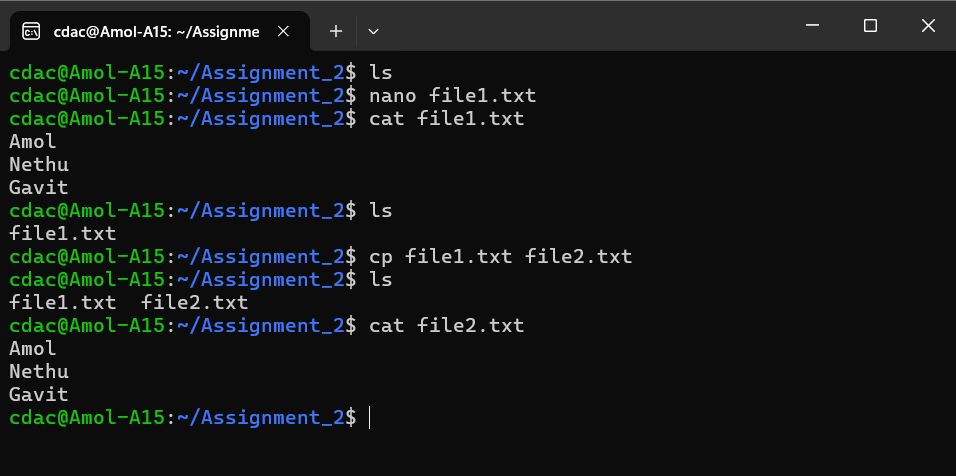
remove files or directories



1. cp file1.txt file2.txt

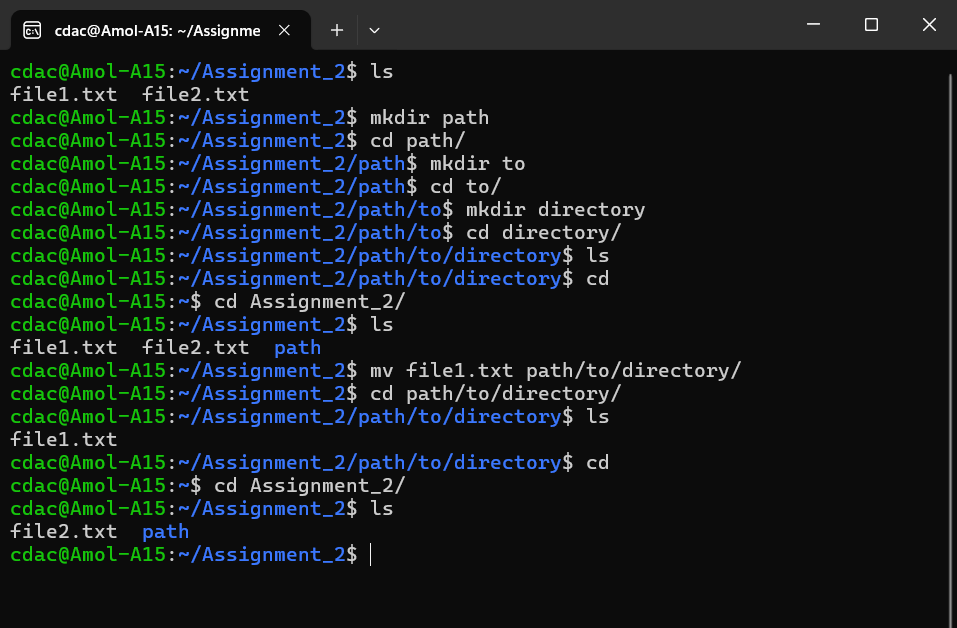
copy files and directories

used to copy the contains of file1.txt to file2.txt



1. mv file1.txt /path/to/directory/

move files

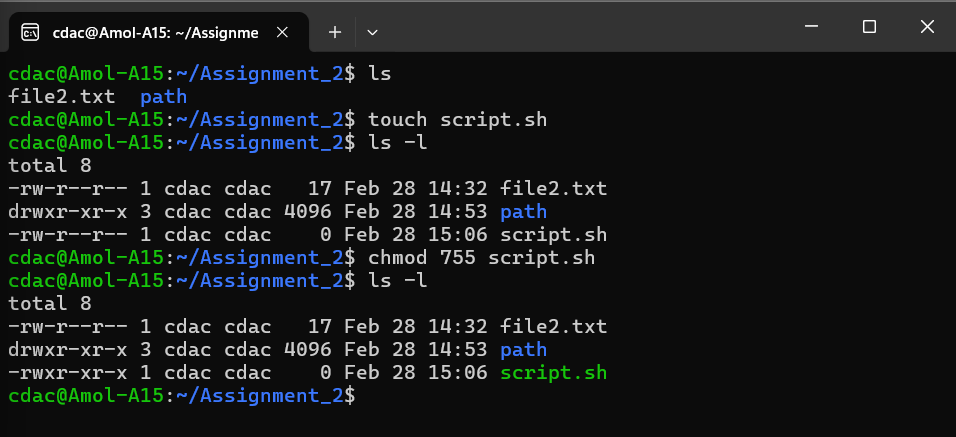


1. chmod 755 script.sh

change file permissions

Default file permissions are: u=rw, g=r, o=r { 644}

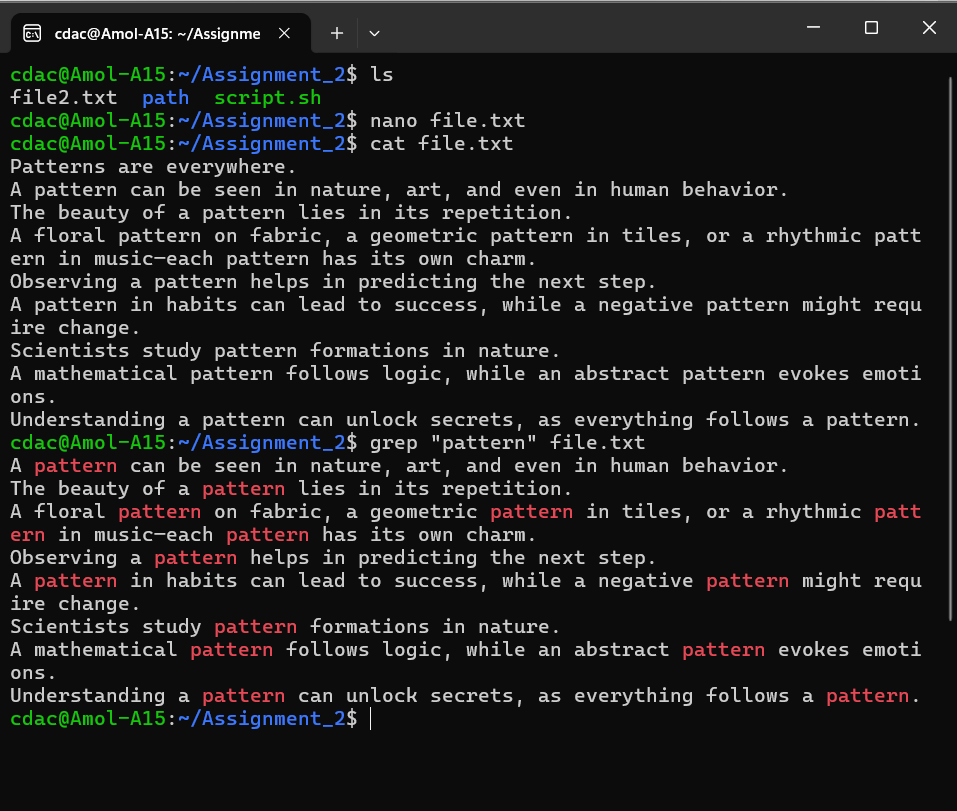
Change it to: { 755 } means u=rwx, g=rx, o=rx



1. grep "pattern" file.txt

print lines that match patterns

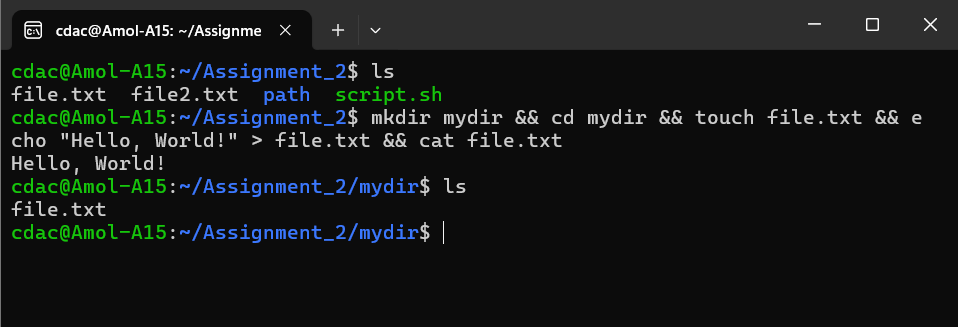
here we use “pattern” word to search in the { file.txt }



1. kill PID

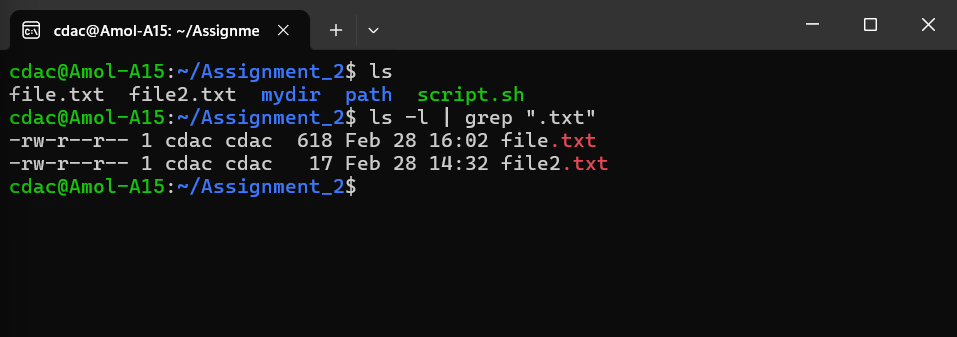
This option specifies the process ID of the process to be killed.

1. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
2. Crate the directory name mydir.
3. Change the directory to mydir.
4. Create the new file with name as “file.txt” in current directory.
5. Then append the output of echo “Hello, World!” in the file.txt.
6. Display the contains of the file.txt.

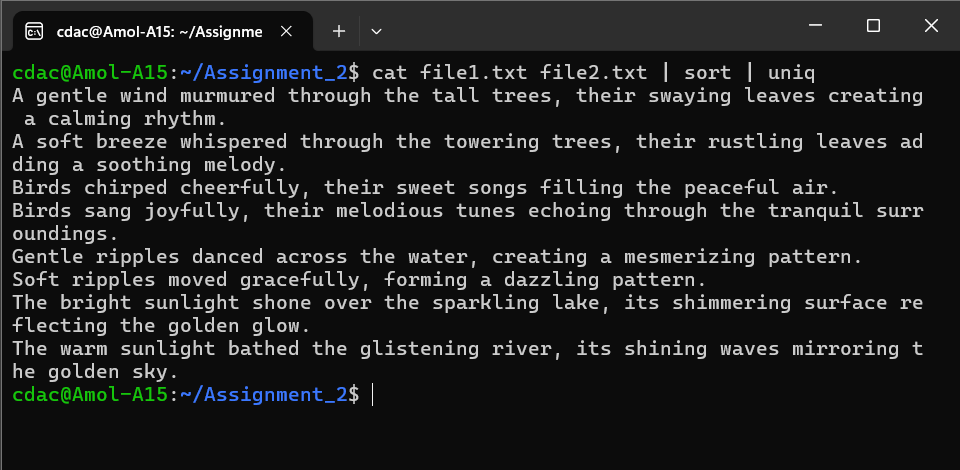


1. ls -l | grep ".txt"

{ | } this piping symbol is used to pass a program's output into another program's input.



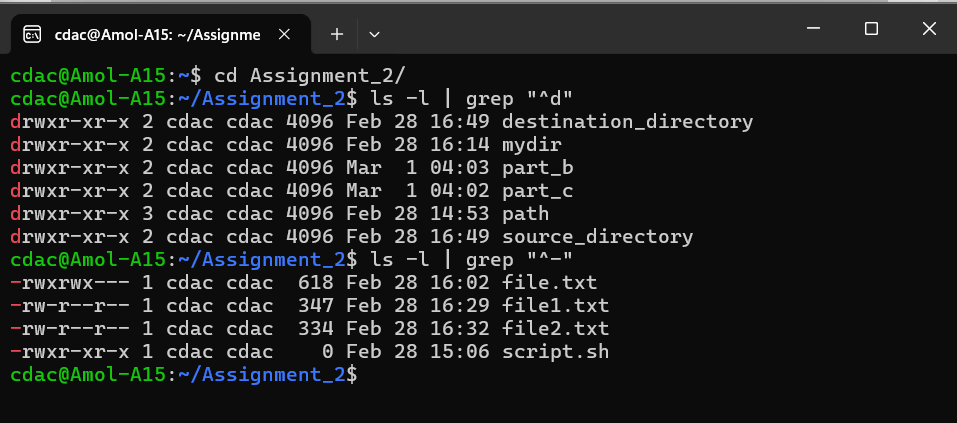
1. cat file1.txt file2.txt | sort | uniq
2. Print the contains of the both file
3. Then the output (contain of both files) is sort all lines according to the alphabetical order.
4. Then exclude all repeated lines.



1. ls -l | grep "^d"

ls -l is use to print the metadata of all directories as well as files present in the current directory.

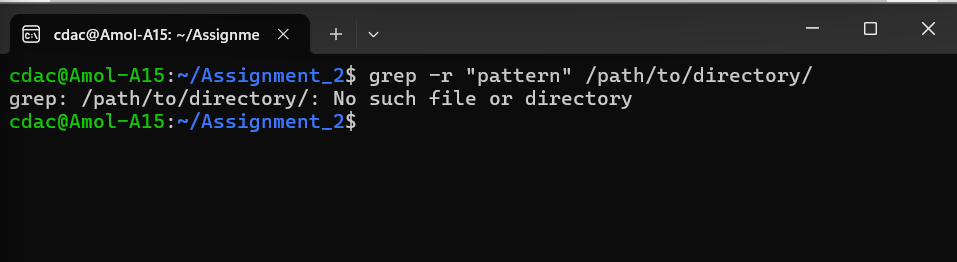
With grep we can filter out the { “^d” } only directories. or files by using { “^-” }.



1. grep -r "pattern" /path/to/directory/

Here grep command is used recursively to search for given pattern “pattern” in the given directory path.

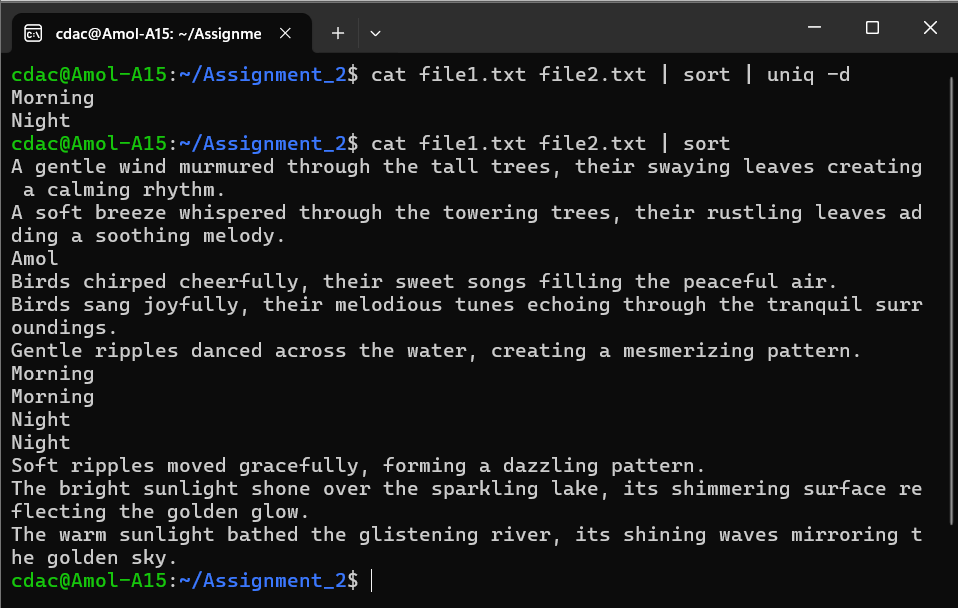
But there is no such file present in our directory.



1. cat file1.txt file2.txt | sort | uniq –d

we are combining the contains of the file1.txt and file2.txt then sorting them alphabetically.

Then by using uniq -d command we are printing only duplicate values.

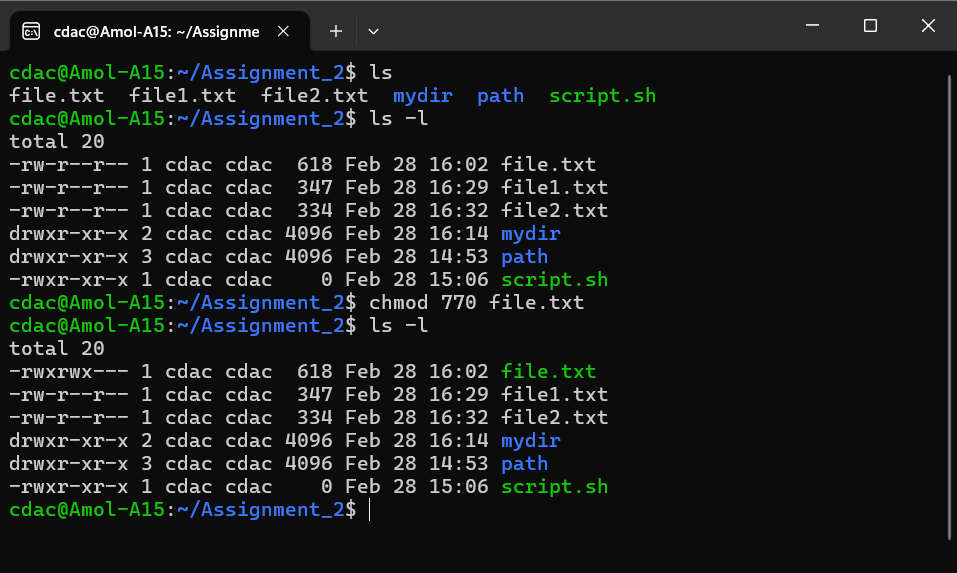


1. chmod 644 file.txt

change file permissions

Default file permissions are: u=rw, g=r, o=r { 644}

Change it to: { 770 } means u=rwx, g=rwx, o=---

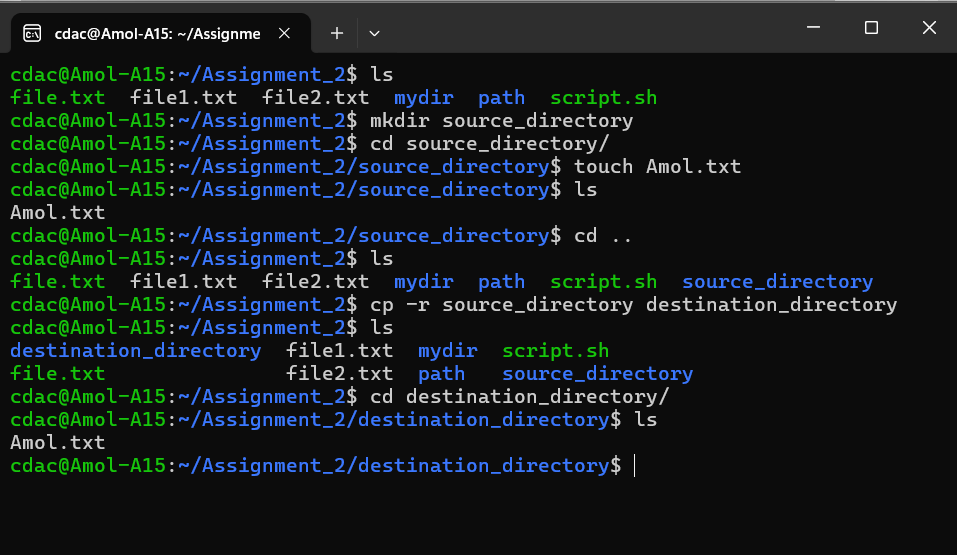


1. cp -r source\_directory destination\_directory

copy files and directories

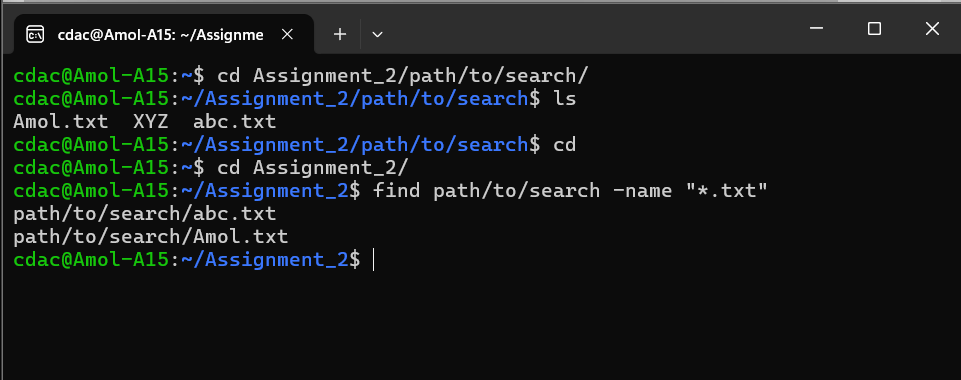
-r is use to copy directories recursively

Here recursively means all the contains of the source\_directory are copied to destination\_directory.



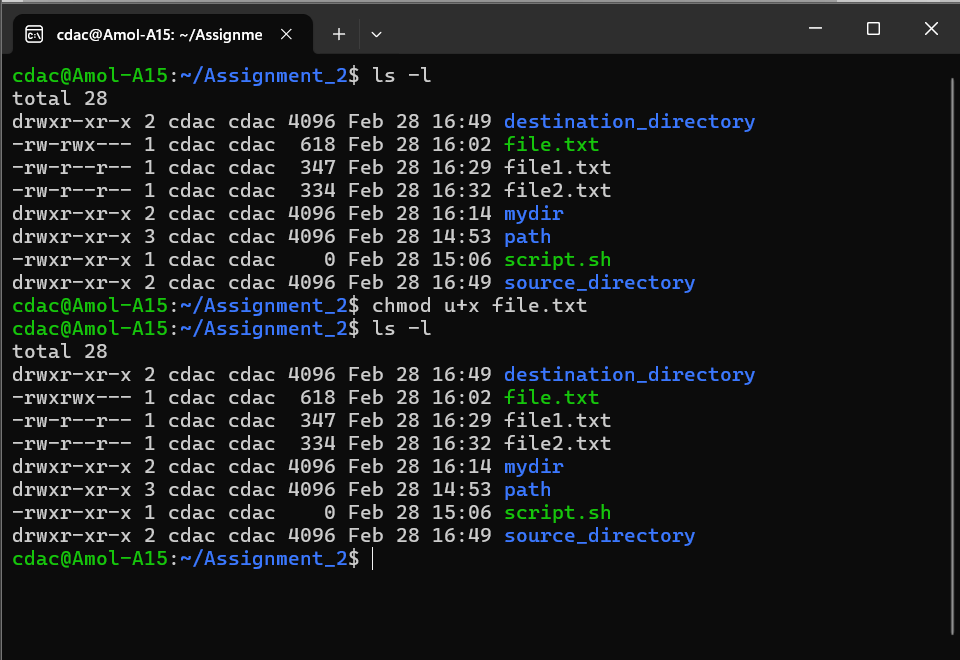
1. find /path/to/search -name "\*.txt"

use to find the { .txt } file name in the given directory path.

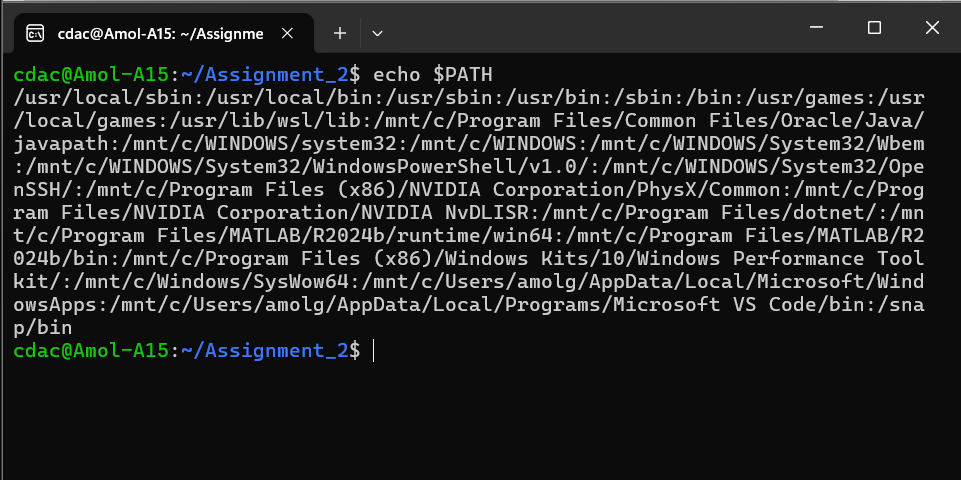


1. chmod u+x file.txt

Initially user didn’t have the permission to execute the file, but by using the{ u+x } we give permission to execute the file.txt to user.



1. echo $PATH

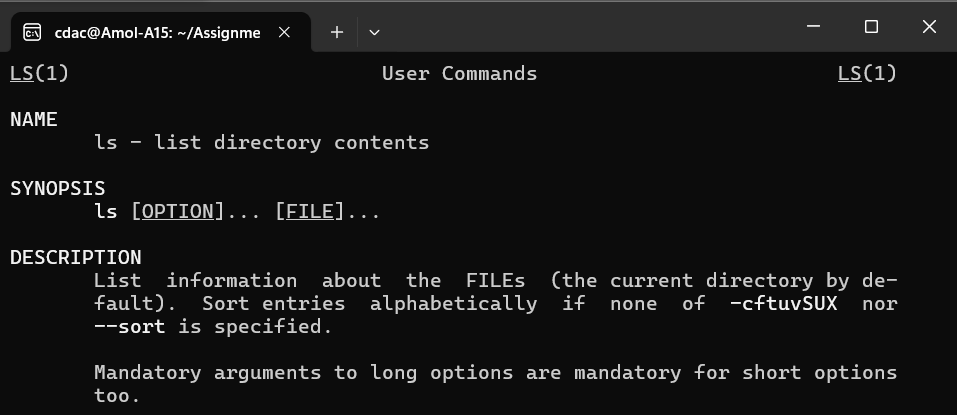


***Part B***

**Identify True or False:**

1. ls is used to list files and directories in a directory.

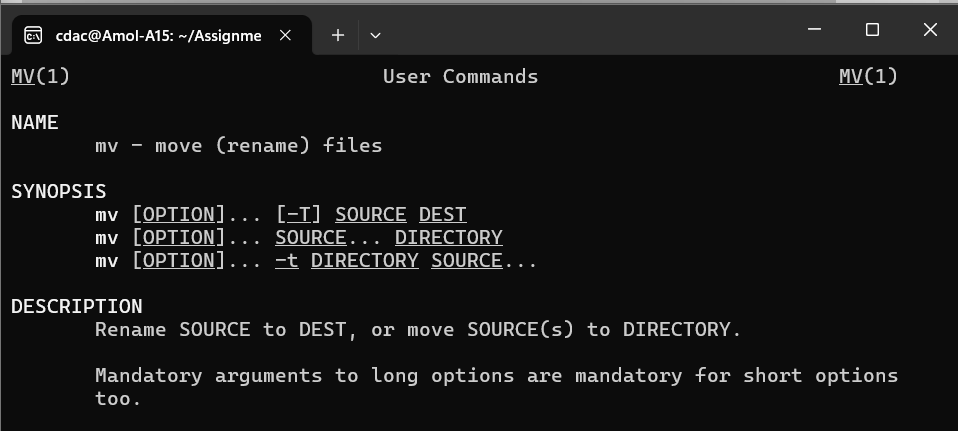
This statement is **true**, Check the manual of ls.



1. mv is used to move files and directories.

This statement is **true**, Check the manual of mv.

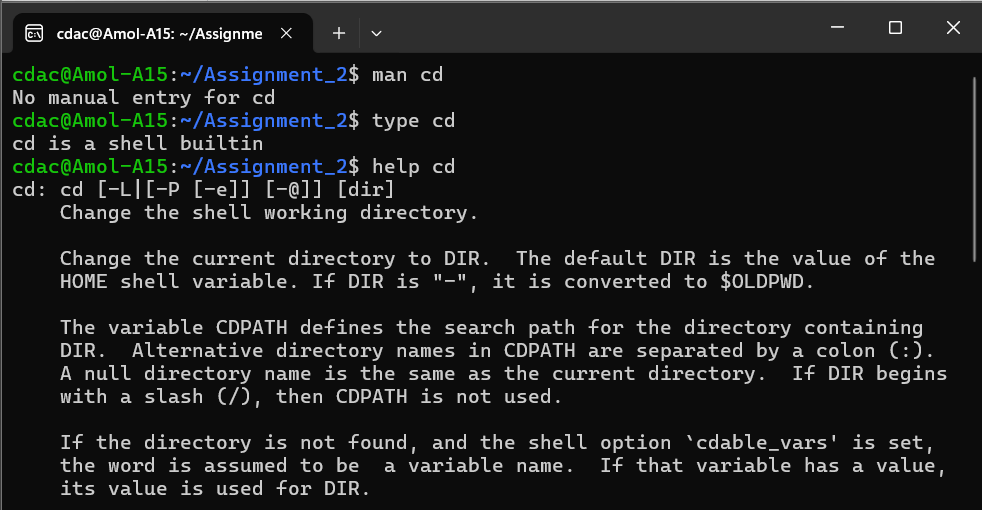
Also, this command is use to rename files.



1. cd is used to copy files and directories.

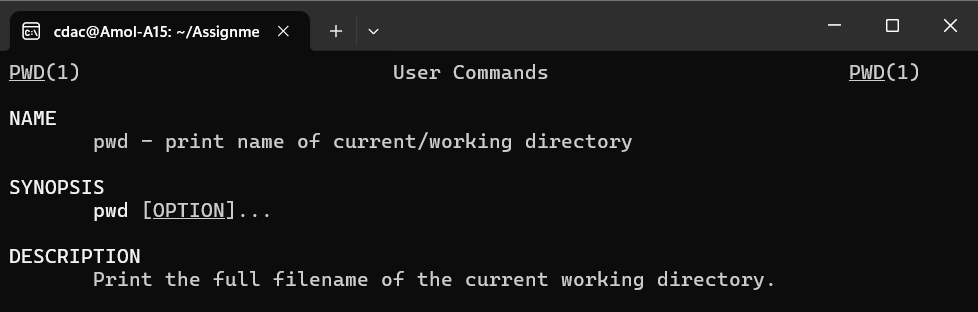
This statement is **false**, cd command is use to change the directory.

{cd is builtin shell command therefor we can’t call it’s manual to need to use help for the manual to print}



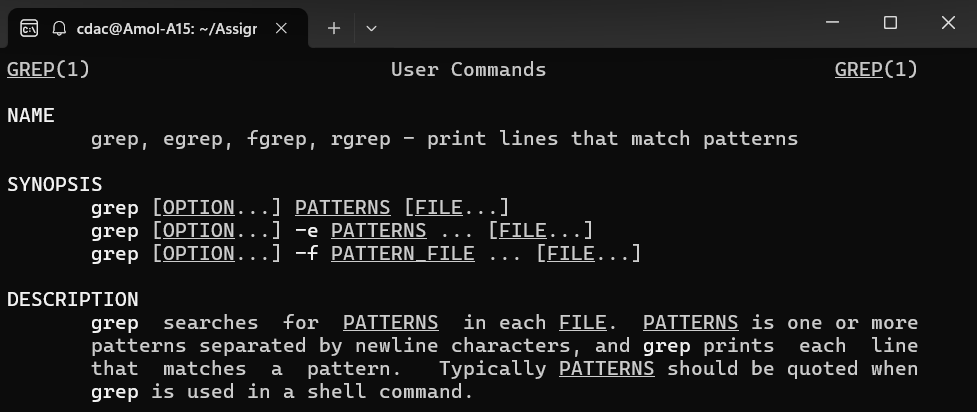
1. pwd stands for "print working directory" and displays the current directory

This statement is **true**, Check the manual of pwd.



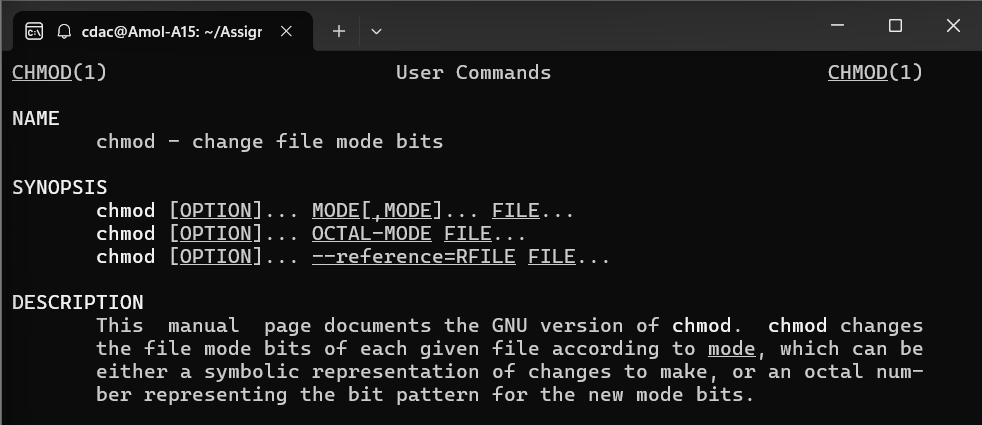
1. grep is used to search for patterns in files.

This statement is **true**, Check the manual of grep.



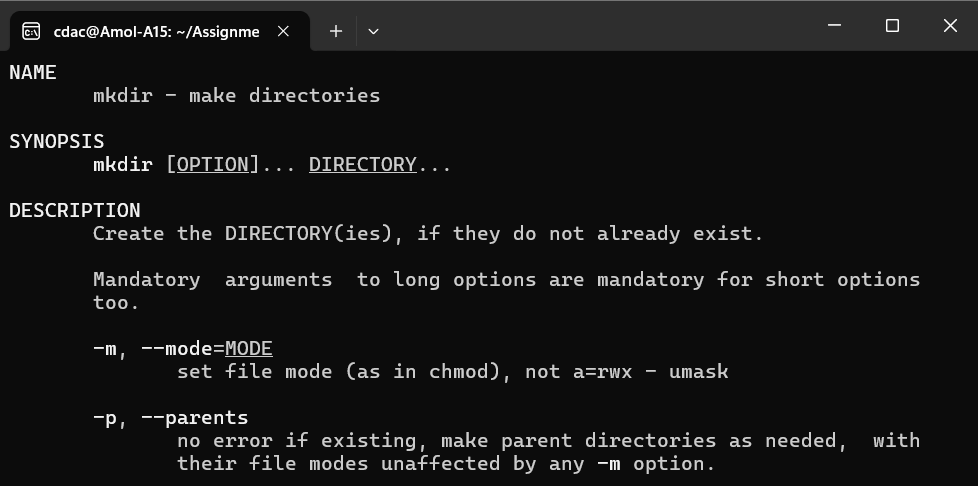
1. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

This statement is **true**, Check the manual of chmod.



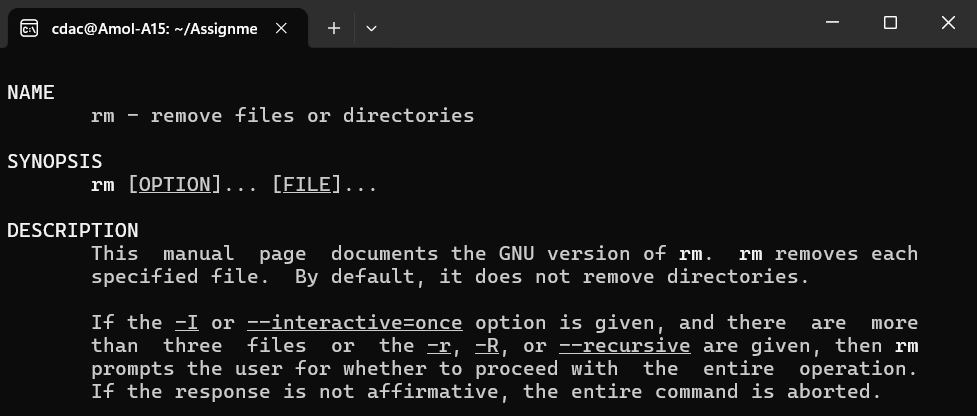
1. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.

This statement is **true**, Check the manual of mkdir.

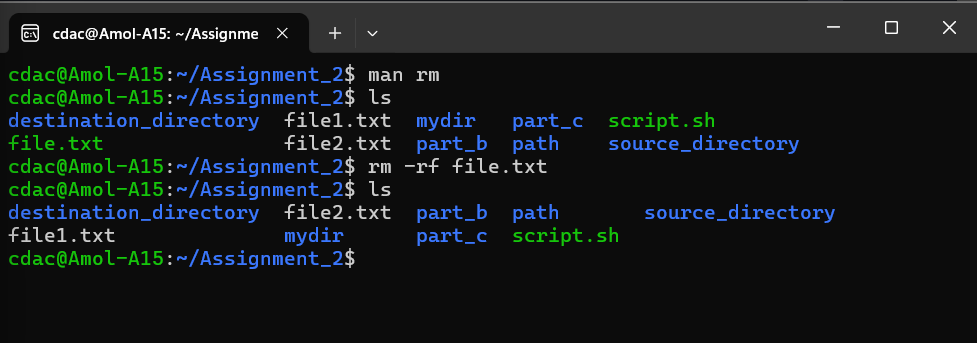


1. rm -rf file.txt deletes a file forcefully without confirmation.

This statement is **true**, Check the manual of rm.

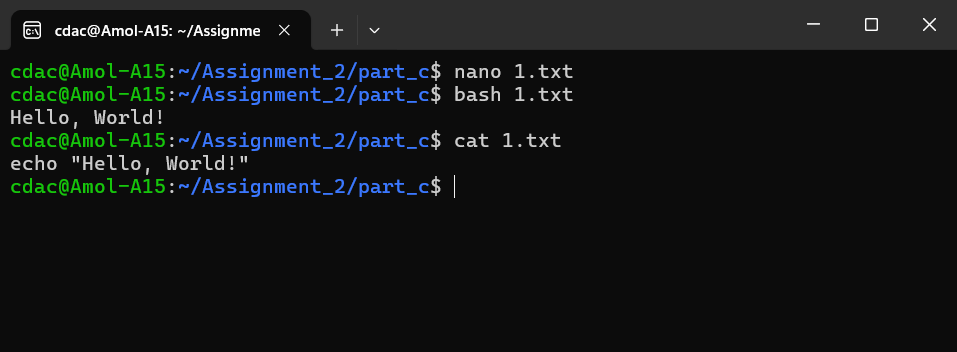


Practical Implementation

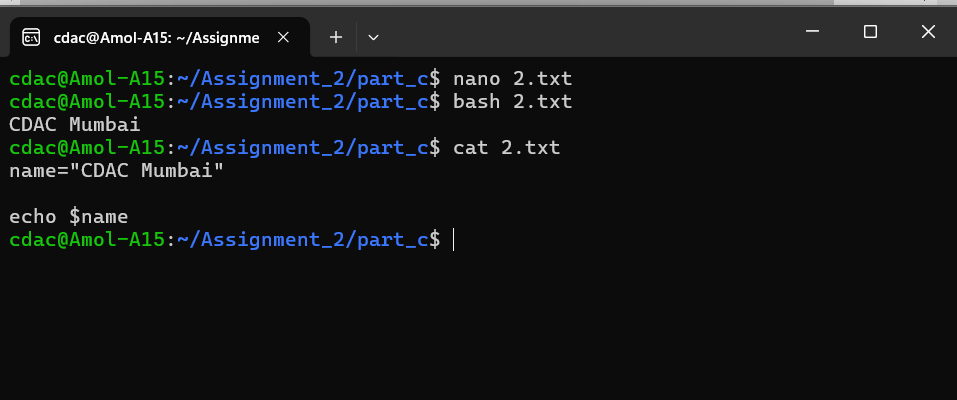


***Part C***

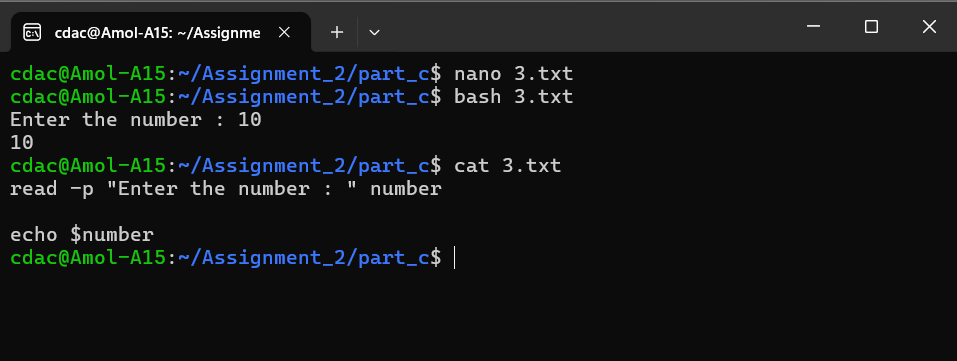
1. Write a shell script that prints "Hello, World!" to the terminal.



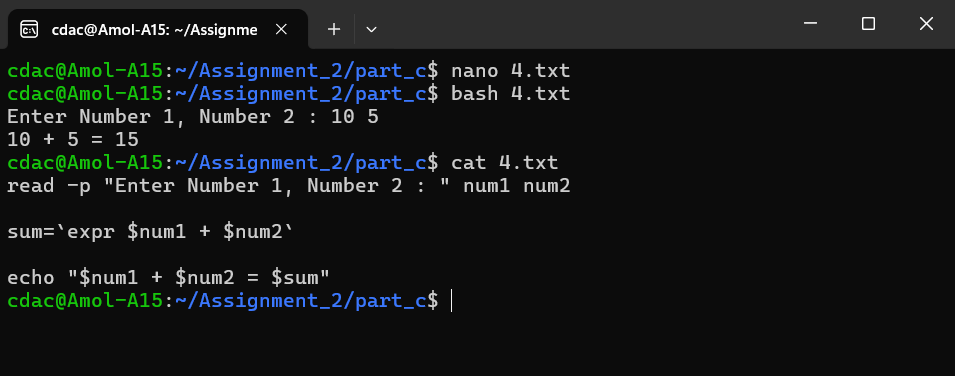
1. Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

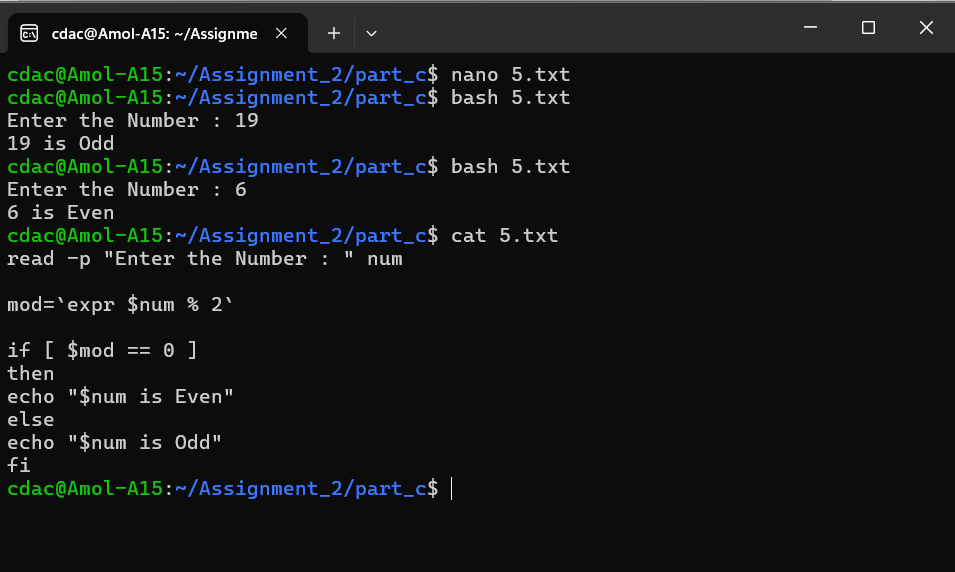


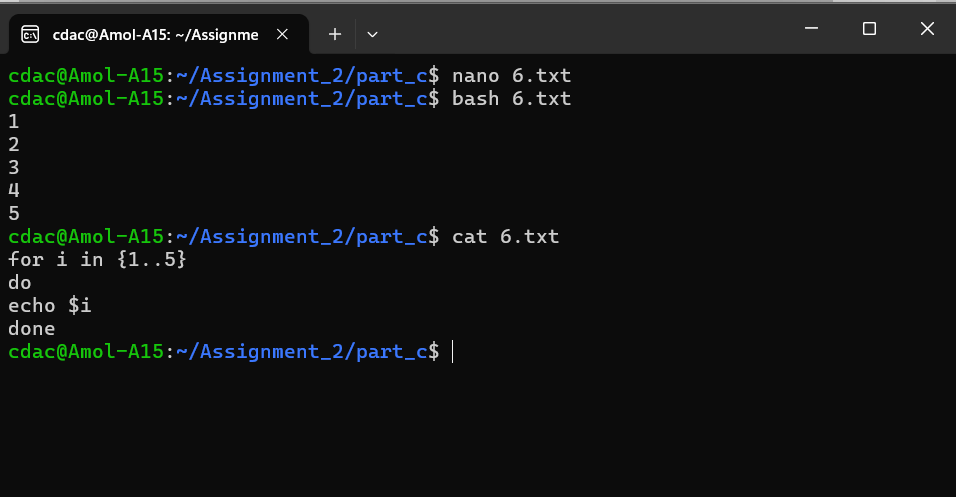
1. Write a shell script that takes a number as input from the user and prints it.



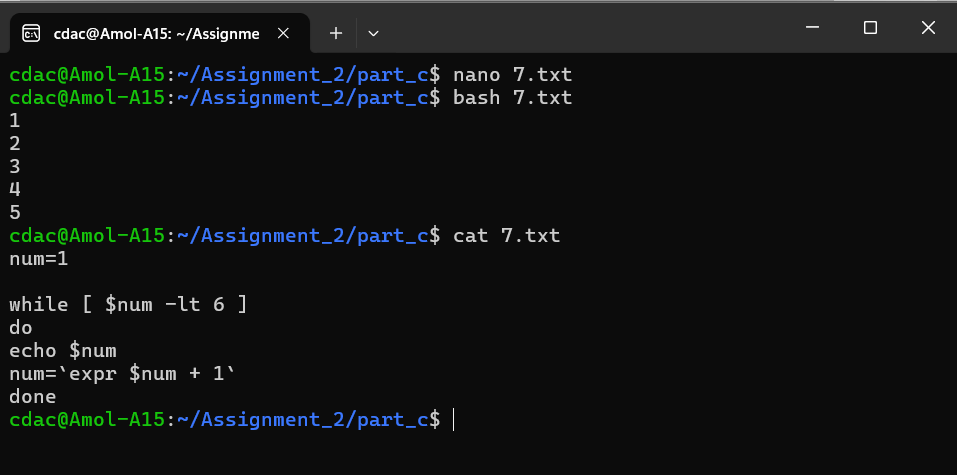
1. Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



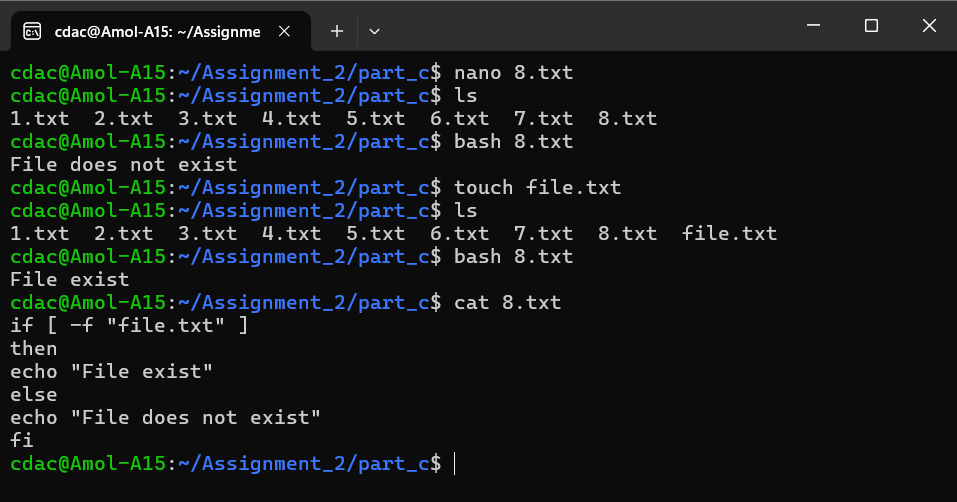
1. Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".
2. Write a shell script that uses a for loop to print numbers from 1 to 5.



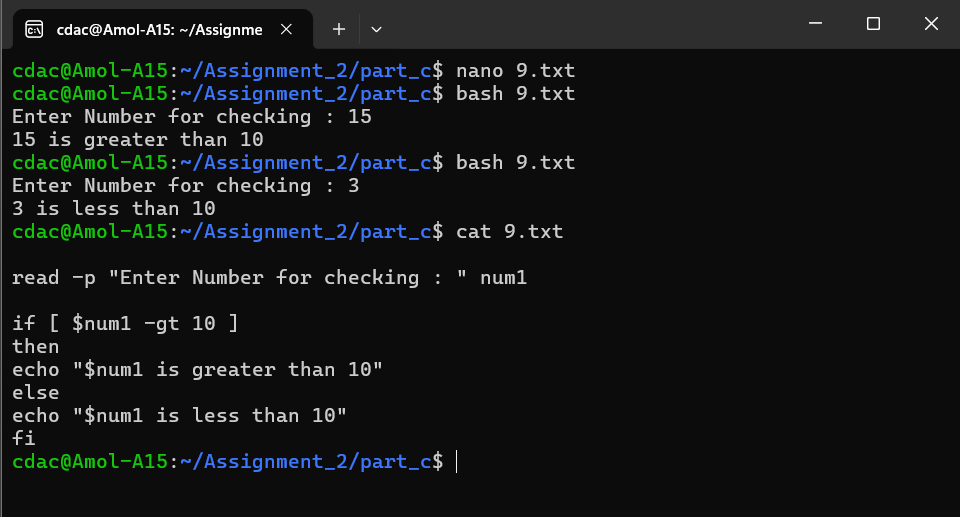
1. Write a shell script that uses a while loop to print numbers from 1 to 5.



1. Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



1. Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



1. Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.
2. Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the break statement to exit the loop when a negative number is entered.